

AIRSS 说明手册

Yang Li

lyang.1915@gmail.com

202.10.24 — 2020.10.28

Contents

1 关于 AIRSS	2
2 计算前的准备	3
2.1 Linux 系统操作	3
2.2 软件的安装与卸载	3
3 buildcell 输入文件	4
3.1 *.cell 文件结构概括	4
3.2 *.cell 文件参数细节	5
3.2.1 结构数据	5
3.2.2 全局参数	9
4 结构弛豫与能量计算	19
4.1 初次运行 AIRSS	19
4.2 联合 airss-pp3 模块弛豫	20
4.3 联合 CASTEP 弛豫	21
4.4 联合 VASP 弛豫	22
4.5 查看计算结果	22
5 计算数据处理	24
A 软件安装	26
A.1 软件主体安装	26
A.2 辅助插件安装	29
A.3 卸载软件	32

1 关于 AIRSS

AIRSS(**A**b **I**nitio **R**andom **S**tructure **S**earching) 是一款由英国剑桥大学 Chris Pickard 教授¹等人开发的材料结构搜索软件。该软件是受 **GPL2 许可证**保护的开源软件。访问其官方网站, <https://www.mtg.msm.cam.ac.uk/Codes/AIRSS>, 便可获取 AIRSS 安装包源码。

所谓结构搜索是指, 对于一晶格结构未知的体系, 在一定的物理条件 (如原子间距, 分布密度, 元素构成, 元素配比等) 限制下, 广泛地猜测其结构, 而后对所得结构进行弛豫, 计算稳定构型能量, 最终得到 DFT 计算上全局最稳定晶格结构的过程。显然人工手动猜测或是计算机盲目地遍历式搜寻是极为笨拙、耗时、甚至是难以实现的。因此, 就需要使用一套成熟的结构搜索软件, 系统且巧妙地捕捉体系构型。

AIRSS 正是这样一款软件。Chris 教授本人关于此软件的介绍可参见 **有关视频**。另外, 还有两个较为常用的结构搜索软件: **USPEX** 和 **CALYPSO**。

与 USPEX 或 CALYPSO 中使用的遗传算法不同, AIRSS 基于完全随机的结构搜索策略。也即, AIRSS 不同结构的产生是完全随机且独立的, 你可以用 AIRSS 随机产生 1000 个结构, 而后独立弛豫。这种随机方法的有效性也在 Chris 教授关于 AIRSS 文章中有所证明²。AIRSS 最强大之处还是在于其众多可调参数。这十分类似于傻瓜相机和专业单反相机的区别。前者简单实用, 可以一键提交任务并获得不错的结果。后者按键众多参数可调, 但可获得各种专业且自定义极高的结果。AIRSS 可以高度自定义自己所要研究的对象, 使用 AIRSS 我们甚至可以验证结构的稳定性, 以及进行晶格表面势能的计算。

令人遗憾的是, AIRSS 作为一个功能强大的结构搜索软件, 其官方手册相对简陋, 未能覆盖全部用途。再加上网络上相关说明文档或教程十分匮乏, 所以大部分人对 AIRSS 这样一个材料设计领域异常优秀的程序并不熟悉。

以下内容算不上指南或教程, 仅仅是学习 AIRSS 的一些记录。本文除了一小部分内容是借鉴软件包自带的 **examples** 中的说明外, 其他大部分结论是自行分析源码、尝试摸索所得。受各种因素限制, 理解和解释上的错误或不可避免。

¹Chris 教授同时还参与研发过 DFT 计算软件 CASTEP。

²[1] PRL 97, 045504 (2006); [2] JPCM 23, 053201 (2011)

2 计算前的准备

2.1 Linux 系统操作

阅读本记录前, 需要您对 Linux 操作系统有一定的了解. 例如, 能理解以下指令的含义:

```
user@machine_name$ ls | grep '.cell'
```

以及下述指令所能引起的灾难性事故:

```
root@machine_name# rm -rf / home/user_name/trash_directory
```

2.2 软件的安装与卸载

附录中以 airss-0.9.0 版本为例, 简要记录了 AIRSS 的安装过程. 最新的airss-0.9.1 在安装过程上做了少许简化, 您可以自行阅读airss-0.9.1中包含的README文件安装此软件.

AIRSS 主运行脚本使用 perl 语言书写, 仅能安装在*nix系统中, 且只支持在命令行 (Command Line) 使用. 安装此软件前, 您最好已经了解 GNU make 的使用方法.

3 buildcell 输入文件

在使用 AIRSS 预测结构时, 一般需要准备一个 *.cell 的文件. 如果要使用 AIRSS 自带的 pp3 计算模块还需要准备一个名为 *.pp 文件.

官方推荐 AIRSS 结合 CASTEP 使用, 且构建了 AIRSS 和 CASTEP 较为完善的接口. 使用 AIRSS 联合 CASTEP 计算时还需要准备两种输入文件: *.cell 文件与 *.param (与 VASP 中的 INCAR 类似) 文件. 前者用于确定材料结构 (包括实空间与倒空间), 后者用于设置 CASTEP 计算过程中需要使用的配置参数 (截断能, 交换关联函等).

AIRSS 的核心组件名为 buildcell, 顾名思义这个组件的作用就是依据用户给出的 *.cell 文件生成一系列结构的随机但符合给定物理条件约束的结构. 如此一来学习 AIRSS 的根本, 就落在了学习如何书写高度自定义化的 *.cell 上.

buildcell 区块的执行大致分为以下几步:

- i- 通过 cell.f90 模块读取 *.cell 文件中的配置信息.
- ii- 通过 build.f90 模块生成满足要求的结构.
- iii- 首先根据给定的原子半径确定合理的晶格常数, 若存在晶格标签 #FIX 则忽略此步骤.
- iv- 依次按要求生成随机的原子新位置.
- v- 根据约束选择接收这一位置, 拒绝这一位置, 或者 **PUSH** 当前原子. **PUSH** 是指: 当两原子位置过近时, 强行将两原子在其连线方向反向移开相同距离的操作. 当原子标签中有 FIX 或 NOMOVE 时, 忽略这一操作.
- vi- 依次作用新的原子, 直至更新完整个结构.
- vii- 视用户设置, 用维象的对势 (pp3) 对结构进行简单弛豫.
- viii- 通过 buildcell.f90 模块输出上述满足要求的随机结构.

下面详细介绍一下 *.cell 文件的设置.

3.1 *.cell 文件结构概括

*.cell 文件相当于结构搜索的种子文件, 您可以在此文件中设置搜索约束条件.

*.cell 文件的内容主要有以下特点:

- (1) 由于 AIRSS 就是部分参与 CASTEP 研发的人员编写的程序, 因此该程序对 CASTEP 极其友好, 这里的 *.cell 文件与 CASTEP 中的 *.cell 文件完全兼容.
- (2) *.cell 文件设置参数时, 需要使用一定的 keywords 标明所设参数的含义.
- (3) AIRSS 完全复用了 CASTEP 中进行结构声明的 keywords, 如 LATTICE_CART 等.
- (4) AIRSS 的核心部分是 buildcell 模块, 可以魔改 AIRSS 同时保留此模块, 使其与其他程序, 如 VASP, 适配.

- (5) 对于那些 CASTEP 中未曾定义只对 AIRSS 有效的 keywords, 需要在其之前附加#, #POSAMP=, 以使得 AIRSS 的 *.cell 文件与 CASTEP 完全兼容.
- (6) *.cell 文件主要由两部分组成: “结构数据” 和 “全局参数”. 相应的, *.cell 中的 keywords 也可以分为上述两类.
- (7) “结构数据” 和 “全局参数” 没有书写顺序上的限制.
- (8) *.cell 中设定的所有的 keywords 和数据均不区分大小写. 同时, 任何标点符号 (除了标明注释内容的符号), 多余的空格和任何空行都将被自动忽略.
- (9) 文件的一行中只能出现一个 keywords 及其对应参数.

以下是预测金属铝结构所用到的 Al.cell 文件:

```

0 user@machine_name$ cat Al.cell
1 %BLOCK LATTICE_CART
2 2 0 0
3 0 2 0
4 0 0 2
5 %ENDBLOCK LATTICE_CART
6
7 %BLOCK POSITIONS_FRAC
8 Al 0.0 0.0 0.0 # Al1 % NUM=8
9 %ENDBLOCK POSITIONS_FRAC
10
11 #MINSEP=1.5
12 user@machine_name$

```

上述 *.cell 文件的结构可做如下概括:

- 前 9 行是由 %BLOCK [keywords] 格式定义的数据读取区块. 这种格式在 CASTEP 中是十分常见的.
- 1 至 5 行数据的 [keywords] 是 LATTICE_CART. 声明使用笛卡尔坐标系定义的单胞基矢.
- 7 至 9 行使用的 [keywords] 是 POSITIONS_FRAC. 这在 CASTEP 中的意义是 “以分数坐标定义的原子位置”. 而在 AIRSS 中, 该数据模块不仅可以锁定原子位置, 还可以定义搜寻过程中对单种原子的约束条件. 具体细节将在之后给出.
- 文件第 11 行的 #MINSEP 是只在 AIRSS 结构搜索种子文件中有效的 **全局参数**. #MINSEP=1.5 表明了两原子间距不得低于 1.5 Å.

可以看到, *.cell 文件的结构大致分为两大部分: **结构数据** 和 **全局参数**.

3.2 *.cell 文件参数细节

3.2.1 结构数据

该部分沿用了 CASTEP 的结构文件中定义数据的模式. 结构数据由两部分构成: 晶格参数数据块, 原子位置数据块. 结构数据的具体模式如下所示:

```

%BLOCK [keywords]
[...]
[structure data]
[...]
%BLOCKEND [keywords]

```

可以在 AIRSS 中使用的结构声明关键字 ([keywords]) 已在 Table 1 中列出.

Table 1: AIRSS Cheat Sheet – Data BOLCK

参数名称	功能说明
LATTICE_CART	使用笛卡尔坐标系, 以一个 3×3 的矩阵定义单胞基矢. 通过在该区块中附加以下标签: #FIX, #CFIX, #ABFIX, 可分别实现: 固定整个晶格形状, 固定晶格常数 c, 固定晶格常数 ab. 另外, 再次强调, 如果要确保晶格参数不变, 除了在晶格参数结构数据加入 #FIX, 以保证在猜测晶格结构时保持其为常值外, 还需要在 DFT 弛豫软件中做相应的设置, 以保证其在结构弛豫时也不变.
LATTICE_ABC	以一个 2×3 的矩阵定义单胞的晶格参数. 矩阵第一行是参数 abc, 第二行是三个夹角. 在同一个 .cell 中, 与 LATTICE_CART 二选一即可. 通过在该区块中附加以下标签: #FIX, #CFIX, #ABFIX, 可分别实现: 固定整个晶格形状, 固定晶格常数 c, 固定晶格常数 ab.
POSITIONS_FRAC	以晶格基矢坐标系下的分数坐标定义原子位置.
POSITIONS_ABS	以笛卡尔坐标系下的绝对数值坐标定义原子位置. 与 POSITIONS_FRAC 二者选一即可.
SYMMETRY_OPS	定义生成的单胞所必须满足的对称操作, 四行一组, 前三行定义旋转操作, 第四行行定义平移操作. 具体用法可参考 CASTEP:SYMMETRY OPS .

一般进行结构搜索时, 只需要在 AIRSS 的 *.cell 文件中给出**晶格参数**和**原子位置**的数据即可.

晶格参数 首先介绍晶格参数的设定, 以 LATTICE_CART 为例.

例 1.

```

1 %BLOCK LATTICE_CART
2 20 0 0
3 0 20 0
4 0 0 20
5 #FIX
6 %BLOCKEND LATTICE_CART

```

上述字段构建了一个 $20 \times 20 \times 20 \text{ \AA}^3$ 的正方体作为晶体的单胞. #FIX 称为晶格标签 (Lattice Tags).

它声明了晶格常数在搜寻过程 (生成初始随机结构) 中是不能改变的. 如果您想保证体系晶格常数在使用 CASTEP 或 airss-pp3 进行结构粗略优化时也不变, 则需要在相应计算软件的输入文件中设置.

原子位置 下面再介绍原子位置的设置方法, 以POSITIONS_FRAC为例.

例 2.

```
1 %BLOCK POSITIONS_FRAC
2 Al 0.0 0.0 0.0 # Al1 % NUM=2
3 Mg 0.0 0.0 0.0 # Mg1 % NUM=4
4 O 0.4 0.2 0.3 # O1 % NUM=1 POSAMP=0 FIX
5 O 0.1 0.1 0.1 # O2 % NUM=1 POSAMP=0 UNMOVE
6 H 0.3 0.3 0.6 # free_H
7 H 0.0 0.0 0.0 # H-set % ANGAMP=0 POSAMP=0
8 H 0.0 0.0 0.0 # H-set % ANGAMP=0 POSAMP=0
9 %BLOCKEND POSITIONS_FRAC
```

通过这个例子, 我们看到, 原子位置结构数据的基本格式是:

```
[element] [x] [y] [z] # [atoms_set_name] % [tag1] [tag2] [tag3]
```

每一行内部的元素含义如下:

- 第一项是元素名称, 如果将元素名称设置为 Z, 则表示此原子是个空位, 空位与真实原子一样具有体积和 PUSH 属性, 但不会在最终结果中输出.
- 二三四项是原子位置坐标.
- 井号 (#) 后的第一项是原子所在原子集的名称, 原子集名称可以被设置成任意字符. 原子集可以由多个原子构成, 他们是结构搜索中的一个基本单元. **原子集名称相同的原子被标定为同一个原子集, 同一个原子集中的原子做相同的随机移动, 且不受 PUSH 影响.** 如无特殊需求, 为了实现自由度相对的最大化, 一般需要为每个原子都单独设置原子集 (每个原子集只含有一个原子).
- 百分号 (%) 之后的内容都是原子标签 (Atom Tags).

Table 2 是各个原子标签的具体说明.

Table 2: AIRSS Cheat Sheet – Atom Tags (distance in Å)

参数名称	输入类型	默认值	功能说明
NUM= n , NUM= $n_{min} - n_{max}$	int	1	定义该行原子实际代表的原子个数. 使用 NUM 定义的重复原子会被逐个单独分配到不同的原子集中.
POSAMP= d	float	-1.0	定义该行原子结构搜索过程中, 能偏离初始位置的最大距离. 若为负值则无限制.
MINAMP= d	float	0.0	定义该行原子结构搜索过程中, 偏离初始位置的最小距离.

Continued on next page...

Table 2 – Continued from previous page

参数名称	输入类型	默认值	功能说明
XAMP= d	float	-1.0	定义该行原子结构搜索过程中, 在 X 轴方向上移动的最大距离. 若为负值则无限制. 设置该项会令 POSAMP 和 MINAMP 失效.
YAMP= d	float	-1.0	定义该行原子结构搜索过程中, 在 Y 轴方向上移动的最大距离. 若为负值则无限制. 设置该项会令 POSAMP 和 MINAMP 失效.
ZAMP= d	float	-1.0	定义该行原子结构搜索过程中, 在 Z 轴方向上移动的最大距离. 若为负值则无限制. 设置该项会令 POSAMP 和 MINAMP 失效.
ANGAMP= θ	float [0, 360]	-1.0	原子绕自身所在原子集中心旋转角度的最大值 (与 POSAMP 互相独立, 分开作用). 存在晶格标签 #FIX 时, ANGAMP=0, POSAMP=0 与 NOMOVE(\FIX) 共用, 可保证原子在原始位置不动. 如果 cell 文件中所有原子集都只有 1 个原子, 则该参数失效, ANGAMP 强制设为 0. 若为负值则无限制..
RAD= d	float	0.0	设置原子的半径, 用于判断两个原子间的最小距离, MINSEP. 其优先级低于直接设置全局参数 MINSEP.
FIX	void	off	该原子在产生结构时不受 PUSH 影响, 同时向 cell 文件中写入指令使其在 CASTEP 弛豫时也不动. 任何接近该原子的可移动原子都会以 2 倍的 PUSH 步长被 PUSH 回去. 这一指令仅在晶格被 #FIX 时有效. 原子标签中的 #FIX 与晶格标签中的 #FIX, #CFIX, #ABFIX 作用不同.
NOMOVE	void	off	该原子在产生结构时不受 PUSH 影响, 任何接近该原子的可移动原子都会以 2 倍的 PUSH 步长被 PUSH 回去. 这一指令仅在晶格被 #FIX 时有效.
COORD= n , COORD= $n_{min} - n_{max}$	int	-1	约束改行原子的配位数 (最近邻原子数). 若为负值则无限制.

Continued on next page...

Table 2 – Continued from previous page

参数名称	输入类型	默认值	功能说明
NN= $\pm elm$	string	null	规定原子最近邻元素种类, ‘+’ 代表必须近邻该元素, ‘-’ 代表不能近邻该元素. 一个原子只能指定一种元素近邻或不近邻 (NN 参数仅会被读入一次). 若为空, 则表示无限制
OCC= p	float [0, 1]	1.0	该点位占据原子的几率, 同种元素在不同位置的占据几率之和必须为大于 0 的整数. 接收分数形式的输入, 如 OCC=1/3. 若该值设为小于 0 的数, 则强行将该原子的 OCC 设置为 1, 同时不再输出该原子可能因对称性衍生的其他原子位置. 若原子同时有 FIX 或 NOMOVE 标签, 则OCC 强制设为 1. 若该原子 OCC 大于 1, 则将其设置为空位 (与将元素名称设为 Z 的效果相同).
MULT= p	float	-1.0	设置该原子点位的 multiplicities. 这一数值为正时, 强制覆盖 OCC 参数的设置, 将其设置为 MULT/SYMM_NUM. 其中 SYMM_NUM 为晶格对称操作的个数. 若该值为负数, 则直接将 MULT 的值赋给 OCC. 关于格点 multiplicities 与对称性的关系可参见 BCS: Wyckoff Position
PERM	void	off	在完成原子位置选定后, 重排 (按一定概率互相交换) 指定原子位置间的元素种类. 需要配合全局参数 #PERMUTE 使用, 否则该参数将被关闭.
ADATOM	void	off	表明该行原子是等, 未加该标签的所有原子位置生成结束之后, 再加入的原子.
ATHOLE	void	off	表明该原子位于被挖掉的孔洞之中, 输出结构中将自动删除该原子. 常与全局变量 #HOLE=, #HOLEPOS= 联用.

3.2.2 全局参数

全局参数是 *.cell 文件中指明结构搜索过程中应遵守的条件的语句. AIRSS 中的全局参数均以#开头, 对体系中晶格以及全部的原子作用. 另外, 有

些全局参数有和原子标签有相同的作用, 此时, 原子标签设置的优先级要高于全局参数.

全部可用的全局参数的详细使用方法和功能如 **Table 3** 所示.

Table 3: AIRSS Cheat Sheet – Global Pragma (distance in Å)

参数名称	输入类型	默认值	功能说明
<code>#CELLCON=$a\ b\ c\ \alpha\ \beta\ \gamma$</code>	int	off	定义晶胞应满足的条件, a, b, c 是晶格常数, α, β, γ 是三个晶角. 晶格常数可选填的值有 $\{0, -1\}$, 晶角可选填的值有 $\{0, -1, \theta\}$. 0 表示无约束, -1 表示相等. 如 <code>#CELLCON=-1-1 -1 90 90 90</code> 表示立方晶系.
<code>#SYSTEM=crs</code>	string	off	设置结构所在的晶系, 可选的输入有 [Tric, Mono, Hexa, Rhom(/Trig), Orth, Tetr, Cubi]. 该参数实际为 <code>#CELLCON=</code> 参数的若干特殊组合.
<code>#CONS=p</code>	float (0, 1]	0.4	约束单胞边长. 接近 0 表示完全没有约束, 等于 1 表示尽一切可能约束晶胞 abc 三边等长.
<code>#ACONS=p</code>	float (0, 1]	0.5	约束晶胞晶角, 使其远离某一晶格参数极小、晶胞扁平的情况. 接近 0 表示完全没有约束, 等于 1 表示尽一切可能约束三个键角严格等于 90 度.
<code>#CELLAMP=p</code>	float	-1.0	在一定限度内, 随机变化晶格形状. 设为负值时采用晶格形状将完全随机变化. p 越小, 晶格偏离 cell 文件中给定值越小. 若要启用该参数, 建议不要设置超过 1.0 的值. ³ 使用该参数后, 晶格标签 <code>#ABFIX</code> 与 <code>#CFIX</code> , 原子标签 <code>#CONS</code> 与 <code>#ACONS</code> 都将被弃用. 若为负数则关闭该参数设置.
<code>#VACUUM=d</code>	float	0.0	在晶格 (Z 方向) 中加入 d Å 的真空层.
<code>#NFORM=n</code>	int	-1	声明单胞中实际存在的原子个数是的是 <code>%BLOCK POSITIONS_*</code> 或 <code>#SPECIES</code> 中定义的 n 倍. 该选项不可与 <code>#NATOM</code> 联用. 若为负值则此参数被关闭.

Continued on next page...

³该参数使用方法详见 `airss-0.9.1/src/buildcell/src/build.f90` 第 100 行 `cellamp` 变量的使用.

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#SUPERCELL= n , #SUPERCELL= $n_a n_b n_c$, #SUPERCELL= $a_x a_y a_z b_x b_y b_z c_x c_y c_z$	int	off	定义超胞的尺寸. 可以使用超胞中单胞的个数 n , 超胞晶格基矢在三个方向的数值 $n_a n_b n_c$, 或是超胞与单胞晶格基矢之间的变换矩阵 $a_x a_y a_z b_x b_y b_z c_x c_y c_z$ 定义新的超胞. 超胞的构建策略是: 首先用单胞构建符合条件的原子构型, 而后根据复制粘贴得到超胞.
#SLAB	void	off	声明该结构是一个二维平面结构, 取超胞时将不再向 c 方向扩胞, 同时检查结构对称性时也将忽略 c 方向.
#SURFACE	void	off	声明该结构是一个表面结构, 表面结构继承 SLAB 的全部特性, 同时增加一条与表面相关的对称性限制条件. ⁴
#MOLECULES	void	off	声明同一个原子集中的原子构成了一个基本分子构型, 检查配位时将不再对同一个原子集内的原子做检查.
#CLUSTER	void	off	声明预测的结构是无周期性的团簇.
#FOCUS= n	int	0	约束最终结构必须是 n 组分的 (由 n 种不同的元素构成). 该参数一般用于变组分分析. 小于等于 0 表示无约束.
#OCTET	void	off	检查化学配比是否电子配平 (全部电子数是否可以被 8 整除). 默认关闭.
#POSAMP= d	float	-1.0	与原子标签中的定义相同, 结构搜索中原子偏离初始点的最大位置. 负值为无限制.
#MINAMP= d	float	-1.0	与原子标签中的定义相同, 结构搜索中原子偏离初始点的最小位置. 负值为无限制.
#XAMP= d	float	-1.0	与原子标签中的定义相同, 结构搜索中原子 X 方向偏离初始点的最大振幅. 负值为无限制. 设置该项会令 POSAMP 和 MINAMP 失效.

Continued on next page...

⁴见源码 airss-0.9.1/src/buildcell/src/cell.90 第 3216 行

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#YAMP= d	float	-1.0	与原子标签中的定义相同, 结构搜索中原子 Y 方向偏离初始点的最大振幅. 负值为无限制. 设置该项会令 POSAMP 和 MINAMP 失效.
#ZAMP= d	float	-1.0	与原子标签中的定义相同, 结构搜索中原子 Z 方向偏离初始点的最大振幅. 负值为无限制. 设置该项会令 POSAMP 和 MINAMP 失效.
#ANGAMP= θ	float [0, 360]	-1.0	与原子标签中的定义相同, 原子绕自身所在原子集中心旋转角度的最大值. 负值为无限制.
#MINBANGLE= θ	float (0, 360]	0.0	设置搜索结构中, 原子键角的最小值.
#MAXBANGLE= θ	float (0, 360]	180.0	设置搜索结构中, 原子键角的最大值.
#MINSEP= d #MINSEP= d X-X= d_{X-X} X-Y= d_{X-Y} ...	float	0.0	两原子间最小距离, 也可以用来定义两原子距离固定是多少. 比如 #MINSEP=2.0 Li-Li=2.6 Ge-Ge=2.51
#RAD= d	float	0.0	与原子标签中的定义相同, 定义原子的半径大小.
#COORD= n	int	-1	与原子标签中的定义相同, 设置原子的配位数限制. 负值为无限制.
#SPIN= S_{total} S_{abs} , #SPIN= S_{total} S_{abs} "elm ₁ elm ₂ ... "	float, string	off	随机设置体系每个原子上共线自旋的取值. 并要求, 指定原子自旋值的和/原子数 = S_{total} , 指定原子自旋值绝对值的和/原子数 = S_{abs} , 未被指定的原子自旋保持为 0. 可以用被双引号包裹的带空格的元素符号指明哪些是指定原子. 若未指定任何原子, 则默认所有原子都是指定原子. 例如 #SPIN=0 5 "Fe Co " 或者 #SPIN=1 4
#SPECIES=elm ₁ %tags ₁ , elm ₂ %tags ₂ , ... #SPECIES=elm ₁ , elm ₂ ...	string	off	使用简化记号定义体系原子组分. 例如, #SPECIES=Si%NUM=1 COORD=2, O%NUM=2. 该参数不能与 BLOCK POSITIONS_* 同时出现.

Continued on next page...

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#NATOM= n , #NATOM= $n_{min} - n_{max}$	0	int	与 #SPECIES 联用, 定义单胞中总原子数, 一般用于变组分分析 (变胞预测). 且使用此全局参数会使 #SPECIES 指令包含的%后的原子标签全部失效. ⁵ 若 #SPECIES 中含有多中元素, 则每种元素随机数目, 保持总和为 NATOM. 该参数设为 0 时自动失效.
#TARGVOL= V , #TARGVOL= $V_{min} - V_{max}$	float	init. cell vol.	固定晶格体积为 V , 或 V_{min} 到 V_{max} 之间的随机数值. 存在晶格标签 #FIX 时该参数失效. 默认为初始给定原胞的体积. 该参数常在不引入 BLOCK POSITION_* 区块时使用.
#VARVOL= V	float	init. cell vol.	与 #TARGVOL= 作用相同, 该参数会覆盖 #TARGVOL= 的设置. 默认为初始给定原胞的体积.
#SLACK= p	float [0,1)	0.0	使用此参数可整体降低体系对原子间成键 (原子相对位置) 的限制, p 越大对原子间间距和角度的要求越低. 默认为 0, 推荐值 0.1–0.3.
#AUTOSLACK= p	float [0,1)	off	使用此参数可整体降低体系对原子间成键 (原子相对位置) 的限制, p 越大对原子间间距和角度的要求越低. 若给定的初始值 p 不合适, 则以 0.01 的步长递增 SLACK, 直到找到合适的 SLACK 值.
#FLIP	void	off	搜索结构时, 对原子集引入随机翻转操作. 若体系中所有原子集中均只有一个原子, 则该参数失效.
#REMOVE	void	off	删除 (PUSH 后) 重叠的原子 (之一). 可以用于高初始原子密度的结构预测.
#TIGHT	void	off	使得生成的结构更加紧密.
#SYMMORPHIC	void	off	只检查体系是否存在点式对称操作.

Continued on next page...

⁵详细原因请参见源码 “airss-0.9.1/src/buildcell/src/cell.f90” 第 493 行与 514 行区别.

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#SYMMOPS= n , #SYMMOPS= $\sim n$ #SYMMOPS= $n_{min} - n_{max}$	int	off	声明生成的结构中必须含有 n 种对称操作. 若体系是周期性晶体结构, 推荐从下述整数中选取: 1,2,3,4,5,6,8,12,16,24,48. 若输入中含有波浪号 \sim 则表明, 体系大致满足该对称性即可, 只检查体系 general positions 的对称性. ⁶
#SYMM= spg , #SYMM= $\sim spg$	string	off	生成的结构必须在 spg 空间群中, spg 是空间群的名称. 若输入中含有波浪号 \sim 则表明, 体系大致满足该对称性即可, 只检查体系 general positions 的对称性.
#SYMMNO= n , #SYMMNO= $\sim n$	int	off	生成的结构必须在第 n 号空间群中, n 是空间群的序号. 若输入中含有波浪号 \sim 则表明, 体系大致满足该对称性即可, 只检查体系 general positions 的对称性.
#SGRANK= n	int	230	设置空间群寻找/锁定的序号上限 n . 此值设为 230 时, 接受任何空间群的对称性锁定.
#ADJGEN= n , #ADJGEN= $n_{min} - n_{max}$	int	0	调整晶胞中使用 general positions(GP) 的个数. 该值为 0 时, 会强制最大程度地使用与对称相关的 GP. 增大该值则将允许更多地使用 Special positions(SP) 而降低对 GP 的要求. 如果尝试后发现难以生成符合条件结构, 则程序将动态增加该值. 关于 SP/GP 与晶体空间群的关系请参考: Wiki: Wyckoff Position
#BREAKAMP= d	float	0.0	在晶格矢量 a 方向随机移动原子破坏原有对称性, 原子分数坐标移动距离: $d_a^{(frac)} = (random(0,1) \times d)^{1/3}$
#NOPUSH	void	off	对于距离过近的原子不引入 PUSH 步, 直接拒绝该构型. 该关键字不会关闭 #SPHERE 等关键字引入的晶体中心势场的 PUSH 步.
#PUSHSTEP= p	float	0.25	每一步 PUSH 移动距离大小 (step-size) 的比例参数.

Continued on next page...

⁶源代码中这样描述: Symmetry is only be approximately applied (filling general positions only)

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#PUSHMAX= n	int	100	设置最大 PUSH 步数 (在 buildcell 的输出中, 两个 X 之间所夹的 * : - 符号的个数).
#OVERLAP= cov	float	off	结束 PUSH 之后, 再附加采用 TPSD ⁷ 对势对晶体结构进行简单弛豫. cov 为收敛判据, 其值越小对晶格收敛限制越高, 推荐值为 0.1–0.2.
#RASH	float	off	在使用 TPSD 对势弛豫结束后再引入原子集之间的随机位移和旋转 (SHAKE step). 设置过 #OVERLAP 之后此参数才有效.
#RASH_POSAMP= d	float	1.0	设置由 RASH 引入的 SHAKE 步移动原子集的最大距离, 与 POSAMP 类似. 该项必须为一个小的正值, 不可设为负数 .
#RASH_ANGAMP= θ	float (0, 360]	30.0	设置由 RASH 引入的 SHAKE 步原子集绕自身中点转动的最大角度, 与 ANGAMP 类似. 该项必须为一个小的正值, 不可设为负数 .
#CELLADAPT	void	off	在设置#OVERLAP的情况下附加设置此项, 可强制要求 TPSD 简单结构弛豫时同时尝试在保持体积不变的情况下改变单胞的形状. 体系默认不会在 TPSD 结构优化步改变单胞形状. 若存在晶格标记#FIX 或者设置了 #CELLAMP=0 则此参数失效.
#THREE= p	float	TODO	使用三体势能代替 TPSD 弛豫结构, 该参数对应的功能尚未在 airss-0.9.1 中实现 .
#COMPACT	void	on	对最终生成的单胞进行 nigli reduce 操作. 在晶格形状没有被锁定时 (未引入晶格标签#FIX, #CFIX, #ABFIX), 该项默认打开. R. W. Grosse-Kunstleve, N. K. Sauter and P. D. Adams, Acta Cryst., A60, 1-6 (2004)
#NOCOMPACT	void	off	强制关闭 COMPACT 操作.

Continued on next page...

⁷Two-point Step Size Gradient Methods - Barzilai and Borwein, IMA Journal of Numerical Analysis (1988) 8, 141-148

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#PERMUTE	void	off	在完成原子位置选定后, 重排 (按一定概率互相交换) 指定原子位置间的元素种类. 可以联合原子标签 PERM 使用.
#PERMFRAC= p	float [0, 1]	1.0	设置重排发生的概率.
#HOLE= d	float	-1.0	设置在晶格上切割球洞的半径. 设为负数时不对成型的结构做任何处理.
#HOLEPOS= $f_a f_b f_c$	float	random	设置在晶格上切割球洞的位置 (分数坐标). 默认为随机位置. 可与原子标签中的ATHOLE联用.
#VACANCIES= $n@elm$	float, string	off	等待结构生成完毕后, 选取 n 个元素种类为 elm 的原子替换为空位. ⁸
#MAXTIME= t	float	1.0	设置对一个猜测结构 PUSH 步使用时间的上限, 超过该时间程序将停止 PUSH 并重新猜测新结构. 默认 $t = 1s$.
#NFAILS= n	int	0	每个结构允许的失败次数 (在 buildcell 命令输出中出现 X 的次数). 若其值为 0, 则无限制.
#SPHERE= r	float	off	在单胞中心处引入一球状势能. 设置球势能的吸引半径为 r . 当原子与晶格中心距离大于 r 时, 会受到指向晶格中心的 PUSH.
#ELLIPSOID= $r \epsilon$	float	off	在单胞中心处引入一椭球状吸引势. r 为椭球势能半长轴长度, ϵ 为形变程度. 当原子与晶格中心距离大于 r 时, 会受到指向晶格中心的 PUSH. $\epsilon = 0$ 为球形, ϵ 越大畸变越严重.
#PANCAKE= $r \epsilon$	float	off	在单胞中心处引入一圆饼状吸引势. r 为圆饼半径, ϵ 为形变程度. 当原子与晶格中心距离大于 r 时, 会受到指向晶格中心的 PUSH. $\epsilon = 0$ 为偏平圆饼, $\epsilon = 1$ 为接近球形的势能.

Continued on next page...

⁸该参数本来还设计了一种不加 @ 符号的输入, 但目前在 airss-0.9.1 中, 这种输入在后续处理中存在一些 BUG, 因此在此未予说明.

Table 3 – Continued from previous page

参数名称	输入类型	默认值	功能说明
#CIGAR= $r \ \varepsilon$	float	off	在单胞中心处引入一雪茄状吸引势. r 为雪茄长度, ε 为形变程度. 当原子与晶格中心距离大于 r 时, 会受到指向晶格中心的 PUSH. $\varepsilon = 0$ 为针尖状势能, $\varepsilon = 1$ 为接近球形的势能.
#CYLINDER= r	float	off	在单胞中心处引入一圆筒状势能 (原子在 Z 方向不受力). r 为圆筒势能吸引半径. 当原子与晶格中心距离大于 r 时, 会受到指向晶格中心的 XY 方向的 PUSH.
#CORE= r	float	off	对定义的球状 (椭球状, 圆饼状, 雪茄状, 圆筒状) 吸引势附加排斥核心. 设置排斥核心半径 (长轴长度, 半径, 雪茄长度, 圆筒半径) 为 r . 当原子与晶格中心距离小于 r 时, 会受到远离晶格中心方向的 PUSH.
#WIDTH= l	float	off	使用平面状势能 (原子在 X 和 Y 方向均不受力), 附加计算 PUSH 步的距离. l 为平面状势能吸引长度. 当原子与原点 (origin) 距离大于 l 时, 会受到指向原点的 Z 方向的 PUSH.
#SHIFT= h	float	0.0	将平面势移动至 $Z=h$ 的位置 (origin 的位置), 默认 $h = 0$.

现在您应该可以轻松读懂下述内容了:

```

1 %BLOCK LATTICE_CART
2 20 0 0
3 0 20 0
4 0 0 20
5 #FIX
6 %ENDBLOCK LATTICE_CART
7
8 %BLOCK POSITIONS_FRAC
9 A1 0.0 0.0 0.0 # A11 % NUM=7-13 COORD=4
10 %ENDBLOCK POSITIONS_FRAC
11
12 #MINSEP=1.5
13 #CLUSTER
14 #OVERLAP=0.2
15 #RASH
16 #POSAMP=3.0
17 #MINBANGLE=80
18 #MAXBANGLE=120

```

对于只存在全局参数的结构文件, 只要您对晶体结构 (如晶体原子构成, 晶格大概的体积大小等) 声明得足够清楚, AIRSS 也是可以接受的. 下面是两则合法且十分简洁的 AIRSS 结构种子文件.

例 4.1

```
1 #VARVOL=15
2 #SPECIES=A%NUM=4,B%NUM=1
3 #NFORM=2
4 #MINSEP=1.5
```

例 4.2

```
1 #VARVOL=15
2 #SPECIES=A,B,C
3 #NATOM=2-8
4 #MINSEP=1.5
```

可见, SPECIES=[element1]%NUM=[n1],[element2]%NUM=[n2] 相当于一个简化的BLOCK POSITIONS_FRAC原子位置结构数据.

4 结构弛豫与能量计算

4.1 初次运行 AIRSS

初次接触 AIRSS, 您可以在终端输入`airss.pl`指令查看软件欢迎界面.

```
user@machine_name$ airss.pl

      .o.      ooooo ooooooooo.      .oooooo..o      .oooooo..o
      .888.      '888' '888      'Y88. d8P'      'Y8 d8P'      'Y8
      .8:888.      888      888      .d88' Y88bo.      Y88bo.
      .8' '888.      888      888ooo88P'      ':Y8888o.      ':Y8888o.
      .88ooo8888.      888      888'88b.      ':Y88b      ':Y88b
      .8' '888.      888      888 '88b. oo      .d8P oo      .d8P
o88o      o8888o o888o o888o o888o 8::88888P' 8::88888P'

      Ab Initio Random Structure Searching
      Chris J. Pickard (cjp20@cam.ac.uk)
      Copyright (c) 2005-2018

Please cite the following:

[1] C.J. Pickard and R.J. Needs, PRL 97, 045504 (2006)
[2] C.J. Pickard and R.J. Needs, JPCM 23, 053201 (2011)

Usage: airss.pl [-pressure] [-build] [-pp0] [-pp3] [-gulp] [-lammps]
               [-gap] [-psi4] [-cluster] [-slab] [-dos] [-workdir]
               [-max] [-num] [-amp] [-mode] [-minmode] [-sim] [-symm]
               [-nosymm] [-mpin] [-steps] [-best] [-track] [-keep]
               [-seed]
      -pressure f Pressure (0.0)
      -build      Build structures only (false)
      -pp0        Use pair potentials rather than Castep (0D) (false)
      -pp3        Use pair potentials rather than Castep (3D) (false)
      -gulp       Use gulp rather than Castep (false)
      -lammps     Use LAMMPS rather than Castep (false)
      -gap        Use GAP through QUIP/QUIPPY/ASE (false)
      -psi4       Use psi4 (false)
      -vasp       Use VASP (false)
      -cluster    Use cluster settings for symmetry finder (false)
      -slab       Use slab settings (false)
      -dos        Calculate DOS at Ef (false)
      -workdir s  Work directory ('.')
      -max        n Maximum number of structures (1000000)
      -num        n Number of trials (0)
      -amp        f Amplitude of move (-1.5)
      -mode       Choose moves based on low lying vmodes (false)
      -minmode    n Lowest mode (4)
      -sim        f Threshold for structure similarity (0.0)
      -symm       f Symmetrise on-the-fly (0.0)
      -nosymm     f No symmetry (0)
      -mpin       n Number of cores per mpi Castep (0)
      -steps      n Max number of geometry optimisation steps (400)
      -best       Only keep the best str. for each compos. (false)
      -track      Keep the track of good str. during RESH(false)
      -keep       Keep intermediate files (false)
      -seed       s Seedname ('NONE')
user@machine_name$
```

`airss.pl`是使用 AIRSS 执行结构搜索的主要指令. 欢迎界面中已经系统且简要地说明了该指令的用法. 用法解释中的表格有三列. 第一列是传入参数的调

用名称; 第二列是传入参数的数据类型, **f**代表浮点数, **n**代表整数, **s**代表字符串, “空”代表逻辑`true||false`. 第三列是对此参数的详细解释.

4.2 联合 airss-pp3 模块弛豫

airss-pp3是 AIRSS 自带的 pp3 对势 (pair potential) 计算模块. 其功能是使用化学上经典的分子势场 (如 6-12 势能) 弛豫原子结构并输出体系能量. 由于这一模块使用的是维象的晶体能量模型, 不涉及任何第一性原理的复杂运算, 因此其实现简单、效果稳定、计算速度快, 适用于简单组分的小体系. 更具体的, AIRSS 中计算对势采用了如下公式:

$$E_{ij} = 4\varepsilon_{ij} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^m - \beta_{ij} \left(\frac{\sigma_{ij}}{r_{ij}} \right)^n \right] \quad (1a)$$

$$E = \sum_{i \neq j}^{\text{all ions}} E_{ij} \quad (1b)$$

其中 i, j 标定了不同原子位置, 当元素种类不同时, $\varepsilon_{ij}, \sigma_{ij}$ 等变量对应不同的值.

使用该模块时, 需要首先配置名为 ***.pp** 的参数文件. 这个文件中存储了对势的相关参数, 其内部书写形式如下:

```

1 n_spec m n range
2 specs
3 # Epsilon
4 eps_11 eps_12 ...
5 eps_22 ...
6 ...
7 # Sigma
8 sgm_11 sgm_11 ...
9 sgm_11 ...
10 ...
11 # Beta
12 beta_11 beta_12 ...
13 beta_22 ...
14 ...

```

上述参数中,

- 第 1 行各项分别是: 元素个数; 对势中指数 m, n 的数值; 对势中能量极小值 (力为 0 处) 处距离原子中心距离 d_{min} 与 σ 的比值, 也即 $d_{min}^{(ij)} = \sigma_{ij} * d_{range}$.
- 第 2 行指明了体系中的元素种类, 不同元素间用空格隔开.
- 第 3 行是注释行, 在程序中无意义, 但必须存在.
- 第 4 至 $(n_{spec} + 3)$ 行声明了元素之间的 *varepsilon* 值对应的矩阵.
- 第 $(n_{spec} + 4)$ 行是注释行, 在程序中无意义, 但必须存在.
- 第 $(n_{spec} + 5)$ 至 $(2n_{spec} + 4)$ 行声明了元素之间的 *sigma* 值对应的矩阵.
- 第 $(2n_{spec} + 5)$ 行必须为书写含有 “Beta” 字符的注释. 若此字符未出现, 则系统将强制把全部 *beta* 值设为 1.
- 第 $(2n_{spec} + 6)$ 至 $(3n_{spec} + 5)$ 行声明了元素之间的 *beta* 值对应的矩阵.

例如,

```
1 2 12 6 5
2 A B
3 # Epsilon
4 1.00 1.50
5 0.50
6 # Sigma
7 2.00 1.60
8 1.76
```

下面通过一个例子来演示 AIRSS 联合pp3的使用过程.

```
user@machine_name$ ls
Al.cell Al.pp
user@machine_name$ cat Al.cell
%BLOCK LATTICE_CART
2 0 0
0 2 0
0 0 2
%ENDBLOCK LATTICE_CART

%BLOCK POSITIONS_FRAC
Al 0.0 0.0 0.0 # Al1 % NUM=8
%ENDBLOCK POSITIONS_FRAC

#MINSEP=1.5
user@machine_name$ cat Al.pp
1 12 6 2.5
Al
# Epsilon
1
# Sigma
2
# Beta
1
user@machine_name$ airss.pl -pp3 -max 3 -seed Al
user@machine_name$ ls -l
Al-43867-3302-1.res
Al-43867-3302-2.res
Al-43867-3302-3.res
Al-43867-3302.cell
Al.cell
Al.pp
user@machine_name$
```

4.3 联合 CASTEP 弛豫

要使用 CASTEP 联合 AIRSS 计算, 首先需要将成功安装的 CASTEP 可执行文件 `castep.serial` 或 `castep.mpi` 复制到 AIRSS 的 `bin` 目录中, 并重命名为 `castep`.

使用 CASTEP 联合 AIRSS 计算时, 除了 `*.cell` 外需要准备 `*.param` 文件. `*.param` 是 CASTEP 的配置文件, 您可以在其中定义 CASTEP 计算过程中的必要配置参数, 包括, 计算的类型 (结构优化, 自洽, 光学性质计算, 能带计算等), 电荷, 自旋取向, 截断能, 收敛标准等. 该文件通常由若干行组成, 每一行包含一个 keyword 及其相应的赋值.

`*.param` 文件的内容主要有以下特点:

(1) 任何两个 keywords 之间没有书写顺序上的限制.

- (2) 您可以使用 # 或 ; 或 ! 又或者单词 COMMENT 来添加注释.
- (3) *.param中设定的所有的 keywords 和数据均不区分大小写, 同时, 任何标点符号 (除了标明注释内容的符号), 多余的空格和任何空行都将被自动忽略.
- (4) 文件的任何一行中最多只能出现一个 keywords 及其对应参数.

*.param文件每一行的基本格式均为:

[keywords] : [value]

其中的 ‘:’ 是为了书写美观便于区分内容所加, 程序实际执行时会自动忽略, 您也可以完全不加入这一符号转而在用空格代替.

CASTEP 中[keywords]的定义和使用方法, 可参考: [CASTEP cell keywords and data blocks](#).

AIRSS 默认联合 CASTEP 计算, 因此运行下述命令即可开始结构搜索.

4.4 联合 VASP 弛豫

AIRSS 自带了 `airss.pl --vasp` 选项用于联合 VASP 弛豫计算. 但官方自带的接口在有些情况下略显粗糙. 基于 AIRSS, 笔者用一套 bash 命令集重新编写了 AIRSS 至 VASP 的接口, 将其命名为 `airss4vasp(a4v)`. 后虽考虑过将此命令集使用 python 重新书写, 但 AIRSS 本身无法在 Windows 上运行, 而复写工程又过于庞大且收效甚微, 因此 a4v 目前仍然主要基于 bash 实现.

a4v 可以看做是 AIRSS 的改版, 无需安装原生 AIRSS 便可独立运行. 其主要基于 AIRSS 原生的 `buildcell` 模块, 同时内嵌了 PBS, NSCC, Slurm 等作业提交系统指令, 真正做到了一键提交 AIRSS+VASP 任务的功能. 其具体用法可参见项目内部的 `README.md` 文件.⁹

a4v 增加了对应原子位置弛豫固定的 SD-XYZ, SD-XY, SD-X 等原子标签, 同时删减了原先 `buildcell` 中设置共线自旋数值的全局参数关键字 `SPIN=`.

4.5 查看计算结果

AIRSS 的计算结果全部储存在了 .res 文件中. 这种 .res 结构文件最早在 [SHELX](#) 中使用. 由于一些历史原因被 CASTEP 和 AIRSS 复用. 由于 SHELX 本身是对 Windows 友好程序, 因此其书写格式也沿袭了部分 Windows 文档的特点, 如 REM 代表注释行, 文件使用 END 结尾等.

```
user@machine_name$ cat Al-43867-3302-1.res
TITL Al-43867-3302-2 0.0000000004 60.4852769773 -53.2712053113 0 0 8 (
P63/mmc) n - 1
REM
REM in /Users/alex/Documents/ProgramCode/MaterialCalculateProgram/AIRSS
REM /airss-0.9/examples/1.1
REM
REM
REM
REM
CELL 1.54180 2.2 5.2 5.2 86.6 90.0 90.0
LATT -1
SFAC Al
Al 1 0.2544637028970 0.9316224149716 0.6657635302849 1.0
Al 1 0.7544640475988 0.0982890099295 0.3324301203388 1.0
Al 1 0.2544640470150 0.3482890078890 0.5824301202379 1.0
```

⁹事实上, 该说明手册也在此项目中.(笑)

```

Al      1  0.2544640470479  0.8482890103459  0.0824301202324  1.0
Al      1  0.7544640476367  0.5982890097930  0.8324301190566  1.0
Al      1  0.7544637023180  0.1816224159838  0.9157635306900  1.0
Al      1  0.7544637024482  0.6816224143044  0.4157635299253  1.0
Al      1  0.2544637030384  0.4316224167828  0.1657635292340  1.0
END
user@machine_name$

```

AIRSS 输出的.res文件各行的含义如下:

- 第一行TITL中的第一项是软件分配给该结构的名称标签, 第二项是系统外加静水压 (GPa), 第三项是单胞体积, 第五项是每个单胞总的焓 (能量), 第六项是原子自旋值的平均值, 第七项是原子自旋绝对值的平均值, 第八项是体系所在空间群名称, 最后一项是固定字符 $n - 1$.
- 之后若干以REM开头的行为注释行, 记录了文件生成的基本信息, 删除后不会有任何影响.
- 紧接着以CELL开头的行记录了基本的晶胞信息. 其中, 第一项是一个无意义的小数, 这个小数在原来的 SHELX 程序中是用来记录得到相关结构所用 XRD 的波长, 在 AIRSS 中锁定为了一个无意义的小数. 第二至四项是晶格常数 $a b c$, 接下来五至七项是晶角 $\alpha \beta \gamma$.
- 下面一行是LATT -1. 这一行在 SHELX 中用于标定晶格的对称性, 在 AIRSS 中锁定为固定值-1.
- 接下来以SFAC开头的行记录了体系中全部的元素种类, 不同的元素用一个空格隔开.
- 最后若干行标定了单胞中原子的位置. 原子位置信息行的第一项是元素符号, 第二项指明了该元素在SFAC 行中是第几个出现的元素, 第三到五项是该行对应原子的分数坐标, 最后一项是该点位的占据数, 一般设为 1.
- 文件最终以END行结尾.

5 计算数据处理

有了计算数据.res文件后, 就可以使用ca指令进行数据处理了. ca 是对AIRSS 中基本分析套件 cryan 的封装.

```
user@machine_name$ ca
ca [-R(recursive)] [command line arguments for cryan]
user@machine_name$
```

cryan的使用方法如下:

```
user@machine_name$ cryan

Usage: cryan [OPTIONS]

The structures are read from STDIN, for example:

    cat *.res | cryan -s
    gunzip -c lots.res.gz | cryan -f H2O
    find . -name "*.res" | xargs cat | cryan -m

cryan options (note - the order matters):

-r, --rank                               Rank all structures, of any
    composition
-s, --summary                             Summary, most stable from each
    composition
-e, --enthalpy <length_scale>           Plot enthalpy vs. pressure,
    interpolate with <length_scale>
-f, --formula <formula>                 Select structures of a given
    composition
-fc, --formula_convert <formula>        Attempt to convert structure to
    this composition
-t, --top [num]                          Output top few results (default
    10)
-u, --unite <thresh>                     Unite similar structures
-dr, --distance <rmax>                   Distance threshold for structure
    comparison (default 20)
-de, --delta_e <energy>                  Ignore structures above energy (
    per atom)
-sd, --struc_dos <smear>                  Plot a structural density of
    states, smeared
-p, --pressure <pressure>                Additional pressure (default 0
    GPa)
-m, --maxwell                            Extract the stable compositions
-ph, --pressure_hull                      Extract the stable structures
    with pressure
-<n>                                       Component <n>
-xg, --xmgrace                           Plot output with xmgrace
-c, --compare <thresh> <structure>      Compare structure to all others
    --delete                             Delete unwanted structures
-g, --geometry [thresh]                  Calculate the atomic geometry for
    the structures (default 0.1)
-n, --num_units                          Only report structures with n
    separate units (default -1)
-d, --dimensionality                     Only report structures with
    dimensionality of d (default -1.0)
-cl, --cluster                           No periodic boundary conditions
-bl, --bondlength                        Maximum bond length (default 0.0,
    negative for modularity)
-bs, --bondscale                         Bond length scaling (default 1.0,
    negative for modularity)
```



```

-dm, --deltamodularity          Modularity bias parameter
-wt, --weight                    Weight the adjacency matrix
    toward short contacts
-ns, --notsymm                  Do not calculate point group of
    clusters
-sc, --struct_comm <thresh>     Determine the community structure
-cm, --community                Output the community structure
-am, --adjacencymatrix          Output the adjacency matrix
-x, --xyz                       Output clusters in XYZ format
-o, --off                       Output polyhedra in OFF format
-al, --alpha                    Construct alpha shapes
-l, --long                      Long names for structures
-h, --help, -?                 Print usage information and exit

user@machine_name$

```

使用ca就可以对之前的计算结果进行分析.

```

user@machine_name$ ca -r > analysis.data
user@machine_name$ cat analysis.data
Al-43867-3302-2    0.00  7.561  -6.659   8 Al  P63/mmc  1
Al-43867-3302-1   -0.00  7.561   0.000   8 Al  P63/mmc  1
Al-43867-3302-3    0.00  7.564   0.005   8 Al  Fm-3m   1
user@machine_name$

```

上述结果: 第一列是 AIRSS 软件分配给该结构的名称标签, 第二列压力值 (GPa), 第三列是每个化学式结构单元 (fu) 的体积, 第四列第一行是一个化学式结构单元 (fu) 的焓值, 之后的几行是不同结构下相对于第一行的焓值, 第五列是单胞中化学式结构单元 (fu) 的总个数 (单胞中 fug 的个数乘以一个 fug 中 fu 的个数.), 第六列是化学式结构单元 (fu) 的化学式, 第七列是空间群名称, 第八列是所有搜索结果中出现该结构的次数.

如果您认为所列结果过多, 可以使用-u选项, 但是要注意, -u一定要排在-r之前使用.¹⁰

```

user@machine_name$ ca -u 0.01 -r > analysis.data
user@machine_name$ cat analysis.data
Al-43867-3302-2    0.00  7.561  -6.659   8 Al  P63/mmc  2
Al-43867-3302-3    0.00  7.564   0.005   8 Al  Fm-3m   1
user@machine_name$

```

指令 ca -u 后所跟的数字是一个无量纲的比例. 可以将这一参数简单的做如下理解: 他标定了晶格相似度阈值. 该数值越大, 容忍度越高, 最终展示出来的不同结构越少. 更详细的, 这一参数标定的距离, 是原子结构内部最接近的两个原子之间距离的小数倍. 例如, 现有两个结构十分相似, 晶格中最短键长为 1.5 Å, 则ca -u 0.1 就意味着, 比较全部的两结构对应原子的两两距离, 如果未能发现这些距离差存在大于 0.15 Å 的情况, 则标定这两个结构一致.¹¹ 该值可根据需求调整, 建议在 0.1-0.01 之间选择.

¹⁰这可以算是原程序的 BUG, 已在 airss4vasp 中修正.

¹¹详细见源码 “airss-0.9.1/src/cryan/src/cryan.f90”2133 行

A 软件安装

下面将以 `airss-0.9.0` 版本为例, 简要记录 AIRSS 的安装.

AIRSS 只支持在命令行 (Command Line) 使用, 且仅能安装在 *nix 系统中. 安装此软件前, 您最好已经了解 `GNU make` 的使用方法. 当然, 如果您实在对此不感兴趣, 这不是必须的. 前提是您能完全按照以下步骤操作.

A.1 软件主体安装

具体的安装分为以下几步, 非必须步骤已使用*标出:

- (I) 建立安装包文件管理系统** 在开始一切安装之前, 建议作为非root用户但是有sudo权限的您: 在自己能进行任意操作的家目录~/中建立一个安装包管理文件夹, 如~/install_package; 同时在系统目录/usr/local中建立一个存放 airss 和其他程序二进制可执行文件的目录, 如usr/local/airss-0.9/bin.

之所以这样建议, 是为了减少您安装过程中在系统目录下需要进行的操作, 降低由此可能引发的事故的概率, 同时让安装过程更简洁 (避免每个命令都要使用前缀sudo ..., sudo sh -c "...>...").

当然, 您也可以完全不将软件安装在系统目录, 一切都凭您的个人喜好.

```
user@machine_name$ cd /usr/local/
user@machine_name$ sudo mkdir -p airss-0.9/bin
Password:
user@machine_name$ cd airss-0.9
user@machine_name$ ls -F
bin/
user@machine_name$ cd ~
user@machine_name$ mkdir -p install_package/AIRSS
user@machine_name$ cd install_package
user@machine_name$ ls -F
AIRSS/
```

- (II) AIRSS 安装包下载** AIRSS 软件包下载地址为:<https://www.mtg.msm.cam.ac.uk/files/airss-0.9.tgz> 或者, 您可以访问前文所述[官方网站](#)详细了解相关信息后下载.

您可以选择在浏览器上下载, 也可以使用wget指令.

```
user@machine_name$ wget -P ~/Downloads https://www.mtg.msm.cam.ac.uk/files/airss-0.9.tgz
```

- (III) 拷贝并解压安装包** 将您下载的airss-0.9-2.tag拷贝到安装包管理文件夹中, 并使用tar解压.

```
user@machine_name$ cd AIRSS
user@machine_name$ cp ~/Downloads/airss-0.9-2.tgz .
user@machine_name$ tar -zxvf airss-0.9-2.tgz
x airss-0.9/.hg_archival.txt
x airss-0.9/.hgignore
x airss-0.9/LICENCE
x airss-0.9/README
x airss-0.9/VERSION
...
...
```

(IV) 使用 GNU make 指令安装 AIRSS 使用make等指令安装编译安装 AIRSS.

```
user@machine_name$ cd airss-0.9
user@machine_name$ make
(cd src/pp3/src; make)
gfortran -O3 -c ../../common/constants.f90
gfortran -O3 -c cell.f90
gfortran -O3 -c pp.f90
gfortran -O3 -c opt.f90
gfortran -O3 -c pp3.f90
...
...
user@machine_name$ make install > make_install.log 2>&1
user@machine_name$
user@machine_name$ cat make_install.log
(cp src/pp3/src/pp3 bin/)
(cp src/cabal/src/cabal bin/)
(cp src/buildcell/src/buildcell bin/)
(cp src/cryan/src/cryan bin/)
user@machine_name$
```

十分鼓励您今后使用make isntall指令时, 将其输出重定向到一个记录文件中, 这样会给您卸载软件时提供便利.

*(V) 安放可执行文件 ~/install_package/AIRSS/airss-0.9/bin存放了安装完毕的可执行文件, 将其拷贝至系统目录下.

```
user@machine_name$ sudo cp -r bin/ /usr/local/airss-0.9/bin
Password:
user@machine_name$ ls /usr/local/airss-0.9/bin
airss.pl      cabal          cell2lammps   crud.pl
despawn       gulp_relax    mc            pp3_relax    psi4_relax
spawn-slow    tidy.pl       buildcell     castep2res   lammps2cell
check_airss   cryan         gap_relax     lammps2cell
niggli        press         run.pl        stopairss
ca            castep_relax  comp2minsep   csymm
gencell       lammps_relax  pp3           prim
spawn        symm
```

(VI) 设置系统环境变量 完成以上所有设置后, 您实际上就可以通过使用使用命令/usr/local/airss-0.9/bin/airss.pl -[option] [parameter] ...来运行 AIRSS 了. 为了简便, 可以考虑在~/.bash_profile文件中加入如下内容

```
###Setting PATH for AIRSS
export PATH="/usr/local/airss-0.9/bin:${PATH}"
```

修改储存并退出后, 请重新登入终端, 或运行source指令完成环境变量的更新.

```
user@machine_name$ source ~/.bash_profile
```

这样您就可以在系统中的任何路径上执行airss.pl等 AIRSS 的指令了.

(VII) 检查安装情况 设置好环境变量后, 您可以在~/install_package/AIRSS/airss-0.9/下输入make check指令检查 AIRSS 安装情况.

```

user@machine_name$ make check
(sh bin/check_airss)
Essential:

airss.pl +
run.pl +
crud.pl +
castep2res +
buildcell +
cryan +
pp3 +
cabal +
cellsym - Install cellsym: http://www.tcm.phy.cam.ac.uk/sw/check2xsf/cellsym.html
symmol - Patch and install symmol: http://www.ccp14.ac.uk/ccp/web-mirrors/symmol/~pila/symmol.zip
bob - Get Bob!

Recommended:

castep - Install castep: http://www.castep.org/
optados - Install optados: http://www.tcm.phy.cam.ac.uk/~ajm255/optados/index.html
qhull - Install qhull from package manager, or:
http://www.qhull.org/
qconvex - Install qhull from package manager, or:
http://www.qhull.org/
xmgrace - Install grace from package manager or:
http://plasma-gate.weizmann.ac.il/Grace/
Rscript - Install R/Rscript and ggtern from package manager
or: https://cran.r-project.org/

Optional:

gulp - Install gulp: http://projects.ivec.org/gulp/
cif2cell - Install cif2cell from: http://cif2cell.sourceforge.net/

Very optional:

lammps - Install lammps: http://lammps.sandia.gov/
hull - Install hull: http://www.netlib.org/voronoi/hull.html
off_util - Install antiprism: http://www.antiprism.com/files/antiprism-0.24.1.tar.gz

Pseudopotentials:

pspot - set $PSPOT_DIR to location of the CASTEP pspot
directory

Spawn file:

.spawn -

-----
Tests run in .check:
-----

Running example 1.1 (Crystals):
Al-9002-4643-1    -0.00    7.561    -6.659    8 Al    n/a    1

```

```

A1-9002-4643-2    0.00    7.564    0.005    8 A1    n/a    1

Running example 1.2 (Clusters):

A1-9274-4255-2    0.00    615.385  -3.014    13 A1    n/a    1
A1-9274-4255-1    0.00    615.385    0.019    13 A1    n/a    1

Skipping example 3.1 (Gulp)
Skipping example 2.1a (Castep)

```

如果您仔细阅读了上述输出文件, 会发现必要的组件中还有cellsym和symmol没有安装. 这直接导致了晶体和团簇空间群符号输出为n/a.

A.2 辅助插件安装

AIRSS 支持的全部插件信息可查询~/install_package/AIRSS/airss-0.9/README文件. 下面只演示最核心的cellsym和symmol插件的安装过程.

(I) 下载插件安装包 cellsym的安装包官方网站是:

<http://www.tcm.phy.cam.ac.uk/sw/check2xsf/cellsym.html>

需要注意的是, cellsym源码是使用 C 语言编写的, 安装此程序前, 需要下载并安装库文件spglib.h.

spglib.h的下载地址是:

<http://www.tcm.phy.cam.ac.uk/sw/check2xsf/spglib-1.9.4.tar.gz>

cellsym的下载地址是:

<http://www.tcm.phy.cam.ac.uk/sw/check2xsf/cellsym.tgz>

symmol插件安装包的下载地址是:

<http://www.ccp14.ac.uk/ccp/web-mirrors/symmol/~pila/symmol.zip>

您可以通过浏览器下载上述文件, 也可以使用wget指令下载.

```

user@machine_name$ wget -P ~/Downloads
www.tcm.phy.cam.ac.uk/sw/check2xsf/spglib-1.9.4.tar.gz
www.tcm.phy.cam.ac.uk/sw/check2xsf/cellsym.tgz
www.ccp14.ac.uk/ccp/web-mirrors/symmol/~pila/symmol.zip

```

(II) 拷贝并解压插件 将您下载的三个压缩包拷贝到安装包管理文件夹中, 并使用tar和unzip解压.

```

user@machine_name$ cd ~/Downloads
user@machine_name$ cp cellsym.tar spglib-1.9.4.tar
symmol.zip ~/install_package/AIRSS
user@machine_name$ cd ~/install_package/AIRSS
user@machine_name$ tar -xvf cellsym.tar
...
user@machine_name$ tar -xvf spglib-1.9.4.tar
...
user@machine_name$ unzip symmol.zip -d symmol
...
user@machine_name$ ls -F
airss-0.9/      airss-0.9-2.tgz  cellsym-0.16a/
cellsym.tar     spglib-1.9.4/    spglib-1.9.4.tar
symmol/         symmol.zip

```

(III) 编译插件 将解压好的插件按如下顺序操作.

首先安装库文件 spglib. 使用 GNU make 指令.

```
user@machine_name$ cd spglib1.9.4/
user@machine_name$ ./configure
...
user@machine_name$ make
...
user@machine_name$ sudo sh -c 'make install > make_install.log
2>&1'
Password:
user@machine_name$ cat make_install.log
Making install in src
../install-sh -c -d '/usr/local/lib'
/bin/sh ../libtool --mode=install /usr/bin/install -c
  libsymspg.la '/usr/local/lib'
libtool: install: /usr/bin/install -c .libs/libsymspg.0.dylib /usr
/local/lib/libsymspg.0.dylib
libtool: install: (cd /usr/local/lib && { ln -s -f libsymspg.0.
dylib libsymspg.dylib || { rm -f libsymspg.dylib && ln -s
  libsymspg.0.dylib libsymspg.dylib; }; })
libtool: install: /usr/bin/install -c .libs/libsymspg.lai /usr/
local/lib/libsymspg.la
libtool: install: /usr/bin/install -c .libs/libsymspg.a /usr/local
/lib/libsymspg.a
libtool: install: chmod 644 /usr/local/lib/libsymspg.a
libtool: install: ranlib /usr/local/lib/libsymspg.a
/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault
.xctoolchain/usr/bin/ranlib: file: /usr/local/lib/libsymspg.a(
debug.o) has no symbols
../install-sh -c -d '/usr/local/include/spglib'
/usr/bin/install -c -m 644 arithmetic.h cell.h debug.h delaunay.h
  hall_symbol.h kgrid.h kpoint.h mathfunc.h niggli.h pointgroup.
  h primitive.h refinement.h site_symmetry.h sitesym_database.h
  spacegroup.h spg_database.h spglib.h spin.h symmetry.h version
  .h '/usr/local/include/spglib'
make[2]: Nothing to be done for `install-exec-am'.
make[2]: Nothing to be done for `install-data-am'.
user@machine_name$
user@machine_name$
user@machine_name$ make install check
...
...
...
PASS: spglib_test
=====
Testsuite summary for spglib 1.9.4
=====
# TOTAL: 1
# PASS: 1
# SKIP: 0
# XFAIL: 0
# FAIL: 0
# XPASS: 0
# ERROR: 0
=====
make[1]: Nothing to be done for `check-am'.
user@machine_name$
```

使用make install check检查 PASS 后, 就可以开始编译cellsym了.

```
user@machine_name$ cd ../cellsym-0.16a/
```

```

user@machine_name$ make
...
user@machine_name$ ls -all cellsym
-rwxr-xr-x 1 user groups 53628 Jan 25 12:28 cellsym
user@machine_name$

```

顺利编译完成后, 会生成一个名为cellsym的可执行文件. 注意, make执行过程中可能会出现编译警告, 但这并不影响程序执行, 可忽略.

编译并确认生成了cellsym文件后, 就可以开始编译另一个插件symmol了. symmol是使用 Fortran 写成的. 在网站上下载的是其源码, 需要编译使其变为可执行文件. 需要注意的是, 原版的symmol.f并不兼容 AIRSS, 需要为其打上~/install_package/AIRSS/airss-0.9/misc中提供的symmol.patch补丁.

```

user@machine_name$ cd ../airss-0.9/misc/
user@machine_name$ cp ../../symmol/symmol.f .
user@machine_name$ ls
symmol.f      symmol.patch
user@machine_name$ patch -p0 symmol.f symmol.patch
patching file symmol.f
user@machine_name$ gfortran symmol.f -o symmol
user@machine_name$ ls
symmol      symmol.f      symmol.patch
user@machine_name$ echo '-o 后跟的文件名一定要是 symmol'
-o 后跟的文件名一定要是 symmol
user@machine_name$ ls -all symmol
-rwxr-xr-x 1 user group 106800 Jan 25 12:41 symmol
user@machine_name$

```

至此, 我们完成了所有插件的编译. 生成了symmol和cellsym两个可执行文件.

(IV) **将插件导入 AIRSS** 这一步的操作十分简单, 将编译好的两个插件复制到系统目录下的bin/文件夹即可. 为了以防万一, 可以在安装包管理文件夹保存一个bin/的备份

```

user@machine_name$ pwd
/home/user_name/install_package/AIRSS/airss-0.9/misc
user@machine_name$ cp symmol ../bin/
user@machine_name$ sudo cp symmol /usr/local/airss-0.9/bin
Password:
user@machine_name$ cd ../../cellsym-0.16a/
user@machine_name$ cp cellsym ../airss-0.9/bin/
user@machine_name$ sudo cp cellsym /usr/local/airss-0.9/bin

```

(V) **安装最终检查** 回到airss-0.9中执行make的文件夹. 重新输入make check检查安装情况.

```

user@machine_name$ cd ../airss-0.9
user@machine_name$ make check
(sh bin/check_airss)
Essential:

airss.pl +
run.pl +
crud.pl +
castep2res +
buildcell +

```

```

cryan +
pp3 +
cabal +
cellsym +
symmol +
bob - Get Bob!

Recommended:

castep - Install castep: http://www.castep.org/

...
...
...

-----
Tests run in .check:
-----

Running example 1.1 (Crystals):

Al-14776-403-2  -0.00   7.784  -6.398   8 Al   C2/m   1
Al-14776-403-1   0.00   7.820   0.066   8 Al   P21/m   1

Running example 1.2 (Clusters):

Al-15054-7410-1  0.00   615.385  -3.190  13 Al   Cs     1
Al-15054-7410-2  0.00   615.385   0.006  13 Al   Cs     1

Skipping example 3.1 (Gulp)
Skipping example 2.1a (Castep)
user@machine_name$

```

成功输出了晶体的空间群名称!

至此, 我们完成了 AIRSS 的基本安装, 您现在已经可以使用 AIRSS 的 pp3 模块 (默认是 CASTEP) 进行结构搜索了.

AIRSS 是受 GPL 许可证保护的开源软件. 对此程序您有以下三种权利:

- * 以任何目的运行此程序
- * 再复制
- * 改进此程序, 并公开发布改进

A.3 卸载软件

AIRSS 卸载可分为三步:

(I) **卸载 spglib** 进入安装包管理文件夹, 使用 `make uninstall` 卸载 spglib.

```

user@machine_name$ cd ~/install_package/AIRSS/spglib-1.9.4
user@machine_name$ sudo make uninstall
...
user@machine_name$

```

(II) **删除相关文件夹** 删除系统目录中的 bin 文件. 您可以选择保留安装文件. 保留安装文件可以在您试图恢复使用 AIRSS 时提供便利.¹²

¹²强烈建议您对文件进行删除时, 在离此文件较近的路径上操作, 并杜绝使用绝对路径, 以免打出文章开头提到的毁灭性指令.


```
user@machine_name$ cd /usr/local/  
user@machine_name$ sudo rm -ri airss-0.9  
Password:  
user@machine_name$ cd ~/install_package/  
user@machine_name$ rm -r AIRSS
```

(III) 恢复 PATH 变量 进入~/.bashrc文件, 删除修改环境变量的语句即可.

```
###Setting PATH for AIRSS  
export PATH="/usr/local/airss-0.9/bin:${PATH}"
```