目录

[第一章 基础 3](#_Toc114903639)

[1.1 序言 3](#_Toc114903640)

[1.2 输入文件准备 4](#_Toc114903641)

[1.2.1 基本概念 4](#_Toc114903642)

[1.2.2 INCAR 4](#_Toc114903643)

[1.2.3 KPOINTS 5](#_Toc114903644)

[1.2.4 POSCAR 6](#_Toc114903645)

[1.2.5 POTCAR 7](#_Toc114903646)

[1.3 批量处理方法 9](#_Toc114903647)

[1.3.1 Part 1 9](#_Toc114903648)

[1.3.2 Part II 11](#_Toc114903649)

[1.3.3 Part III 13](#_Toc114903650)

[1.3.4 Part IV 15](#_Toc114903651)

[1.4 Vasp任务提交（在自己的虚拟机上） 16](#_Toc114903652)

[1.5 Vasp输出文件 18](#_Toc114903653)

[1.5.1 Vasp输出文件1 18](#_Toc114903654)

[1.5.2 Vasp输出文件2 21](#_Toc114903655)

[1.6 氧原子能量计算（对应ex6） 24](#_Toc114903656)

[1.7 收敛点 28](#_Toc114903657)

[1.8 O2结构搭建以及POTCAR注意部分 30](#_Toc114903658)

[1.9 O2单点计算和优化结果分析 32](#_Toc114903659)

[1.10 VASP中MAGMOM的简单设定 37](#_Toc114903660)

[第二章 论合理结构的重要性 38](#_Toc114903661)

[2.1 论合理结构的重要性（一） 38](#_Toc114903662)

[2.1.1 测试bad结构 38](#_Toc114903663)

[2.1.2 可能出现情况 40](#_Toc114903664)

[2.1.3 linux中添加自定义命令alias 41](#_Toc114903665)

[2.2 论合理结构的重要性（二） 41](#_Toc114903666)

[2.2.1 不同初始结构对结果影响 41](#_Toc114903667)

[2.2.2 另一个bad结构的计算 42](#_Toc114903668)

[2.3 论合理结构的重要性（三） 44](#_Toc114903669)

[2.3.1 电子步数不收敛的解决办法 44](#_Toc114903670)

[2.3.2 设置键长为0.9，计算失败原因： 45](#_Toc114903671)

[2.3.3 更改POTIM重新计算 45](#_Toc114903672)

[第三章 谁偷走我的机时 46](#_Toc114903673)

[3.1 谁偷走我的机时1 46](#_Toc114903674)

[3.2 谁偷走我的机时2 47](#_Toc114903675)

[3.3 谁偷走我的机时3 51](#_Toc114903676)

[3.4 谁偷走了我的机时4 52](#_Toc114903677)

[3.5 谁偷走我的机时5 53](#_Toc114903678)

[第四章 乙醇 54](#_Toc114903679)

[4.1 乙醇气相分子的优化 54](#_Toc114903680)

[4.2 乙醇分子的震动频率1 55](#_Toc114903681)

[4.3 乙醇分子的振动频率2 56](#_Toc114903682)

[第五章 块体计算 59](#_Toc114903683)

[5.1 Bulk的单点计算（Fe单胞） 59](#_Toc114903684)

[5.2 晶格参数的确定（Birch-Murnaghan状态方程） 60](#_Toc114903685)

[5.3 晶格参数的确定（直接优化晶格常数） 61](#_Toc114903686)

[第六章 DOS计算 63](#_Toc114903687)

# 基础

## 序言

推荐书籍：

* density functional theory :A practical introduction
* VASP官网

学习方法：

老板+Density Functional Theory: A Practical Introduction+vaspwiki+vasp官方论坛

图形用户界面, 文本, 应用程序

描述已自动生成

1：查询参数含义，设置的时候，进行搜索；

2：新手们从这里点击开始；

3：VASP的计算实例，大家可以参考里面的说明计算自己的体系性质；

4：INCAR的参数列表，大家没事多多点击里面的各个参数；查看含义；

5：再下面就是一些理论背景知识了，建议系统学习DFT的相关书籍，Wiki中的内容有限，只能作为参考。

## 输入文件准备

四个重要输入文件：INCAR, KPOINTS, POSCAR, POTCAR

* INCAR 告诉VASP算什么，怎么算。
* KPOINTS 包含计算的K点信息(计算的时候必须有这个东西)。
* POSCAR是计算模型结构的文字描述，也就是各个原子在xyz方向的坐标信息。
* POTCAR对应的是计算过程中每个原子的平面波基组，描述体系中对应的原子核和电子的相关信息。

### 基本概念

* 不懂参数：官网+思考
* 输入文件可以通过文本编辑器打开，Linux用vim，windows用notepad++，不用windows自带的记事本
* 输入文本没有**后缀**

Vasp输入和输出

### INCAR

作用：what to do and how to do it

准备原则：越简单越好，**不知道参数不放进去**

手机屏幕的截图

描述已自动生成

注意事项：

* + #后面表示注释，最好是英语
  + =后面可以有空格，也可以没有，ISMEAR=0 和 ISMEAR = 0 是一样的。
  + **不要使用 tab，**用空格替换tab

参数详解：

* SYSTEM（随便写）
  + SYSTEM = O atom 和 SYSTEM = I LOVE BigBro 和 SYSTEM = VASP， 除了多出几个字符外，对计算没有什么影响。
  + 这一行有或者没有，对计算结果不会有任何影响，看着不爽也可以直接删除。
* ISMEAR
  + 不同的值对应的是不同的展宽方法
  + 对于分子，原子的计算用0
  + 其他事项：

文本

描述已自动生成

* SIGMA：

SIGMA的取值和ISMEAR息息相关，也就是和ISMEAR这个参数同时出现，设定。

* + 如果用了ISMEAR = -5； SIGMA的值可以忽略，也可以不管。（对于所有体系都适用)；
  + 对于金属： ISMEAR = 1 或者0，非金属: ISMEAR= 0 的时候，一般取 SIGMA = 0.10 即可，默认值是0.20。不放心的话，用0.05。
  + 对于气体分子，原子体系（也就是你把分子或者原子放到一个box里面）：死死记住下面组合就可以了， ISMEAR = 0; SIGMA = 0.01。
  + 测试：如果用了ISMEAR = -5； SIGMA的值可以忽略，也可以不管。（对于所有体系都适用)；检查时使用命令：grep 'entropy T' OUTCAR

### KPOINTS

KPOINTS文件在计算中的作用，一方面决定了计算的精度，同时也影响计算所需要的时间。精度高，时间长，精度低，时间短。

图形用户界面, 文本, 应用程序

描述已自动生成

注意事项：

* + 第一行：可以随便写内容，但不能没有;
  + 第二行：数字0，不是字母O，0 在这里表示的是**自动生成K点**。你可能注意到了，0前面有个空格，不过没关系，对计算不影响，可以删掉，也可以再加一个空格，后面的同样如此。 但是避免用tab输入，VASP有时候不识别tab键输入的空白部分，导致计算出错；
  + VASP只认第一个字母，大小写均可。在这里Gamma和gamma效果是一样的，当然这一行也可以直接写字母G或者g。
  + 第四行，在xyz三个方向上生成对应数目的K点，本例是111;
    - 对于原子或者分子的计算，K点取一个gamma点就够了（1 1 1），也就是这个KPOINTS文件可以用于绝大多数的原子或者分子计算，不用再修改；
    - 但如果你要用**ISMEAR = -5** 来计算能量，那么就需要把 1 1 1 改成 2 2 2或者3 3 3。
  + 但如果你要用ISMEAR = -5 来计算能量，那么就需要把 1 1 1 改成 2 2 2或者3 3 3。
  + 第五行，一般都写成 0 0 0 ，不用动即可；

KPOINTS建议：

* 第三行G保持不变
* 气体分子，原子计算的时候，使用gamma点就行

### POSCAR

POSCAR 里面包含了你要进行计算的模型结构信息。如：O原子的计算中，我们要把O原子放到一个格子里面，格子大小为：8 O原子的计算中，我们要把O原子放到一个格子里面，格子大小为：8 × 8 × 8

文本, 信件

描述已自动生成

注意事项：

* 第一行：必须写，可以随便写
* 第二行：Scale factor，称为缩放系数，这里是1.0；
* 第三到五行：是组成格子的三条边的坐标信息；从原点出发，在xyz轴上分别取8
* 第六行：体系中的元素，为氧原子
  + 第一个元素符号要顶格写，前面不要有空格，有可能会出错；
* 第七行：与第六行中元素相对应的原子数目，这里我们只有1个氧原子，所以写成1；
* 第八行：体系中原子的坐标系，可以为笛卡尔坐标，也可以为分数坐标系。
  + 这一行同KPOINTS的第三行一样，即只认第一个字母；
  + C或者c代表笛卡尔坐标， D或者d代表分数坐标系；
  + VASP输出文件CONTCAR里面采用的是分数坐标系。
* 第九行：体系中原子的坐标信息。这里我们把O原子放到了原点（0.0 0.0 0.0）的位置，大家也可以随便放一个位置，比如：（4.0 5.0 6.0），（1.1 2.5 6.5）都是可以的。由于周期性的存在，不管你怎么放，相邻两个格子之间氧原子的距离都是一样的。

说明：

* 第二行中的Scale factor还可以写成其他的数字，例如：写成2.0，则后面的格子以及原子坐标相关的数值都要除以2。一般来说，写成1.0即可
* 笛卡尔和分数坐标系的区别是从原子的坐标行开始的（这个例子里面是第8行），即坐标前面的都保持完全一致。 也就是说，如果想从笛卡尔转换成分数坐标，我们只需将Cartesian改成Direct，然后修改后面的原子坐标，而Cartesian行前面的部分保持不变。

### POTCAR

POTCAR四个主要参数意义：

* VRHFIN 用来看元素的价电子排布
* LEXCH 表示这个POTCAR对应的是GGA-PBE泛函；如果INCAR中不设定泛函，则默认通过这个参数来设定。
* TITEL指的是哪个元素
* ZVAL是POTCAR的价电子数目
* ENMAX指的是默认的截断能。

查看Fe元素：

文本

描述已自动生成

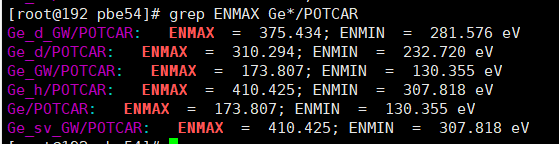
* 与GW 计算的对应的POTCAR，则标注为：Fe\_GW 这样。
* 根据价电子的处理方式，分成了诸如：Fe，Fe\_pv，Fe\_sv的这样的情况。v是valence的缩写。pv代表把内层的p电子作为价电子来处理。sv代表则是把更内层的s电子也作为价电子来处理。

获取Fe相关的价电子信息：

文本

描述已自动生成

查看截断能:Ge的截断能：



* Ge\_d表示把内层d轨道考虑到价电子层里面。
* 某些元素，还有一些以 \_h， \_s 结尾的，是 hard和soft的缩写。带h的POTCAR中截断能比普通的要高出很多。带s的截断能要小很多。

POTCAR选择：

参考vasp官网推荐：https://www.vasp.at/wiki/index.php/Available\_PAW\_potentials



通过这一行命令就可以把Al，C，H各自的POTCAR结合在一起。POTCAR中的元素顺序一定要和POSCAR保持一致,否则计算会出错

图形用户界面, 应用程序

中度可信度描述已自动生成

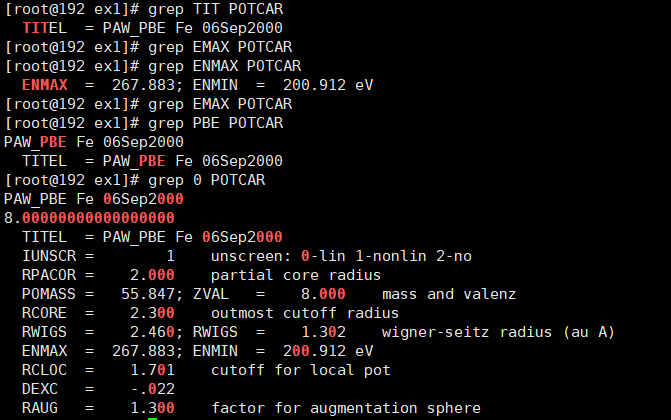
## 批量处理方法

### Part 1

文本

描述已自动生成

grep 后面提取的字符，最好在文件中是唯一存在的或者只出现几次。否则我们不容易得到期望的结果。运行以下结果，会出现多个结果：



More查看文件：

文本

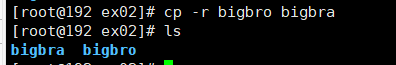
描述已自动生成

Mkdir创建文件夹，创建一个名字为：ex02的文件夹，然后进入，再创建一个birbro的文件夹:

文本

描述已自动生成

Cp命令复制文件夹：



将ex01中的四个输入文件复制到bigbra这个文件夹中：



文本

描述已自动生成

把bigbra 文件移动到bigbro里面：

文本

描述已自动生成

将bigbra文件夹重命名为： 0.01：

文本

描述已自动生成

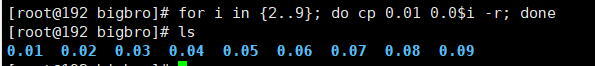
常见多个文件：

1）创建从0.02，0.03到0.10，共9个文件夹，

2）每个文件夹中都有与0.01文件夹中相同KPONTS, POSCAR，POTCAR

3）每个文件夹中都有INCAR，但INCAR中SIGMA这个参数的取值和文件夹的名字一样，其他参数和0.01的相同。

For循环实现：



* for i in {2..9} 和 for i in 2 3 4 5 6 7 8 9 效果是一样的，2和9中间有两个点。for i in XXX 这个句字后面跟着一个分号



* do 翻译过来就是: 我们要实现什么任务，目的; do 后面跟一个空格， 或者几个空格
* $i 被替换成for后面变量 i 的值;
* done 完成任务。

### Part II

Sed命令：

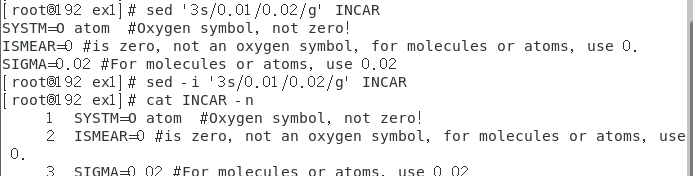
文本

描述已自动生成

* 单引号中是我们的操作， 3s 表示的是选择第三行，因为我们知道 0.01 在第三行中出现，s 是substitute 的缩写
* 3s 后面跟一个斜杠 / 用来和后面被替换的内容分开，这里0.01 表示选择第三行的0.01;
* 0.01后面再用一个斜杠，将其和替换后的数字分开(0.01 0.02 0.03 等)，表示将0.01替换为斜杠后面的内容;
* 再加一个斜杠，后面的g 代表 global ，意思是全部替换。

通过cat INCAR命令可知，并没有将INCAR文件中的0.01替换成0.02。如何更新？

* 先备份
* 先不加 -i 运行下sed命令，确保输出的是正确结果后，然后再加上 -i 运行.



sed –i 是sed 的命令和其附加选项， -i 表示直接对源文件进行编辑，也就是说编辑之后源文件被新文件替换掉

sed 批量将0.01到0.09中所有INCAR中的0.01替换成0.05。

图片包含 日历

描述已自动生成

文本

描述已自动生成

sed -i '3s/0.01/0.05/g' \*/INCAR

文本

中度可信度描述已自动生成

For循环结合sed：

实现每个文件夹中的SIGMA值与文件夹相同。

文本

描述已自动生成

这里我们用的是双引号 “ “ ，sed 命令中你会见到大部分都用单引号 ‘ ‘ 。但如果这里使用单引号，则所有的 0.01 都会被替换成i（单引号中的i 是纯字符)，因为单引号中的所有内容都会被当做字符来处理，也就是里面是什么就输出什么。使用双引号，则可以读取变量 $i 的值.

文本

描述已自动生成

这里单引号中的内容被原封不动地打印出来了。而双引号的话，则可以顺利地把变量调用起来。

说明：for i in \* : 指的是当前目录下所有的文件以及文件夹，本例中没有文件，只有从0.01， 0.02， 0.03 到 0.09 的文件夹；所以： for i in \* = for i in 0.01 0.02 0.03 0.04 0.05 0.06 0.07 0.08 0.09

### Part III

对KPOINTS的第四行操作，将1 1 1改成 2 2 2， 3 3 3 等。

图形用户界面, 文本

描述已自动生成

屏幕上有字

描述已自动生成

使用{}原因：避免i和后面的连在一起，导致调用失败，如：

文本

描述已自动生成

错误做法：

1.

文本

描述已自动生成

文件夹名字是三个数值，i/KPOINTS只引用了一次，导致sed命令对象不正确。

2.使用单引号：

for i in {1..6}; do sed -i '4s/1 1 1/$i $i $i/g' ${i}${i}${i}/KPOINTS ; done   
iciq-lq@ln3:/THFS/home/iciq-lq/LVASPTHW/ex04$ cat 333/KPOINTS -n   
 1 K-POINTS   
 2 0   
 3 Gamma  
 4 $i $i $i  
 5 0 0 0

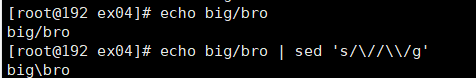
正确做法：

文本

描述已自动生成

转义字符：\，将反斜杠后面的一个变量变为单纯的字符串。

如果被替换的内容中含有 / , 直接输入则会被认为是分隔符,因此我们需要将其作为分隔符的作用去掉。怎么做呢? 输入\/ (一个反斜杠加单斜杠,中间没有空格),这样的话 / 就会被当成字符来处理



注意：要时刻备份自己的文件。

### Part IV

POSCAR批量处理：要创建不同大小的正方形格子，每个格子边长分别为8,10,12,14,16。



屏幕上有字

描述已自动生成

Seq命令：

1. seq + 选项 + 尾数

2. seq + 选项 + 首数 + 尾数

3. seq + 选项 + 首数 + 增量 + 尾数

文本

描述已自动生成

注意：

1. 这里你会想到之前我们使用的花括号 {}，如果间隔是1的话， {1..9} 和 seq 1 1 9 以及 seq 1 9 是一样的; seq 1 9 中省略的是 seq 1 1 9 中间的 1， 因为它是默认值。但是这里我们数字的间隔是2，使用花括号则不能实现我们的目的。
2. `seq 8 2 16` 的效果等于（seq 8 2 16）
3. 3,5s 的意思是：选择第三行到第五行中所有的8

Tree命令：

tree 来查看当前目录下的文件夹级别信息

tree-d：显示目录名称而非内容。

## Vasp任务提交（在自己的虚拟机上）

Vasp编译出现问题，可参考：<https://www.bilibili.com/video/BV16341117yw?spm_id_from=333.999.0.0&vd_source=c67ad0ff2b97c7927ad844cde6366fc6>

图片包含 表格

描述已自动生成

表格

中度可信度描述已自动生成

Linux 终端命令的末尾加上一个 & 符号表示将这个任务放到后台去执行。

. 逗号 (dot,就是“点”)

在 shell 中，使用者应该都清楚，一个 dot 代表当前目录，两个 dot 代表上层目录。

Out没有执行成功：

文本, 信件

描述已自动生成

删掉POSTCAR注释后运行的out：

表格

描述已自动生成

OUTCAR显示初出来时间：

表格

描述已自动生成

tail的命令用来输出某个文件最后的几行，如果想输出文件最前面的几行（通过-n 来实现），可以使用head文件：

文本

描述已自动生成

查看前5行：

文本

描述已自动生成

## Vasp输出文件

### Vasp输出文件1

文本

描述已自动生成

* 用POSCAR来存储模型的结构信息，vasp优化回产生新的结构，CONTCAR就是用来存储新结构的文件。
* 由于只有一个O原子，不存在优化结构，结果是CONTCAR和POSCAR的结构是一模一样的
* CONTCAR或者POSCAR中，每一行中只有空格可以用来分割不同的内容或者开头，绝对不要用tab。
* 在CONTCAR中，还多了一行内容（第11行），这一行主要描述的是体系中原子在xyz三个方向移动相关的信息。因为我们体系只有一个原子，所以也就只多了一行。
  + 以后当你在优化结构的时候，体系中有多少原子，就会多出多少行出来，但全部都是0。
  + 如果是分子动力学，就不为0，代表是分子移动速度
  + 如果你用dimer方法算过渡态的时候，那么这些行中是与过渡态结构相关的振动方式。

CONTCAR作用：

* 如果你的计算半路由于各种各样的原因给停掉了，需要续算，CONTCAR作用就展示出来了：
  + 第一个离子步没有算完，任务就挂掉了。这种情况，CONTCAR是不会更新的，我们再次用原来的输入文件提交一次就行了。
  + 我们的计算已经完成了大于或者等于1的离子步，但小于INCAR中设置的NSW的数值。这个时候CONTCAR的内容已经是离任务死掉最近的结构了。我们只需要将其复制成POSCAR，然后再次提交任务即可。具体操作如下：

mv POSCAR POSCAR\_0

mv OUTCAR OUTCAR\_0

cp CONTCAR POSCAR

**前面使用mv原因：**

因为我们的任务是从之前的POSCAR（标记为A）开始的，中途断掉了，直接把CONTCAR（标记为C\_A）复制成POSCAR（标记为B）的话，我们就会损失掉前面优化的过程。相当于我们直接搭建了这个B这个结构进行优化计算。而实际上，我们后面的计算的性质是续算，而不是从头直接计算。虽然结果大部分情况都是一样的。所以在将CONTCAR复制成POSCAR计算之前，我们要尽可能把任务停掉前的计算过程保存记录下来。这就类似于写实验记录本，我们要从开始搭建设备到开展反应一系列的详细信息都记录在上面。而不能从半路直接跳着写。这也是很多人在做计算的时候，任务死掉后续算所忽略的一个很严肃的问题。

* + 计算达到的INCAR中所设置的NSW的数值。比如设置的NSW = 1000，实际上跑了1000步，任务停下来了，也就是所谓的结构优化没有收敛。这种情况我们需要做的又有2个步骤：
    - 首先，要检查CONTCAR中的结构是不是正确的，如果结构跑乱了，体系中原子乱飞，有很大可能会导致不收敛的情况。如果结构混乱，从以下三点找原因：

1. 初始结构是否合理
2. POSCAR中的元素顺序与POTCAR是否一致
3. 是不是用的gamma点，然后把体系放开。
4. 前面三项没有问题：可能是因为你设置的NSW值太小导致的，或者体系是在是太难收敛

#### OSZICAR含义

优化：

1. 电子结构的优化： 可以理解为对某一固定的几何结构，迭代求解薛定谔方程来获得体系能量极小值的一个过程。这个迭代过程，每一次迭代求解都可以认为是电子结构的一个优化。（通常被大伙称为：电子步）
2. 几何结构的优化：可以理解为在电子结构优化的结果上，获取原子的受力情况，然后根据受力情况，调节原子的位置，再进行电子结构优化，获取新的受力情况，然后再调节原子位置，一直重复这样的过程，直至找到体系势能面上一个极小值的过程。（通常被大伙称为：离子步）

OSZICAR作用：用来记录优化过程一些信息的文件。这里的优化过程既包括电子结构，又包括几何结构。

OSZICAR示例：

文本

描述已自动生成

1. N 代表电子结构的迭代步数，通常被大家称为电子步。
2. E 代表当前电子步的体系能量(当前的自由能);
3. dE当前电子步和上一步体系能量的差值;
4. d eps 表示能带结构能量的变化;
5. ncg表示作用在波函数上的哈密顿量的计算次数
6. rms代表试验波函数的残差范数(即它们的近似误差)
7. rms（c）代表输入输出电荷密度之差。
8. DAV含义：Blocked Davidson algorithm的缩写简单点，就是一个电子迭代求解的自洽算法（对于选择哪种算法，可以在INCAR中通过参数ALGO设置，链接为https://www.vasp.at/wiki/index.php/ALGO），对于电子结构收敛的算法来说，默认的是DAV这个方法，也就是INCAR中： ALGO = N。
9. 看到一个vasp参数就要努力琢磨，能明白怎么回事。（不知道的参数不要写）
10. F前面的 1 代表几何结构优化的次数（也称为离子步的步数），本练习只有1步
11. F = 是体系的总能量, 与OUTCAR中 free energy TOTEN 后面的值相等；
12. E0 后面的能量对应OUTCAR中 energy (sigma->0)后面的能量
13. 通过OSZICAR获取体系的能量，也就是E0后面的那一项。很多人在使用VASP的时候，不知道该选择哪个能量，这里大师兄就告诉你：选择E0后面的即可。不管你有什么疑问，不管别人怎么跟你争论，都不要管，先老老实实记住：我们选E0后面的这个能量。随着你的学习，很多疑问自己就解开了。



### Vasp输出文件2

OUTCAR包含更加详细的内容，如优化结构、频率、声子谱，能带、DOS等计算

OUTCAR各个部分用很长的横杠分割（——————————-）：

文本

描述已自动生成

查看元素的价电子：

文本

中度可信度描述已自动生成

POSCAR内容：

文本

描述已自动生成

K点信息：查看k点个数：



INCAR参数：

文本

描述已自动生成

O原子的计算，就是一个离子步内，电子步的迭代：

文本

描述已自动生成

当迭代结束后，输出为：费米能级和能带信息。

文本

描述已自动生成

1. Band 1 对应2个2s电子
2. Band 2-4 对应的是4个2p电子。固体物理中，费米能级对应的是最高电子占据轨道的能量，也就是HOMO

查询费米能级：



使用OUTCAR能量：要使用E0后面的那个数值，在OUTCAR中与E0对应（相同）的是energy(sigma->0) = 后面的那个：

手机屏幕的截图

描述已自动生成

tail -n 1是为了保证我们取的是最后一步的能量

## 氧原子能量计算（对应ex6）

1. 氧原子轨道：

图片包含 图形用户界面

描述已自动生成

1. 计算结果查看：

文本

描述已自动生成

电子占据情况查看（OUTCAR）：

图形用户界面, 文本

描述已自动生成

Band1有2个电子，对应2S2

Band2-4Band 2-4 为三个简并态,每个能带上有 1.33 个电子, 对应的2p4

注意：在这里,电子的占据是不正确的:

因为 轨道上含有两个电子, 而不是把 2px上的一个电子平均分配到 3 个 2px轨道上;

此外,通过这个电子的占据情况,我们不能正确得到氧原子的磁矩信息。

如何解决：

1. 氧原子有磁性，考虑自旋极化，需要在INCAR加入ISPIN=2.

ISPIN 的取值为 1 或 2:

1. ​ 代表不考虑自旋极化,是VASP的默认值;
2. ​ 代表打开自旋极化该选项,计算中考虑。
3. 自旋极化什么时候考虑：
   1. 单原子的计算
   2. O2分子（基态为三重态）
   3. 自由基相关的计数
   4. 含Fe,Co, Ni 的体系
   5. 要计算的体系具有磁性：顺磁，铁磁，反铁磁等，要打开自旋极化。
   6. 当关注体系的电子性质时，且自己不知道加或者不加的时候，建议加上。
   7. 如果不知道是否要考虑自旋极化：
      1. 测试1：不加自旋极化，正常算，得到结果1
      2. 测试2：加上自旋极化，并在INCAR中添加： LORBIT = 11
      3. 测试2中OUTCAR的末尾会输出各个原子的磁矩信息：
         1. 如果体系中原子的磁矩不为0，那么需要考虑自旋极化，
         2. 如果体系所有原子的磁矩都为0，那么就不需要考虑自旋极化，而且此时，你会发现测试1 和 2的结果中：结构和能量是一样的。
4. 加入后重新计算：

**. 点号一个 .代表当前目录，两个** **..代表上层目录**



修改INCAR：

文本, 表格

中度可信度描述已自动生成

图片包含 文本

描述已自动生成

OSZICR信息：

表格

描述已自动生成

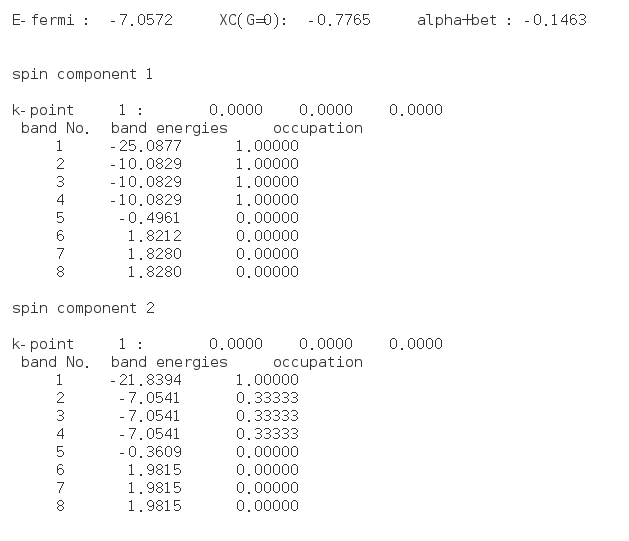
加入自旋极化后，电子迭代步数增加，电子分成两种，分别计算。

Mag为体系磁矩，与2一致。

查看OUTCAR能量：



查看体系电子占据情况：



出现问题：

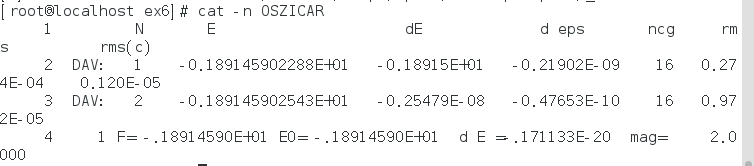
1. Spincomponent 1中,能带2-4中3个alpha 电子的能量是一样的。这不正确，因为其中一个2px轨道中的alpha电子已经成对,能量要比py 和 pz 低;描述失败!

2. spincomponent 2 中,另一个beta电子还是平均分配在了三个能带上面

原因：体系的高对称性

1. 修改POSCAR

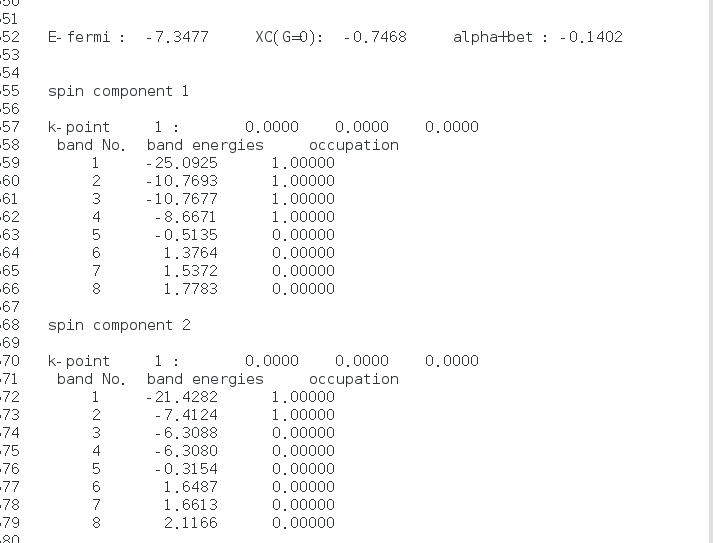
修改格子在三个方向的大小，降低对称性。



磁矩计算正确



能量降低



1. 在 α 电子(spin component 1)中, 能带1 是 2s中的 α电子; 能带2-3为两个简并轨道, 对应 py 和 pz 电子, 能带 4 位于 px 中的 α 电子。 这里能带 4 的能量应该比 2 和 3 要低, 但结果恰恰相反, 描述不合理;
2. 在 β 电子(spin component 2)中, 能带 1 是 2s 的β 电子, 能带 2 为 px 中的 β电子, 此时该 β电子占据了一个轨道, 而不是分布在三个 轨道上, 结果是合理的

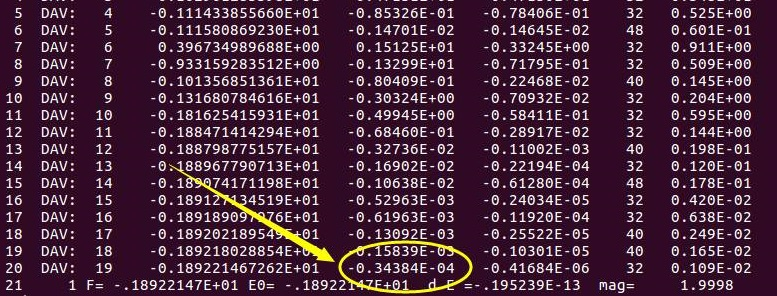
## 收敛点

1. Vasp计算过程：

图示

描述已自动生成

* + 它会猜一个初始的电子密度，然后据此计算体系的势能，求解KS方程，并给出体系的总能量以及对应的电子密度，也就是我们之前说的电子步的优化。
  + 将最新一步的结果与前面进行对比，当前后两者的差值达到我们预设的收敛标准时，计算结束。（预设的收敛标准：EDIFF和EDIFFG，EDIFF 控制电子步（自洽）的收敛标准。）

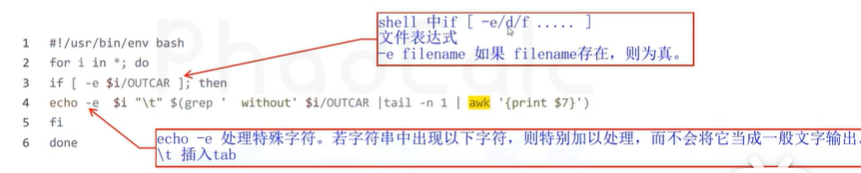
18步和19步能量差小于EDIFF默认值1

精度和计算代价：精度越高，计算代价越高。

图形用户界面, 文本, 应用程序

描述已自动生成

能量随着EDIFF的减小变化甚微



几何优化条件：EDIFFG

* 对于优化，我们可以使用力作为收敛标准，此时EDIFFG为负值。一般来说取值在-0.01到-0.05之间(-0.01对于力收敛来说已经是一个**很严格的要求了**)。
* 当然，对于较大的体系，我们也可以使用能量作为标准：此时，EDIFFG 为正值，一般取值范围在0.0001-0.001即可。

EDIFF，EDIFFG 是控制收敛标准的两个参数。前者负责电子自洽过程(单个离子步内)，取值为1E-4或者1E-5即可，没有特殊要求，不建议设置的太低。EDIFFG后者负责结构优化的过程(可正可负)。对于EDIFFG，默认值采用能量收敛，标准为:EDIFFx10。

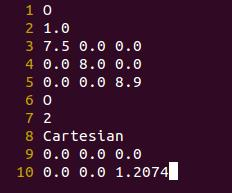
## O2结构搭建以及POTCAR注意部分

1. 氧气分子模型搭建
   * CRC hand book
   * NIST数据库
   * 谷歌

图示

中度可信度描述已自动生成

直线型氧气分子，一个原点在原点位置，其他三个方向随便找个坐标就可以。



氧原子数目由1变2

第10行添加第二个氧原子的坐标

1. POTCAR

图片包含 文本

描述已自动生成

O原子写了两遍，POTCAR就要对应与2个。如果不对应2个，可以查看out（不是OUTCAR），可以发现会有报错信息：

文本

描述已自动生成

POTCAR与POSCAR元素不对应。

解决问题：

* POSCAR：把O O换成O，氧原子数目改为2，POTCAR保持不变
* 变POTCAR，含有2个元素的POTCAR

文本

描述已自动生成

1. 多元素POTCAR制备

文本

描述已自动生成

## O2单点计算和优化结果分析

1. O2的单点计算

单点计算不优化结构，直接算能量。（几何结构计算前后不发生变化）

计算O2分子，查看OSZICAR:

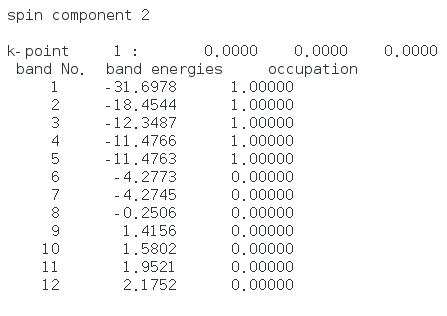
表格

描述已自动生成

查看氧气分子的轨道：

表格

描述已自动生成



第一个图能带5时对应于σ2pz，能量应该低于band3与4（能带3和4 是简并的，应该是π(p2x）和π(p2y） 轨道中的α电子），说明计算有问题。

图表, 图示

描述已自动生成

1. 优化分子结构

优化分子结构所需的关键词：IBRION，一般的三个选择：

* + IBRION=3：你的初始结构很差的时候
  + IBRION=2：共轭梯度算法，很可靠的一个选择，一般来说用它基本没什么问题。
  + IBRION=1：用于小范围内稳定结构的搜索。

如果体系不收敛，首先检查结构的物理化学意义是否正确，然后检查IBRION参数。

在INCAR中加上IBRION参数（IBRION=2），查看运行结果：

文本

中度可信度描述已自动生成

文本

低可信度描述已自动生成

能量和前面没有变化，没有优化，缺少NSW参数。

**NSW**：控制几何结构优化的步数，也就是VASP进行多少离子步。

NSE设置：

* + 它必须是大于等于0的整数
  + 一般来说，简单的体系200步内就可以正常结束
  + 不知道什么时候收敛，初始结构很差，或者设置了很严格的收敛标准，那么你就要增大一下NSW的取值了，比如NSW=500或者更大。

设置NSE=10重新计算：

文本

描述已自动生成

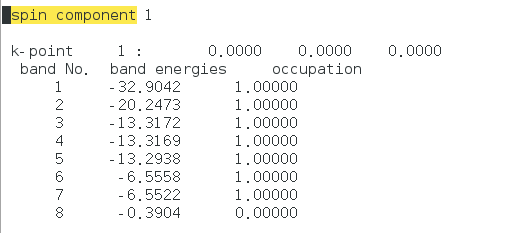
表格

描述已自动生成

EDIFFG控制此过程优化步骤的结束，可以看到，结构优化进行了3步便停止了（如果你设置了NSW=1000，那么也是3步后就结束）。

**注意：**EDIFF:控制电子步，电子结构优化；EDIFFG:控制离子步，几何结构优化。

查看OUTCAR ：



表格

描述已自动生成

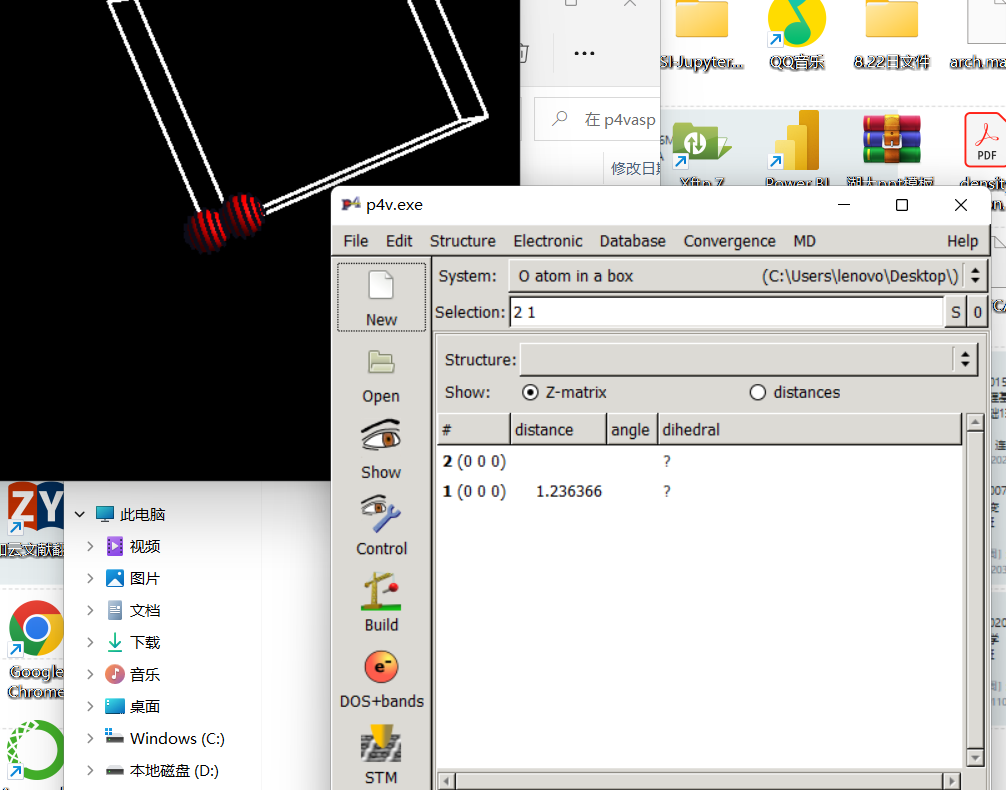
3为σ2pz，4和5为π(p2x）和π(p2y)轨道中的α电子。

β电子并没有发生改变。

查看CONTCAR:

是VASP的一个输出文件，它包含了VASP计算中最后一步几何优化的结构信息，也就是优化完的结果。

1. 通过坐标计算
2. 可视化软件



实验值为1.2075 Å，VASP计算结果为1.23 Å。两者之间的差值为： 0.0289 Å，偏差为：(1.23-1.2075)/1.2075 = 2.26%。对于理论和实验之间的偏差，如果小于5%，我们一般可以认为吻合的很好。

## VASP中MAGMOM的简单设定

1. MAGMOM的简单设定
   * 通过MAGMOM我们可以**指定体系中原子的初始磁矩**。通俗点说，就是给VASP指条计算的明路。对于复杂体系来说，合理的初始值可以加快计算速度，并保持计算结果的正确性。但是对于一些**简单的磁性体系**，我们可以直接使用ISPIN=2, MAGMOM不必进行设置。
   * 如果我们已经知道磁矩是多少（比如O2），可以直接写上去值，也可以差不多写一个； 但如果不知道的话，就需要根据自己的假设，猜一个初始值了。

图形用户界面, 文本, 应用程序

描述已自动生成

MAGMOM的默认值是原子的个数乘以1.0（原子个数在前面，原子磁矩在后面）

氧气分子几个错误示例：

MAGMOM = 2 \* 1（\*有空格）

MAGMOM = 1 \* 2（\*有空格，前后颠倒过来了）

MAGMOM = 1.5\*2（前后颠倒过来了）

MAGMOM = 3 3是否正确？正确，MAGMOM并不按照我们已经知道的数据去输入，就算设置很大，vasp也会自动矫正过来。（注意：如果你不知道体系的磁矩是多少，可以根据原子所处的化学环境， 根据成键情况，大体推测有多少个未成对电子，然后将未成对电子数目\*1.5即可。）

# 论合理结构的重要性

## 论合理结构的重要性（一）

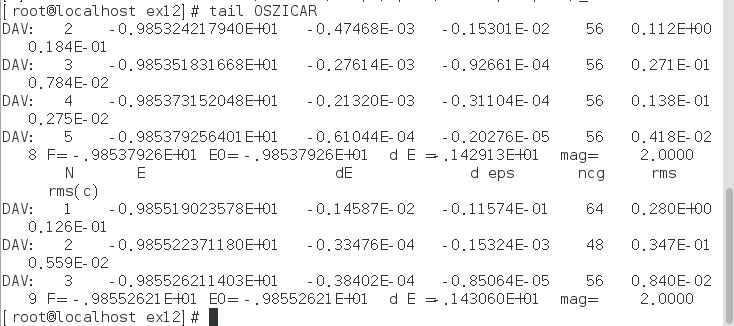
### 测试bad结构

INCAR和POSCAR相同，只是将初始键长改为1.5

文本, 信件

描述已自动生成

对比OSZICAR：

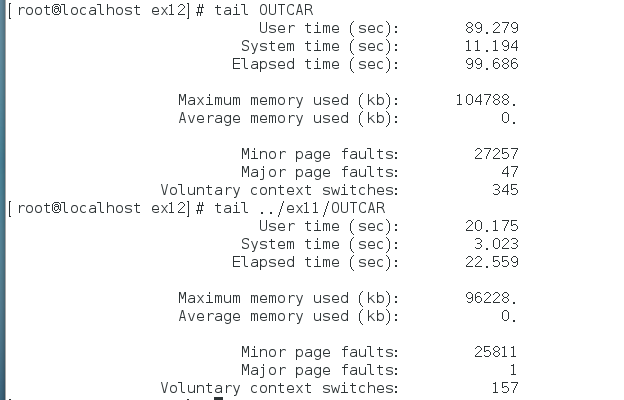


表格

描述已自动生成

在POSCAR设置初始键长为1.5，可以看出9个离子步收敛，在实验值的O2键长，3个离子步收敛。（INCAR设置NSW=10，优化的离子步<设置的步数，结构收敛达到收敛标准。）

查看OUTCAR信息：



### 可能出现情况

如果结构优化的步数等于NSW设置的步数，有可能说明没有达到收敛标准原因有以下几种可能：

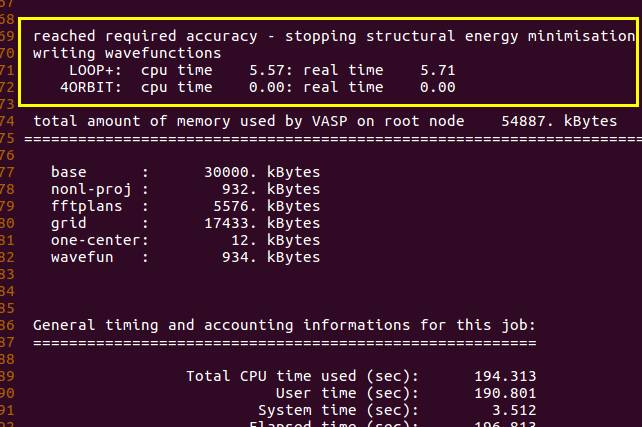
1. NSW设置的值偏小，如设置NSW=6
2. 初始结构不合理，计算需要很多离子弛豫过程
3. 设置收敛标准不严格
4. 结构复杂，每一离子步收敛很困难。

或者是进行到NSW设置步数，计算正好收敛了。（概率极低）

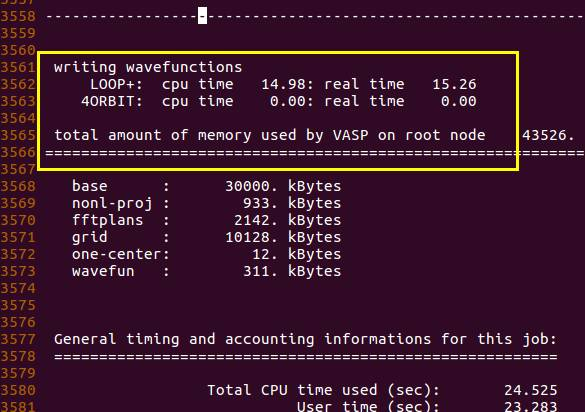
判断上述特殊情况的方法：

1. 查看OUTCAR：

OUTCAR（NSW=10,收敛）



OUTCAR（NSW=2）



可以看出，当出现这句话时：reached required accuracy - stopping structural energy minimisation，计算已经收敛。

可以通过grep提取信息，提取信息原则是：精确匹配并提取最有价值的信息！。如下图：



### linux中添加自定义命令alias

1. Linux alias设置指令的别名命令详解

语法： alias 别名 =指令名称

作用：linux alias 是命令的一种别称，输入 alias 输出形如这样的结果：alias vi="vim"

输入vi后，被自动定向到vim这个命令了。alias就是用来简写命令的。

补充说明：用户可利用alias，自定指令的别名。若仅输入alias，则可列出目前所有的别名设置。　alias的效力仅及于该次登入的操作。若要每次登入是即自动设好别名，可在.bashrc中设定指令的别名。

## 论合理结构的重要性（二）

### 不同初始结构对结果影响

文本

描述已自动生成

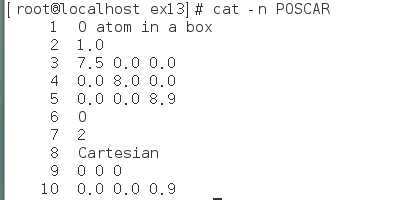
文本

描述已自动生成

可以计算出两者计算结果基本相等。

### 另一个bad结构的计算

键长改为0.9，查看INCAR：

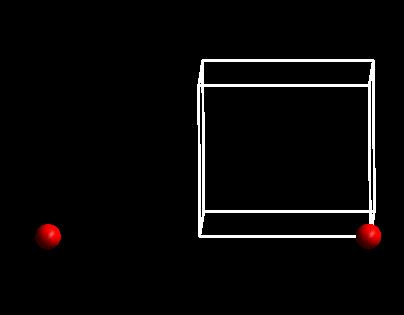


INCAR，KPOINTS，POTCAR等不变，计算，得到结果为查看：

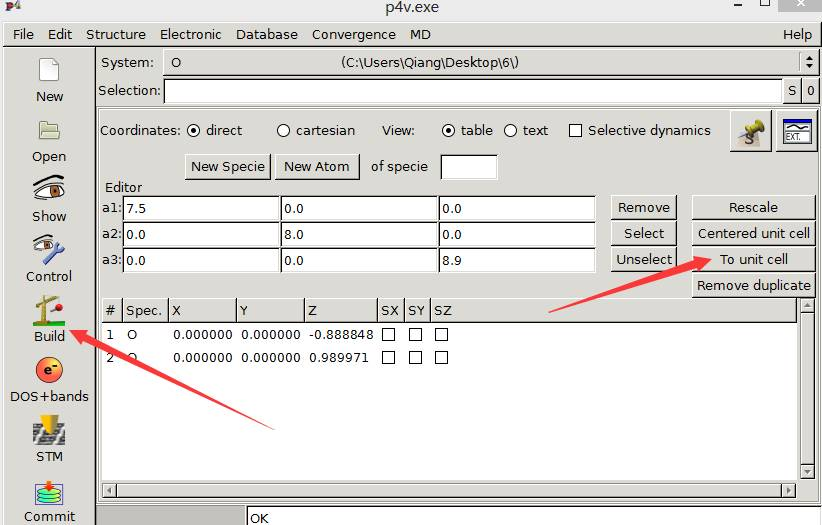
文本

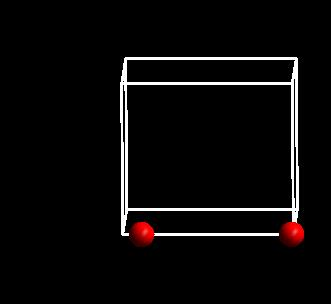
低可信度描述已自动生成

使用p4vasp查看，可以看出两个原子离得很远。



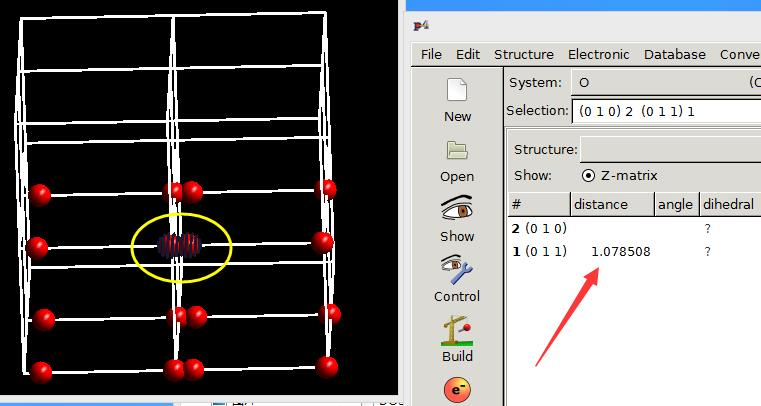
这是由于周期性原因，可以进行调整。





两个原子的距离为7.82，实际键长为8.9-7.82=1.08。

这是因为体系是周期的，将格子进行三维重复，效果如图：



图形用户界面, 文本, 应用程序

描述已自动生成

不知道正确结果判断计算是否成功：

文本

描述已自动生成

## 论合理结构的重要性（三）

NELM控制每一个离子步中的电子步的最大值。，vasp默认值是60.

图形用户界面, 文本, 应用程序

描述已自动生成

### 电子步数不收敛的解决办法

解决办法：

1. **首先检查自己的初始结构是不是合理的（最重要的一步）**
2. 如果第一个离子步内，电子步（SCF）计算不收敛，可以增加NELM的值
   1. 对于普通体系，设置NELM=100
   2. 电子结构难以收敛体系，设置NELM=200
   3. 增加NELM依然不收敛，可以改变AMIX,BMIX
3. 第一个离子步的电子步收敛，但后面不收敛，能量变得非常大，首先要检查结构，调整结构在提交任务
4. 如果前面几个离子步中电子步都不收敛，且能量变化正常，可以尝试着让任务再多算几步，后面跑着跑着可能就收敛了。
5. 如果跑了很长时间，每一个离子步中的电子步都不收敛，可以尝试着换一个更加稳定的电子步迭代算法（ALGO参数）。这种情况：ALGO = ALL 结合 NELM = 200 可以解决大部分的问题。

注意：但需要注意的是：ALGO= ALL 这个算法虽然稳定，但比较耗时。自己要权衡一下。可以尝试着两步走的战略：step1）先用ALGO =ALL的办法算一个单点，保存WAVECAR。step2）然后将ALGO改回原来的，读WAVECAR继续优化。

### 设置键长为0.9，计算失败原因：

文本

描述已自动生成

**IBRION = 2 时，对POTIM的依赖性很强**，因此我们计算的时候要设置一个合理值。在我们的计算中，由于初始的原子间距离很小，第一步计算时，得到的原子间的初始排斥力很强，第二步中，VASP默认的POTIM值是0.50，前面两步导致了尝试步中离子的移动过大，以至于后面没有办法矫正回来，最后导致O2分子计算出错。

### 更改POTIM重新计算

当氧气键长为0.9时，设置POTIM为更小的值，POTIM=0.1，保证vasp离子移动很小。INCAR文件如下：

文本

描述已自动生成

查看计算结果：

文本

描述已自动生成

# 谁偷走我的机时

## 谁偷走我的机时1

计算细节，提高计算速度。

* 体系的磁性

考虑自旋（ISPIN=2），vasp计算将电子分为两部分：α电子和β电子。

如果MAGMOM为负值，应该写为MAGMOM=10\*-2（有10个原子，每个原子的初始磁矩为-2）

* 对称性

体系对称性降低，会增加计算量

* SIGMA的取值

设置较小的的SIGMA会导致收敛变慢

* + 半导体和绝缘体：

1. K点小于4 的时候，用ISMEAR=0, SIGMA取值小一些；比如SIGMA=0.05; 此时用ISMEAR = -5 会出错；
2. K点数目大于4的时候，可以使用ISMEAR= -5；
   * 金属体系

ISMEAR一般用ISMEAR=0 或者整数1，2即可。SIGMA =0.1 足够。

* + 合理的初始结构

如果你的初始结构不合理导致的计算出错，首先应该去调整结构，最后才是去调节参数

* + EDIFF

EDIFF 控制了电子迭代的收敛标准，如果你设置的标准比较严格，则每一个离子步需要更多的电子步数，需要的时间也会随着离子步的增加成线性关系增长。

* + EDIFFG

EDIFFG控制了结构收敛的标准，同样严格的标准需要更多的结构优化步骤来实现。

## 谁偷走我的机时2

本节主要考虑ENCUT的影响。前面我们一直在说EDIFF和EDIFFG对计算时间的影响。为了给大家一个感性的认识，现在我们用O2分子的计算作为测试例子，将结果展示出来。

1. 调节EDIFF和EDIFFG
   * 没有提高精度

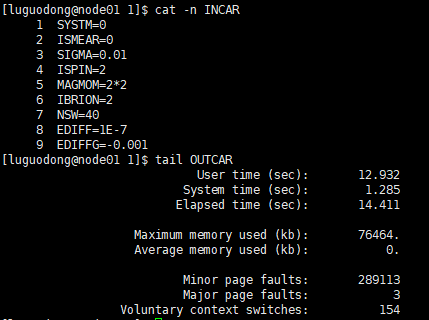
文本

描述已自动生成

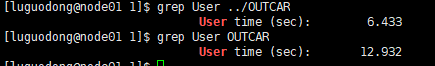
文本

描述已自动生成

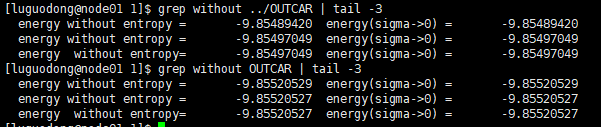
* + 提高精度：



通过grep查找user time命令：



查看改变精度后的能量变化：



可以看出改变收敛对于我们计算体系能量变化很小，一般来说，结构优化的时候，EDIFF=1E-5, EDIFFG =-0.01至 -0.03 都是被认可的。

1. ENCUT测试：

ENCUT默认：POTCAR 文件中最大的 ENMAX，ENCUT 以 eV 表示平面波基的截止能量。

* + ENMAX简介：
    - 每个元素的赝势文件（POTCAR）规定了元素价层电子与对应的最大截断能ENMAX。
    - 如果在参数设置文件（INCAR）中设置了ENCUT的数值，程序将按照实际设定的ENCUT进行计算；
    - 如果没有指定ENCUT，程序将默认ENCUT=ENMAX（注意：当PREC = High时, ENCUT = 1.3\*ENMAX）。

**制作ENCUT测试模板：**

文本

描述已自动生成

Sed 命令：

工作原理：sed会一次处理一行内容。处理时，把当前处理的行存储在临时缓冲区中，成为"模式空间"，接着用sed命令处理缓冲区中的内容，处理完成后，把缓冲区的内容送往屏幕。接着处理下一行，这样不断重复，直到文件末尾。文件内容并没有改变，除非你使用重定向存储输出。

i：插入， i 的后面可以接字串，而这些字串会在新的一行出现(目前的上一行)；

"$"表示行尾的意思，也就是每一行的结尾位置，很好理解，和"^"正好相反。

a ：新增， a 的后面可以接字串，而这些字串会在新的一行出现(目前的下一行)～

s ：取代，可以直接进行取代的工作哩！通常这个 s 的动作可以搭配正规表示法！例如 ‘s/old/new/g ’

**快速制作测试任务：**



如图，即可快速制作测试任务。$i表示调用这个变量。

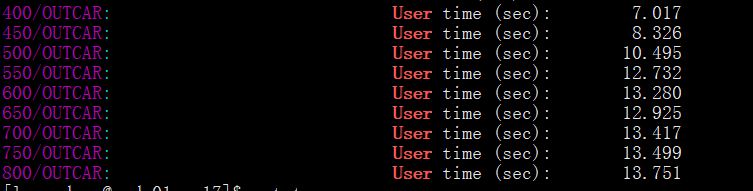
**批量提交任务：**



返回原来的目录下（cd $OLDPWD）

**批量查看结果：**

* + echo –e 后面加上”\t”后（双引号），可以直接输出 tab，方便导入excel；
  + $( )与` `（反引号）都是用来作命令替换的。命令替换与变量替换差不多，都是用来重组命令行的，先完成引号里的命令行，然后将其结果替换出来，再重组成新的命令行。



**总结**：

1. ENCUT值越大，计算的越精确，花费的时间也就越多；
2. 算出来的数值必须要应用到我们的物理化学概念中
3. ENCUT的取值与体系中所有的元素有关：
   1. 体系中含有不同元素的时候，查看这些元素POTCAR中的ENMAX值，找出最大的那个
   2. ENCUT的最小值为所有元素中ENMAX的最大值！也就是说，找到最大的ENMAX（max）， ENCUT值大于等于ENMAX（max）。

提交补充知识：

* + - 投递任务：

qsub -cwd -l vf=\*G,p=n,h=node -q queue \*.sh

-cwd：使用当前工作目录，SGE的运行日志会输出在当前目录。

-l：申请所需的资源。vf=\*G设置任务预计使用的内存大小（一般设置的值要稍微大一些，否则容易让计算节点挂掉），p=n设置所需的CPU数，h=node指定任务投递的节点名称。日常使用时，并不是每次都要设置这三个参数，大家可以根据自己的需要自行删减。

-q：指定任务投递的队列名称。

\*.sh：需要投递的任务，建议在脚本前加上完整路径。

* + - qstat：

## 查看任务状态

qstat

## 查看指定任务状态

qstat -j job-ID

## 删除指定任务

qdel job-ID

job-ID：任务号

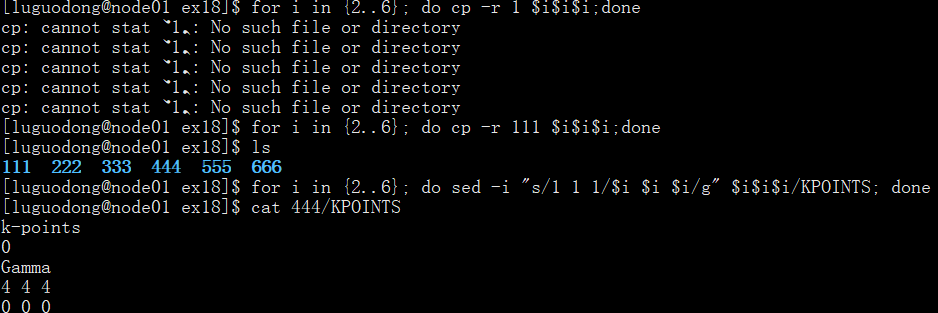
state：任务状态。一般会有以下几种情况，"qw"代表任务在等待执行，"r"代表任务正在执行，"s"代表任务暂时挂起，"dr"代表任务所在的计算节点挂了，需要联系管理员重启节点。任务还有其他状态，大家遇到时可以查询一下。

queue：任务所在的队列及计算节点。Job id 211是给提交的任务分配的任务号，S（常用状态：R代表运行，Q代表排队，E代表正在退出，H代表挂起，C代表运行完毕）

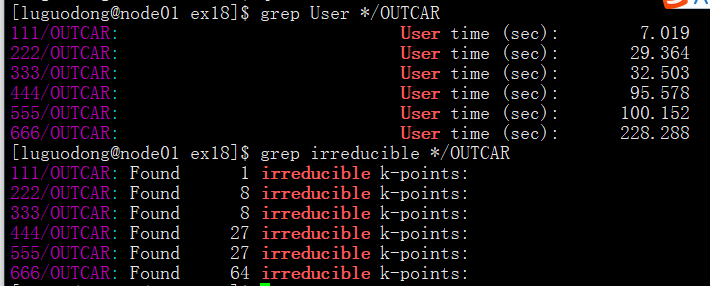
* + - pbsnodes：列出集群中所有节点的状态和属性

## 谁偷走我的机时3

准备模板和测试文件：



**查看测试结果**：



1. 随着K点增多，计算时间增加
2. 2 2 2和3 3 3计算中K点数目是一样的，444和555具有同样的K点数目。

不同K点对能量的影响：

文本, 聊天或短信

描述已自动生成

可以看出111和666的时候，氧气能量差别很小，因此算气相分子的时候，111足够了。

对于其他材料，K点选择：

选择K点规则：乘机ka，（k是k-points在任何方向上的数量，a是在这个方向的基矢）应该是：

1. ka~30，for d区金属
2. ka ~ 25 Å，普通金属
3. ka ~ 20 Å，半导体
4. ka ~ 15 Å，绝缘体

注意：奇数网格和偶数网格有很大不同，如果你仅仅包含在奇数网格中的gamma点。然而，如果像石墨烯这种特殊物质，布里渊区的重要的部分在K而不是在G，并且你应该乘以3倍的k点，14没有12精确。

总的来说：K点具体用什么数值，可以从以下三点考虑：

1. 通过测试不同K点对体系能量的变化
2. 查找参考文献的取值
3. 不同K点之间的数据不能混用。比如计算CO在一个 (3x3) Cu(111)表面上的吸附能：

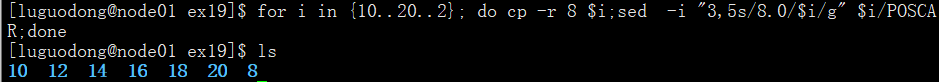
等号后面的前两项，必须要用同一个K点下计算出来的能量，如果ECO+slab用5x5x1 的K点， E(slab) 采用 3x3x1的K点能量，得出的结果必然是错的。

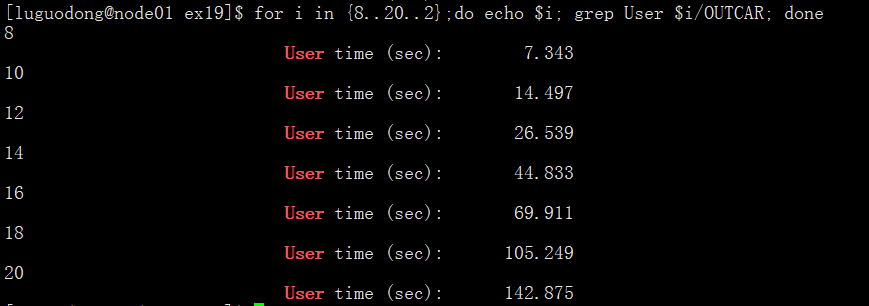
**Vasp官方提醒**：

* 体系中ENCUT的取值不统一
* 采用不同KPOINTS计算出来的结果

## 谁偷走了我的机时4

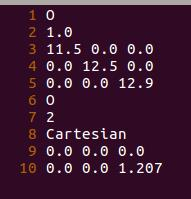
**晶格尺寸的影响：**





随着格子增大，计算时间也随之增大。

**体系的对称性对计算速度的影响**：



修改对称性之前，12文件夹中OUTCAR的USER时长是26.539，修改之后，是28s。

**体系对称性与K点对称性的关系**：

体系的对称性与K点对称性的匹配问题，尤其是对于hexagonal的结构来说，必须要使用 gamma centered points. 也就是第三行的第一个字母必须为G或者g。

体系的对称性与K点对称性的匹配问题，尤其是对于hexagonal的结构来说，必须要使用 gamma centered points. 也就是第三行的第一个字母必须为G或者g。如果你不确定自己的体系，直接用G就可以了。

官网原话：我们强烈建议仅对六边形格使用以 Gamma 为中心的格网。我们执行的许多测试表明，与标准Monkhorst Pack网格相比，居中网格的能量收敛速度明显更快。在第三行中使用“M”设置生成的网格实际上没有完整的六边形对称性。

## 谁偷走我的机时5

这节讨论并行的影响，并行是多个节点计算同一个任务。

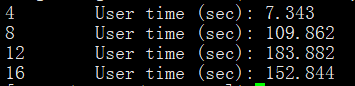
**测试：**

使用8x8x8的格子，K点是333，ENCUT=400，设定每个节点使用4个核，每个核有两个进程。



4，8，12，16，20，24代表使用1，2，3，4，5，6节点计算，每个节点使用4个核。在这里，每个节点可以看成1匹马，4个核可以看作马的四条腿。每个核的两个进程可以看做每条腿的上下两部分。

结果分析：（batch限制4个节点）



增加核数反而成了计算的负担。这是因为并行计算的时候，不同核之间的数据传输浪费了大量的计算时间。这好比是你多安排了几匹马来拉车，但马儿们却彼此交流，各跑各的，不愿意拉车了。

那并行是完全无效的么？不是，需要两个参数才能保证有效。

NCORE：控制多少个核同时计算；

NPAR：如何把计算任务分配到计算资源上面计算

它们之间的关系是：NCORE= 计算使用的核数 / NPAR

**总结**：

1. 使用NCORE以后，单节点运行也加快了；（单匹马儿被你调教的更听话了）
2. 加入NCORE和NPAR参数后，计算时间明显加快了。
3. 如果你想用多个节点计算，NCORE或者NPAR，不要忘了加（二者选其一）
4. 可以肯定的是：NPAR和NCORE的乘积就是我们计算所用的核数，这一点大家要记在心里。
5. 在本测试中（这是前提！！），增加节点个数并未实现 1+1 = 2 的效果，1+N = 2 的效果也没有实现。但在多个节点的计算中，加入NCORE或者NPAR，节点数越多，效果越明显。
6. NPAR的取值可以设置为节点的数目，（默认值为计算的核数！）
7. NPAR实在不懂的话，直接设置NCORE=单节点的核数，单节点的核数/2，单节点的核数/4
8. 个人使用经验是：NCORE = 单个节点核数 / 2 的时候，运行最省时间，设置也最方便。

# 乙醇

## 乙醇气相分子的优化

INCAR(设置了IBRION和NSW)

文本

低可信度描述已自动生成

KPOINTS:

文本

描述已自动生成

POSCAR:

图片包含 文本

描述已自动生成

POTCAR根据赝势文件生成。

## 乙醇分子的震动频率1

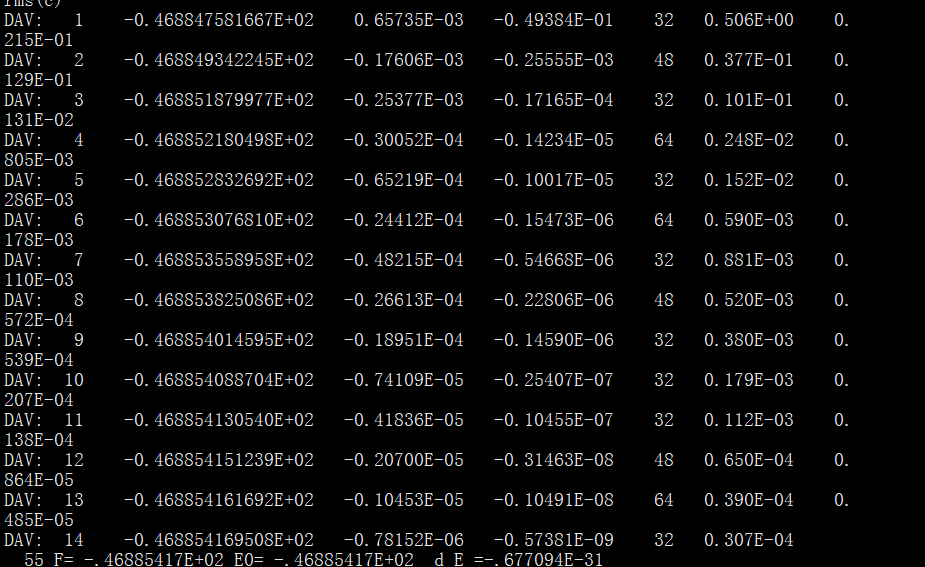
频率计算的作用：

1. 确定结构是否稳定
2. 看振动方式和大小，用来和实验对比
3. 反应热，反应能垒，吸附能的零点能矫正
4. 确认过渡态（有一个振动的虚频）
5. 热力学中计算entropy，用于计算化学势，微观动力学中的指前因子和反应能垒。

计算步骤：

1. 进行结构优化，获取稳定结构
2. 将原来的CONTCAR复制成POSCAR
3. 修改INCAR
   * 当 IBRION = 5或 IBRION = 6时，VASP 用有限差分法计算总能量相对于离子位置的二阶导数，构造动力学矩阵并对角化，在 OUTCAR 文件中报告系统的声子模式和频率。IBRION = 5不使用对称性，而 IBRION = 6使用对称性来减少位移的数量。
   * POTIM用一个更小的值，我们这里用的 0.02，默认值是 0.015
   * NFREE(对于IBRION=5/IBRION=6): NFREE = 2使用中心差，即每个离子在每个方向上被一个小的正负位移所移位.
   * EDIFF也要设置一个严格的值（频率计算时，默认值为1E-6

## 乙醇分子的振动频率2



一共走了55步，原因如下：

1. 乙醇分子为CH3CH2OH,每个原子有3个自由度，共9个原子，一共27个原子。
2. 设置NFREE=2,即在+POTIM和-POTIM移动并计算，27✖2=54
3. 55-1=54，去掉第一个离子步计算。

**固定原子**：

图表, 雷达图

描述已自动生成

可以知道atom1和atom9是羟基的o和H

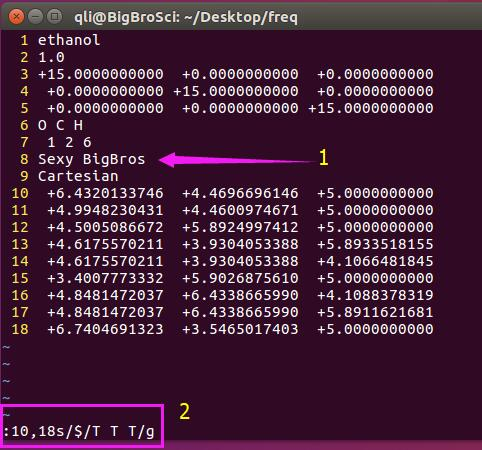
电脑屏幕的照片

中度可信度描述已自动生成

默认的是所有的原子在x,y,z三个方向上均可移动。但很多时候，我们只需要振动感兴趣的原子或者某一特定的方向就可以了，也就是说选择性计算频率。比如我们只想算乙醇的羟基振动峰，那么其他的原子就可以固定。VASP可以通过设置POSCAR来实现这个功能:

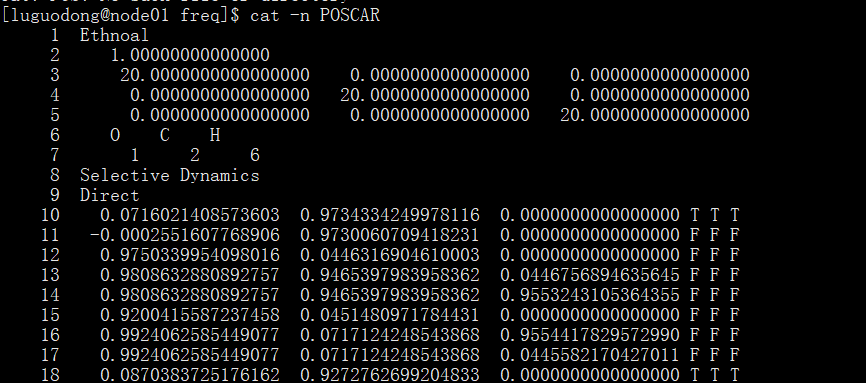
1. 加入 Select 的关键字母S，并在坐标后面全部加上 T T T:

我们需要在每一行的坐标后面加上 T 或者F表示允许和禁止移动。这里我们需要加三个T或者F，表示在 x,y,z 三个方向上选择性固定原子的移动，三个方向都允许：T T T，三个方向上都不允许: F F F



1. 把OH之外的原子全部固定住;

11-17行所有T替换成F，最后效果如图所示：



# 块体计算

## Bulk的单点计算（Fe单胞）

1. Vasp文件

LREAL参数：LREAL 确定投影算子是在实空间中求值还是在倒数空间中求值。默认false

文本

描述已自动生成

文本

描述已自动生成

1. 查看计算结果：

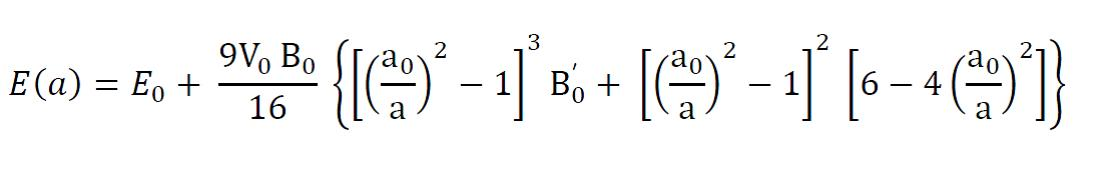
磁矩信息：

日程表

描述已自动生成

## 晶格参数的确定（Birch-Murnaghan状态方程）

1. 通过BM方程获取晶格



此方程可以写成如下形式：，

随后对x求导，解出来

1. 获取拟合数据

我们需要的就是获得的和值。可以设置一系列不同的晶格常数进行计算，得到对应的能量，然后将这些数据带入下面的方程中：进行拟合。

* + 使用分数坐标：

如果我们改变了晶格常数，那么Fe原子的坐标也要发生相应的变化。此时，用分数坐标更加方便。

* + 获得分数坐标后，批量对晶格常数a,b,c进行操作，或者对放缩系数操作。

对放缩系数操作：



批量提交命令：

图形用户界面, 应用程序

中度可信度描述已自动生成

* + 数据提取：

图形用户界面, 文本

中度可信度描述已自动生成

1. 使用脚本

获取参数：

文本

描述已自动生成

查看脚本内容：

文本

描述已自动生成

## 晶格参数的确定（直接优化晶格常数）

通过ISIF和ENCUT=XXX确定晶格参数

1. ISIF参数

作用：ISIF 确定是否计算应力张量，以及在松弛和分子动力学运行中允许哪些主自由度发生变化。

表格

描述已自动生成

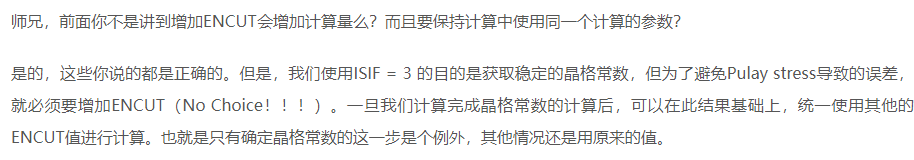
ISIF = 3 的时候，晶胞中原子的坐标，晶胞形状，以及体系都随着优化的过程发生变化。

1. ENCUT取值

如果计算时体积发生了变化，我们需要增加ENCUT的值，max(ENMAX) 的意思是，如果有N个元素，取最大元素的ENMAX值。

官网说设置到130% \* max(ENMAX)，一般来说就可以了。但很多人还是不放心，那我们就继续疯狂一些，设置到600 或者700。本人一般习惯用600。

注意：



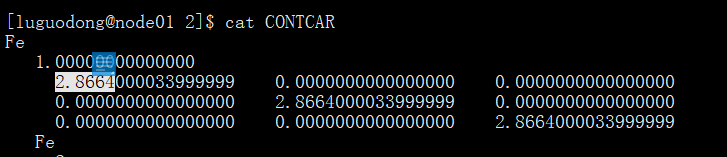
1. Pulay stress

当晶胞体积发生变化时，由于计算时的平面波基组不全（设置的ENCUT不够大）使得计算时的收敛不是绝对的，进而导致应力张量不对，最终导致体系的形状，体积以及能量不正确。

减小 pulay stress方法：

* + 体积不变，采用一系列的晶格常数计算，然后拟合得到准确的晶格常数；
    - 在LVASPTHW的拟合流程中，拟合出来的是BM方程， BM方程的最高次数是3，也就是x的三次方。
    - 如果直接用晶格常数去拟合，则是6次方的关系。
    - 如果你用体积去拟合BM方程的话，最高次数是2
  + 既然是因为ENCUT不够导致的，那么我就使劲增加ENCUT来消除（上图中unless后面的那句）。借用维基百科的一句话：Pulay stress can be reduced by increasing the energy cutoff. 所以，**当有人用ISIF= 3计算，而不去设置ENCUT的时候，尽情地去嘲笑他傻X吧！！！感觉不过瘾，还可以继续吓唬他说后面的计算也有问题**。

1. 计算结果：



可以看出结果为2.8664,与BM相差很小。

补充：\* cmd << text

从命令行读取输入，直到一个与text相同的行结束。除非使用引号把输入括起来，此模式将对输入内容进行shell变量替换。如果使用<<- ，则会忽略接下来输入行首的tab，结束行也可以是一堆tab再加上一个与text相同的内容，可以参考後面的例子

# DOS（能量态密度）计算

1. KPONITS

K点数目：计算DOS的时候，需要更多的K点数目

K点数目的选取：对于DOS计算，K\*a=45可以满足要求

1. NEDOS

NEDOS这个参数在DOS图的质量上面也有着很重要的作用。比如我们的DOS能量区间范围（DOS图的横坐标）为：[-10 eV，10eV]，VASP默认的将这个能量范围分成301点，然后作图。301也就是默认的NEDOS的取值。如果我们设置的NEDOS值够大，那么DOS区间就会被区分地越精确。NEDOS的取值一般来说：

* + NEDOS = 3000左右就足够好了。太大也没什么意义
  + NEDOS越大，VASP输出的DOSCAR，vasprun.xml文件也就越大，占用存储空间。
  + 经常有人抱怨说自己的DOS图有很多尖锐的峰，可以尝试着通过增加NEDOS这个办法来解决。

1. ISMEAR一

. ISMEAR = -5 的时候（Blöchl修正的四面体方法），我们可以得到一个非常平滑的DOS图

注意：设置ISMEAR = -5 的时候，如果K点数目K点的数目小于等于4 ， 计算会出错，得到如下的错误结果：**VERY BAD NEWS! internal error in subroutineIBZKPT:Tetrahedron method fails for NKPT<4. NKPT= 1**

**ISMEAR=-5**适用于所有体系的DOS计算

非DOS计算的时候：

* + - 对于金属体系来说，结构优化的时候不能使用ISMEAR= -5（注意：是优化结构的时候不能用！），这是因为四面体方法不能很好地处理费米能级处的电子占据情况，导致算出来的力会有一定百分比的误差。所以，对金属体系结构优化的时候，ISMEAR > = 0
    - 对于半导体和绝缘体的体系，ISMEAR>0 是不可以的。只能ISMEAR<=0
    - 四面体方法（ISMEAR = -5）不适合计算能带（对所有的体系来说的）
    - 使用ISMEAR= -5 的时候，SIGMA的取值没有影响，如果不放心，取SIGMA = 0.01

1. ISMEAR二

如果体系很大，只能适用gamma点来算，ISMEAR = -5的时候，肯定会出错，但服务器不给力，不能增加Ｋ点的时候，怎么办？

* + 对于所有的体系（K点数目小于4也可以）：可以使用ISMEAR = 0；SIGMA = 0.01。对于大部分的体系都能得到理想的结果。原则上来说，使用GS方法的时候(ISMEAR=0)，SIGMA的数值要测试下，保证entropy T\*S这一项平均到每个原子上小于0.001 eV也就是1meV。不想测试的话，直接用个很小的值，比如这里我们说的：SIGMA = 0.01。
  + 对于金属体系来说，也可以使用ISMEAR = 1；SIGMA = 0.01。SIGMA取值太大，计算出来的能量可能不正确；SIGMA取值越小，计算越精确，需要的时间也就越多。值得注意的是：这里我们使用的0.01已经很小，没有必要设置的再小。因为对于金属体系，使用MP方法(ISMEAR=1..N)时，SIGMA= 0.10 差不多就足够了。官网给的参考值是0.20