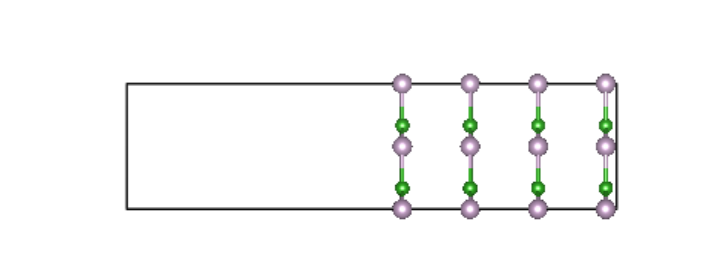
Atom\_constrain.py 使用教程：

在对slab 结构进行计算时，有的时候为了减少计算量，可以考虑把slab 结构中某些层进行固定，目前固定的方法有很多，比如用MS 进行手动固定，vaspkit-402 功能或许楠的POSCARtoolkit.py 脚本等都可以实现不同方式的原子层固定，但目前vaspkit-402 功能对POSCAR 中缩放因子不为1 时，暂时有些bug(近期会修复)，且vaspkit-402功能会把POSCAR 坐标自动转换为笛卡尔坐标，POSCARtoolkit.py 功能虽然比较齐全，但只能针对原本POSCAR 已经被固定过，也就是必须要有 Selective Dynamics 这一行,否则即便被固定，新生成的POSCAR 也会缺失“Selective Dynamics”字样，需要进一步手动添加该行，因此，本脚本克服了上述问题。

脚本主要功能：

实现两种方式对POSCAR 进行选择性固定：

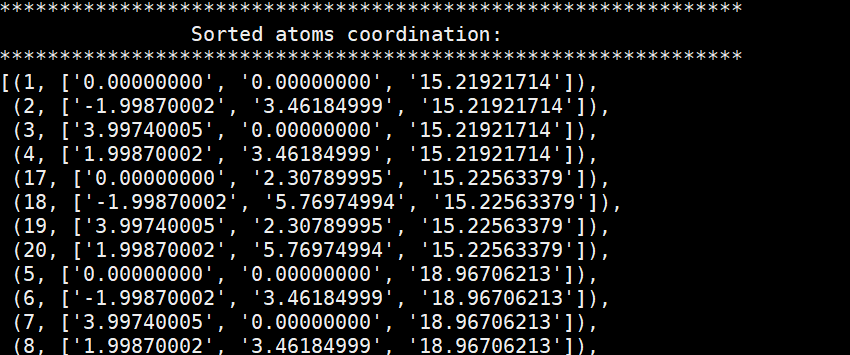
方式1：选择固定的总层数，适用于不对称表面的slab 结构，如图：



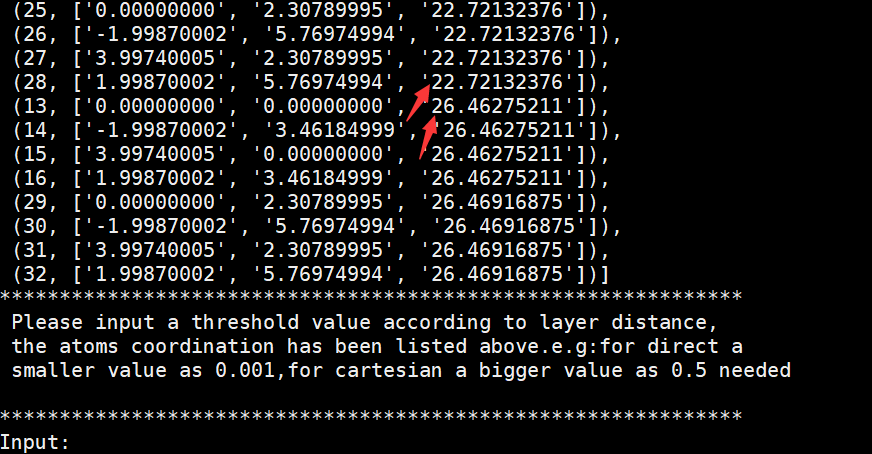
可以选择固定总的层数，比如要固定3层

把atom\_constrian.py 脚本拷贝到POSCAR 所在目录 然后python atom\_constrain.py

屏幕将输出如下提示：



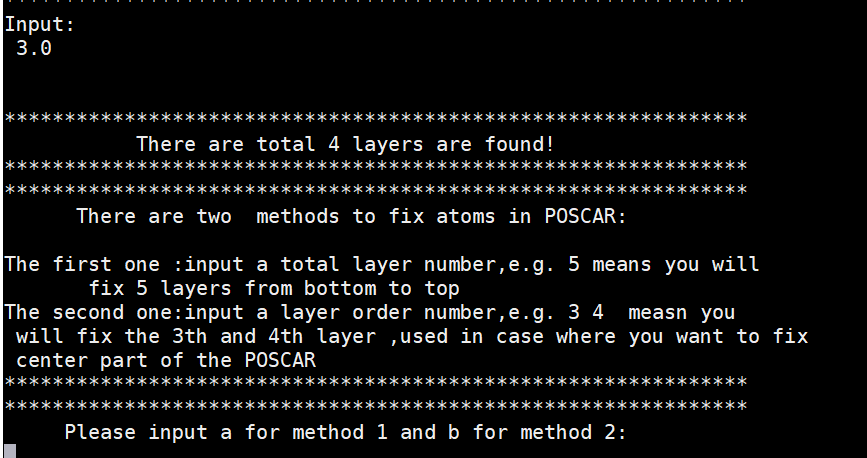
。。。。。。。。。。



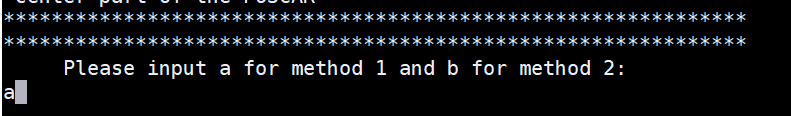
脚本已经把POSCAR 按照 z 方向坐标进行按层排序，并把所有原子序号及坐标打印到屏幕，

由于原本的POSCAR 是笛卡尔坐标，所以z 坐标单位是真实长度，且此时其层间距数值较大，如上图中红色箭头，倒数第二层与最后一层间距数值差异达到3.6A,这个差值可以作为“Input:” 后面层间距的阈值判断的依据，输入的阈值在（0,3.6）范围内时，比如0.3或1.3、或3.3 等，脚本会自动把POSCAR 中两层间的距离大于输入的阈值的原子算作不同的层，比如此例中将会计算出4层，如果输入的阈值为3.7 或更大，由于POSCAR 没有任何两层间的距离比3.7大，因此整个POSCAR 将会被脚本认为只有一层。

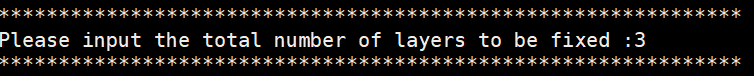
本例输入阈值为3，如图



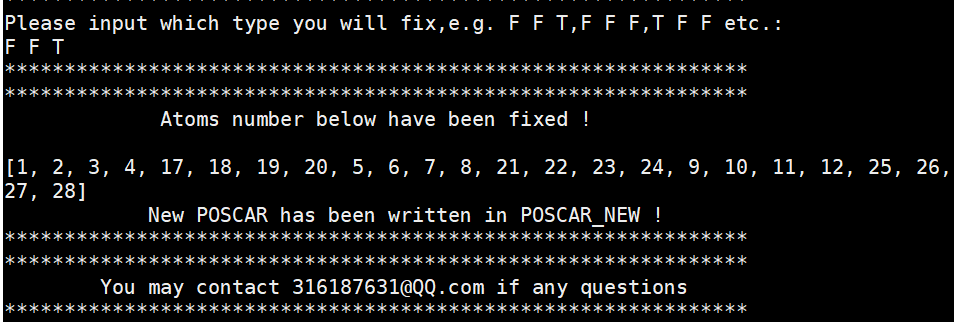
如上分析，脚本计算出有4层，然后提示有两种方式可选择，此步骤选择方式1，按照提示输入a 如下图



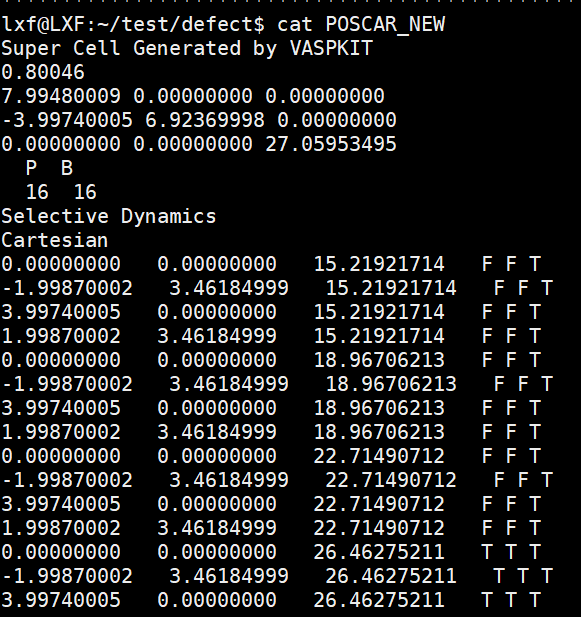
接下来提示输入需要固定的总的层数：输入3 如下图：



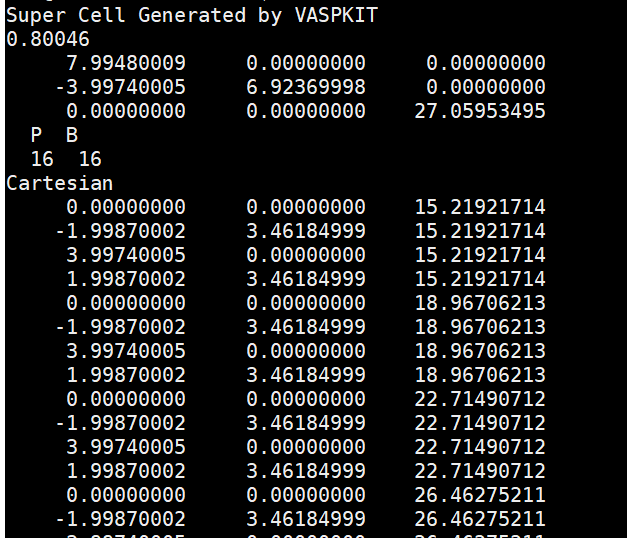
接下来提示输入固定的方式，可输入F F T, F F F ,T F F 等，取决于用户想如何固定原子坐标，输入如下：



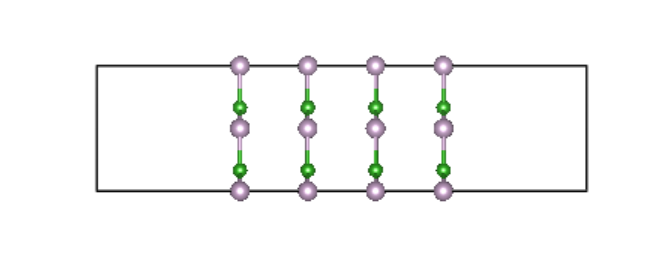
我输入了 F F T，表示将要把POSCAR 中从底部开始的 1-3层中的x y 坐标固定，而放开z坐标，同时，屏幕还输出了被固定原子的所有序数，该序数与原POSCAR的原子序数一致（也就是在图2 中对原子排序时，并没有打乱原POSCAR 原子序数），新的POSCAR 已经被写进POSCAR\_NEW文件，cat POSCAR\_NEW 如图所示：



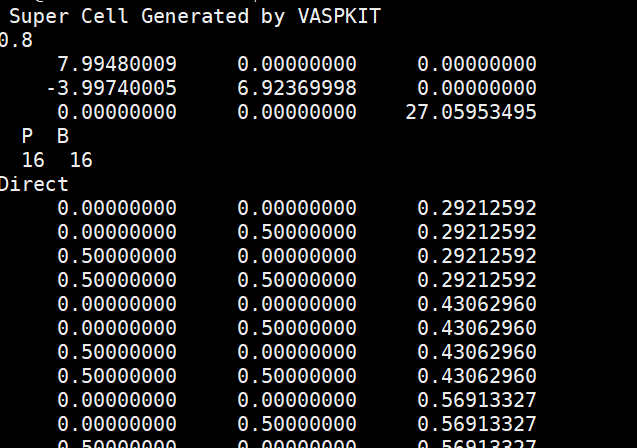
原POSCAR 如图所示：



方式2使用与方式1 打同小异，主要考虑到有的用户想只固定对称型slab 中的中间几层，放开上下两个表面(写教程时临时加的一种方法)，其结构如图下：

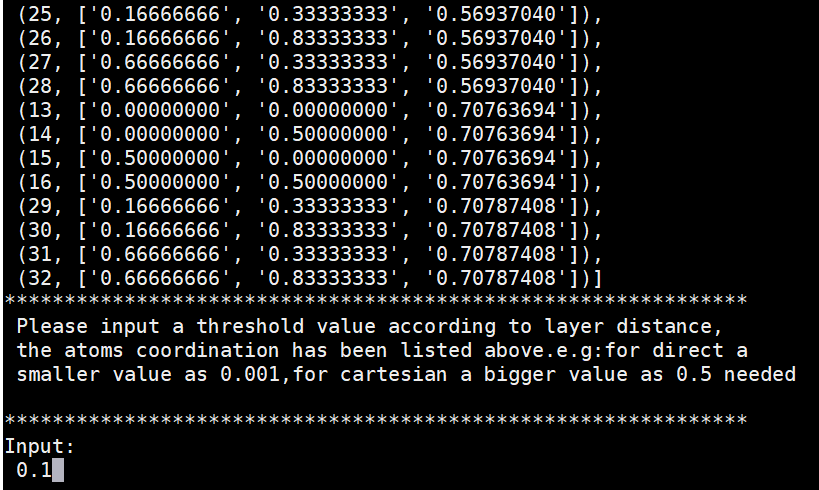


结构同样有4层，这次我把笛卡尔坐标转换为了分数坐标如下：

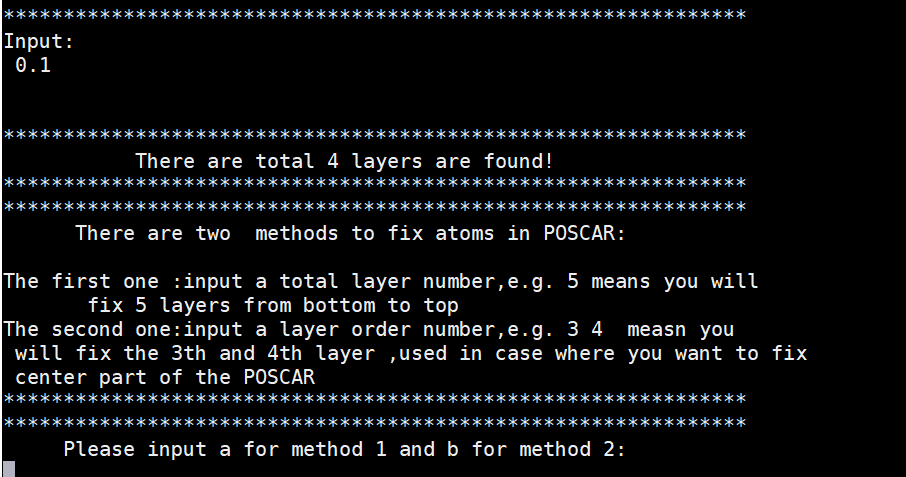


（该转换用我编写的multiplayer.py 脚本完成，该脚本可以有单层二维结构得到多层结构，并实现层间距控制，两层间旋转特殊角度，笛卡尔和分数坐标之间互转等功能，欢迎大家使用。）

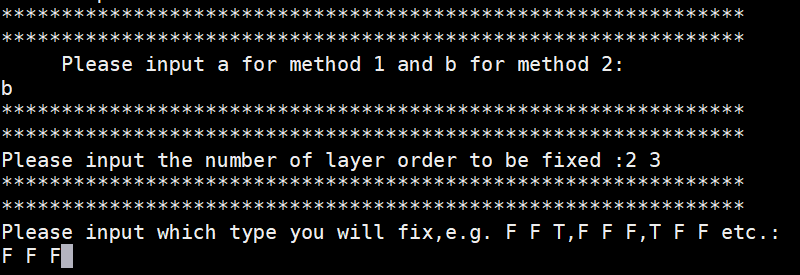
Python atom\_constrain.py



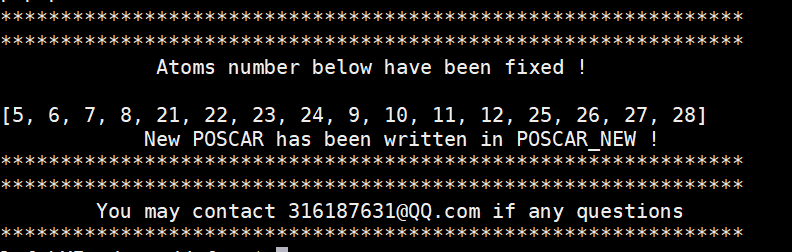
在提示界面我输入0.1 ，其原因是此时由于是分数坐标，两层间的数值差异不是真实的空间距离，因此比较小，但由于已经排序，所以还是很容易分别出层属关系，比如上图0.569 与0.707 应该是属于不同的层，数值差异为0.14，因此上图输入0.1 就足够小了，输出如下图：



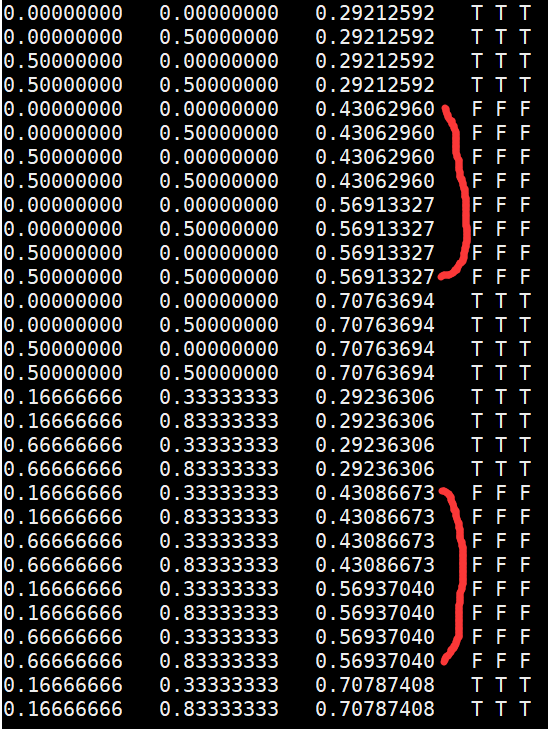
不出意料地判断出有4层结构，此时选择方式2，输入b, 假设想固定2 3 层，输入 2 3 固定方式为F F F



输出提示如下图：

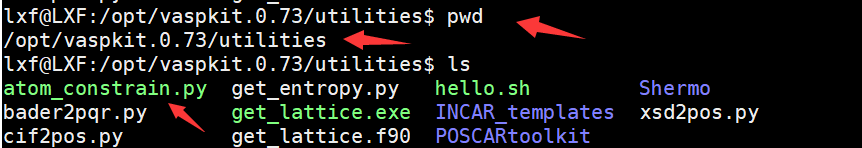


如下图所示：分数坐标为0.569 和0.430 处于2 3 层的原子被固定



Vaspkit 目前功能越来越完善，如果用户自己有些小的脚本可以补充vaspkit 的功能，那么，可以把自己的小脚本“集成到”vaspkit 里面，用vaspkit 统一管理自己的脚本，现在以atom\_constrain.py 为例讲解如何把该脚本“放到”vaspkit 里面，利用vaspkit 惯有的工作模式进行使用：

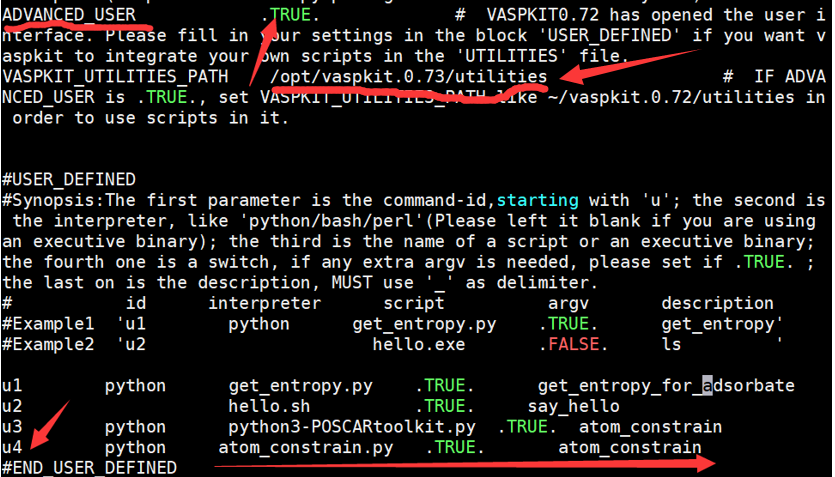
第一步：进入vaspkit 安装目录下的utilities 目录，我安装vaspkit的路径如下 ：



第二步：把atom\_constrain.py 拷贝到此

第三步：vim ~/.vaspkit （或vi~/.vaspkit）

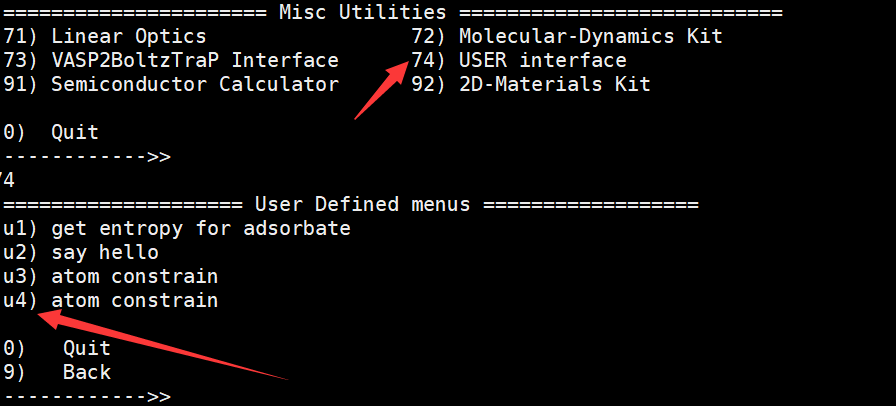
按图下修改：注意图中箭头处



输入“:wq” ,保存并退出

第四步，进入含有POSCAR 的目录体验一把：

输入vaspkit🡪74🡪得到如下图示：



接着输入u4🡪 输入任何字符🡪输入回车后进入脚本工作页面：

