

# AIRSS 指南

Yang Li

lyang.1915@gmail.com

2018.1.21 — 2018.1.27

## Contents

<b>1 一些简单的说明</b>	<b>1</b>
<b>2 计算准备</b>	<b>2</b>
2.1 Linux 系统操作	2
2.2 软件安装	2
2.2.1 软件主体安装	2
2.2.2 辅助插件安装	5
2.2.3 联合 CASTEP	9
2.2.4 卸载软件	9
2.3 文件准备	10
2.3.1 .cell 文件的结构	10
2.3.2 .param 文件的结构	17
2.3.3 .pp 文件的结构	20
<b>3 计算指令</b>	<b>20</b>
<b>4 计算数据处理</b>	<b>22</b>
<b>5 附录</b>	<b>23</b>
5.1 未说明的编译指示	23

## 1 一些简单的说明

AIRSS(Ab Initio Random Structure Searching) 是一款由英国剑桥大学 Chris J Pickard 教授等人自主开发的材料结构搜寻软件. 该软件是开源的, 且受 **GPL2 许可证** 保护. 访问其官方网站便可获取 AIRSS 安装包源码: <https://www.mtg.msm.cam.ac.uk/Codes/AIRSS>.

所谓结构搜寻是指, 对于一原子位置甚至是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, 元素组成及配比等) 限制下, 广泛地猜测其结构并计算其能量, 最终取所猜测的能量较低的几个结构作为进一步研究的对象的过程. 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, 需要使用一套成熟的结构搜寻软件, 系统且巧妙地捕捉体系结构.

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 [有关的 YouTube 视频](#). 另外两个较为常用的结构搜寻软件分别是 **USPEX** 以及 **CALYPSO**.

以下内容算不上指南或教程, 仅仅是学习 AIRSS-0.9 的一些记录. 由于是初学者, 身边又恰好没有人能熟练使用此软件, 再加上网络或是官方上相关说明文档或教程匮乏, 因此, 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不断尝试摸索所得. 受各种因素限制, 理解和解释上的错误或不可避免.

## 2 计算准备

### 2.1 Linux 系统操作

阅读本记录前, 需要您对 Linux 操作系统及其相关指令有一定的了解. 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 软件安装

下面将以 airss-0.9 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. 前提是你能完全按照以下步骤操作.

#### 2.2.1 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 GNUmake 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 install`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.(即使这里的 `AIRSS` 并不必要这样做.)

**\*(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
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
Al-9002-4643-2     0.00    7.564     0.005    8 Al    n/a    1

Running example 1.2 (Clusters):

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

### 2.2.2 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 GNUmake 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) 进行结构搜寻了.

另外, 值得一提的是, 上述所有软件 (AIRSS 及其两个插件) 都是开源的, 且受 GPL 许可证保护. 对此程序, 您有以下三种自由权利:

- \* 以任何目的运行此程序的自由
- \* 再复制的自由
- \* 改进此程序, 并公开发布改进的自由 (前提是能得到源代码)

尤其是 AIRSS 源码的buildcell模块的symmetry.f90中, 有 230 个空间群的 Wyckoff 点信息, 或许对您会有所帮助.

### 2.2.3 联合 CASTEP

要使用 CASTEP 联合 AIRSS 计算, 只需要成功安装 CASTEP, 且将 CASTEP 可执行文件路径添加至系统环境变量\${PATH}中, 而后将其安装文件夹中的castep.serial或者castep.mpi重命名 (或复制) 为castep即可.

### 2.2.4 卸载软件

卸载 AIRSS 是十分简单的, 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.

```
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
```

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.

(III) 恢复 PATH 变量 进入~/.bash\_profile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}"
```

执行完上述指令后, AIRSS 就完全被卸载了. 要恢复使用, 请跳至本节的开头.

## 2.3 文件准备

在使用 AIRSS 预测结构时, 一般需要准备一个扩展名为 `.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.

一般推荐 AIRSS 结合 CASTEP 使用, 此时需要准备两种输入文件: `.cell` 文件与 `.param` 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).

下面详细介绍 `.cell` 文件与 `.param` 文件的设置.

### 2.3.1 `.cell` 文件的结构

`.cell` 文件相当于结构搜寻的种子文件, 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) \*.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.
- (2) \*.cell 文件主要由两部分组成: “数据区块” 和 “编译指示”. 相应的, \*.cell 中的 *Keywords* 也可以分为上述两类.
- (3) “数据区块” 和 “编译指示” 没有书写顺序上的限制.
- (4) \*.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.
- (5) 文件的一行中只能出现一个 *Keywords* 及其对应参数.
- (6) “数据区块” 的设置需要遵循一定的格式.(稍后具体介绍)
- (7) 未出现在 \*.cell 文件中的 *Keywords* 都将被设为默认值.

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$
```

该文件结构和语法十分类似于 CASTEP 的 `.cell` 文件. 或者说 AIRSS 中的 `.cell` 文件就是 CASTEP 中结构文件的变形.<sup>1</sup>

文件前 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`. 这是 “cell lattice vectors in Cartesian coordinates” 的缩写, 也即, 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

<sup>1</sup> 甚至可以直接使用 `VESTA` 打开 AIRSS 的 `.cell` 文件而不报错

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 结构搜寻种子文件中有效的相关参数, 或更官方的, 有效的编译指示 (Pragmas). `# MINSEP = 1.5` 表明了两原子间距不得低于  $1.5\text{\AA}$ .

可以看到, `.cell` 文件的结构大致分为两大部分: **数据区块**和**编译指示**

### 数据区块

首先声明, 由于从官方提供的例子中能提取到的信息十分有限, 从此处开始, 部分结论是分析源码之后得到的, 认识上的错误可能在所难免, 一切请以实践为准.

该部分沿用了 CASTEP 的结构文件中定义数据的模式, 但在细节上又会有所不同. 数据区块的具体模式如下所示:

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

可以在 AIRSS 中使用的 `[keywords]` 已在表1中列出.<sup>2</sup>

CASTEP 中其他 `[keywords]` 的定义和使用方法, 可参见 [CASTEP cell keywords and data blocks](#).

一般进行结构搜寻时, 只需要在 AIRSS 的 `.cell` 文件中给出**晶格参数 (单胞基矢)** 和**原子坐标**的数据即可.

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` 称为晶格标记 (Lattice Tags).

它声明了晶格常数在搜寻过程中是不能改变的. 这里的搜寻过程单指猜测结构这一步, 不包括使用 CASTEP 或 pp3 进行结构粗略优化的过程. 如果您想确保粗略结构优化时体系晶格常数也不变, 还需要在编译指示区域加上 `FIX_ALL_CELL:TRUE` 这一句. 同时, 将晶格参数贴上 `#FIX` 标签还会引起一些“副作用”, 我们将在之后做详细说明. 其他晶格标记还有 `#CFIX` 和 `#ABFIX`, 他们的意义是显然的.

下面介绍**原子坐标**的设置方法, 以 `POSITIONS_FRAC` 为例.<sup>3</sup>

---

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
```

---

<sup>2</sup>因为时间有限, 源码并未看完, 目前只发现了 AIRSS 支持这几个关键词的源码

<sup>3</sup>例子中的物质可能并不存在, 只是作为一种演示.

Table 1: CASTEP BOLCK keywords In AIRSS

名称	功能
LATTICE_CART	使用笛卡尔坐标系, 以一个 $3 \times 3$ 的矩阵定义 <b>单胞</b> 基矢.
LATTICE_ABC	以一个 $2 \times 3$ 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.
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.
SYMMETRY_OPS(不常用)	定义单胞的对称性信息, 四行一组, 前三行定义旋转操作, 第四行行定义平移操作. 对于一个确定的点, 有几组对称性操作, 就意味着有几个等效点.(不常用, 具体用法可参考 <b>CASTEP:SYMMETRY OPS</b> .)
SPECIES_POT(不常用)	定义赝势文件的位置和名称, 每行的元素的顺序要和POSITIONS_*中的一致. AIRSS 和 CASTEP 联合运行时可以使用的数据块. 具体用法可参考 <b>CASTEP:SPECIES POT</b> .
HUBBARD_U(不常用)	This data block defines the Hubbard U values to use for specific orbitals. 更具体的信息请参考 <b>CASTEP:HUBBARD U</b> .

```

5 | O    0.1 0.1 0.1 # O2 % NUM=1 POSAMP=0 UNMOVE
6 | H    0.3 0.3 0.6 # My_H
7 | H    0.0 0.0 0.0 # H2 % POSAMP=0 FIX
8 | %BLOCKEND POSITIONS_FRAC

```

通过这个例子, 我们看到, 原子坐标数据区块的基本格式是:

```
[symbol] [x] [y] [z] # [label] % [tag1] [tag2] [tag3]
```

每一行的第一列是元素名称, 二三四列是坐标, #号后的第一列是此元素的标签, 元素标签可以设置成任意字符, %号之后的各个列都是原子标记 (Atom Tags).

**表2**是各个原子标记的具体说明.(如不特殊指出, 表中长度物理量均以埃(Å) 为单位.)

最后, 再回到**晶格标记**, 下面将试图说明使用晶格标记#FIX时要十分小心的一点.

拿出官方例程 1.1 来说明这个问题.

#### 官方例程 1.1

```

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

```

以下例 3 中的内容和官方例程 1.1 完全等效.

#### 例 3.

```

1 | %BLOCK LATTICE_CART
2 | 2.52 0 0
3 | 0 2.52 0
4 | 0 0 2.52
5 | %ENDBLOCK LATTICE_CART
6 |
7 | %BLOCK POSITIONS_FRAC
8 | Al 0.0 0.0 0.0 # Al1 % NUM=4
9 | Al 0.0 0.0 0.0 # Al2 % NUM=4
10 | %ENDBLOCK POSITIONS_FRAC
11 |
12 | #MINSEP=1.5

```

您可能会觉得 2.52 这个数字很奇怪, 但这其实是  $2 \times \sqrt[3]{2} = 2.52$ .

为了说明上的方便, 我们声明三个概念:<sup>4</sup>

**原子堆垛 (atomic package, ap)**: POSITIONS\_FRAC数据中第一列元素符号 (同种元素或不同种元素) 简单罗列成的整体.(也可认为是忽略所有行的NUM标记后得到的整体)

**化学式结构组 (formula unit groups, fug)**: POSITIONS\_FRAC所定义的所有原子组成的整体.

<sup>4</sup>这都是笔者胡诌的名字, 为了之后方便解释.

Table 2: AIRSS Cheat Sheet – Atom Tags

名称及使用格式	功能
POSAMP=	定义该行元素结构搜寻过程中所能移动的最大距离.
ANGAMP	(需要进一步解读源码)
MINAMP=	定义该行元素结构搜寻过程中偏离初始位置的最小距离.
ZAMP=	在 z 轴方向上移动的最大距离, 一般用在层状结构的分析中. 其他两个方向也可设置标记XAMP和YAMP.
NUM= $n$ or $n_{min} - n_{max}$	定义该行原子在单胞中的个数, 缺省为 1.
ADATOM	表明该行原子是等前面所有原子结构优化结束后, 后加入的原子.
NOMOVE	一般进行结构搜寻要分两步, 首先预测一个结构, 而后粗略结构优化求其焓值.NOMOVE标签表明该原子结构预测时分数坐标不能改变, 粗略结构优化时可以改变. 可以与POSAMP=0联用以确保 AIRSS 不会调整原子坐标初始设定值.
FIX	表明该原子预测时分数坐标不能改变, 粗略结构优化时绝对坐标 (笛卡尔坐标) 不能改变. 可以与POSAMP=0联用以确保 AIRSS 不会调整原子坐标初始设定值.
RAD	(需要进一步解读源码)
OCC	(需要进一步解读源码)
MULT	(需要进一步解读源码)
PERM	(需要进一步解读源码)
COORD	(需要进一步解读源码)
NN	(需要进一步解读源码)

**化学式结构单元 (formula units, fu):** 所研究物质化学式的最简配比. 可由 fug 化简得到.

表3是例 2. 和例 3. 以及官方例程 1.1 关于上述三个概念的具体形式:

Table 3: Example for AP,FU and FUG

名称	例 2.	例 3.	官方例程 1.1
原子堆垛 (Atomic Package)	$MgAlO_2H_2$	$2Al$	$Al$
化学式结构组 (Formula Unit Groups)	$Al_2Mg_4O_2H_2$	$8Al$	$8Al$
化学式结构单元 (Formula Unit)	$AlMg_2OH$	$Al$	$Al$

再回到例 3. 与官方例程 1.1. 仔细研究这两个文件后, 不难发现, 要使二者的内容是等价的, LATTICE\_CART中所描述的体积必然是一个原子堆垛 (ap) 的体积. 而不是一个化学式结构组 (fug) 或者化学式结构单元 (fu) 的体积.

事实上, 实际操作时, AIRSS 生成了两个变量, 一个是targvol, 用来存储LATTICE\_CART中矩阵行列式的值; 另一个是scale\_vol, 用来储存原子总数 (fug 中原子个数) 与原子堆垛中粒子的个数 (ap 中原子个数, 简单来说就是POSITIONS\_FRAC中数据行数) 的比值. 并将上面两个变量的乘积作为单胞的总体积. 当然, 这里的原子总数不只需要乘以设定的NUM个数, 还要乘以由对称性所引起的等效点的个数. 实际程序的逻辑较为复杂, 限于篇幅, 不再做更详细的解释.

然而, 当给LATTICE\_CART贴上#FIX的标签后, 上面定义的两个变量会全部失效, 所研究体系的总体积直接等于LATTICE\_CART中矩阵的行列式. 这意味着, 晶格标记#FIX, 不仅仅强制搜索过程中不得改变晶格常数数值, 他还会直接把LATTICE\_CART中定义的体积直接锁定为整个单胞 (或者团簇) 的体积.

如果不明白这一点, 对比官方例程例 1 和例 2 的结果及.cell文件, 就会让您感觉十分疑惑.(怎么LATTICE\_CART中定义的体积一会儿是单原子的一会儿是整体的?)

另外, 再次强调, 如果要确保晶格参数不变, 除了在晶格参数数据区块加入#FIX, 以保证在猜测晶格结构时保持其为常值外, 还需要在编译指示中添加FIX\_ALL\_CELL : true这样一句, 以保证使用 CASTEP 结构优化时也不会改变晶格参数.

**同时注意**, 原子标记和晶格标记中都有FIX, 但二者的功能和使用格式显然是不同的.

至此, 完成了对数据区块设置的讨论.

**编译指示** 编译指示是.cell文件中指明结构搜寻过程中应遵守的条件的语句. 原 CASTEP 中包含的编译指示均可在 AIRSS 中使用. 同时, AIRSS 另外添加了若干只能由 AIRSS 识别的编译指示. 这部分额外的编译指示均以#开头.

有些编译指示与原子标记有相同的作用, 此时, 原子标记的优先级要高于编译指示.

**部分编译指示详细的使用方法和功能如表4所示.**

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

```

Table 4: AIRSS Cheat Sheet – Pragma

名称及使用格式	功能
#NFORM = $n$ or $n_{min}-n_{max}$	定义单胞中化学式结构组 (fug) 的个数.(?? 不确定??)
#SUPERCELL $n$ or $a\ b\ c$ or $a_x\ a_y\ a_z\ b_x\ b_y\ b_z\ c_x\ c_y\ c_z$	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 $a\ b\ c$ , 超胞与单胞晶格基矢之间的变换矩阵 $a_x\ a_y\ a_z\ b_x\ b_y\ b_z\ c_x\ c_y\ c_z$
#SLAB	声明超胞生成过程中, 不向 $z$ 方向上叠加单胞.
#CLUSTER	声明要搜寻的体系其实是团簇而不是晶体
#SYMMOPS = $n$ or $n_{min}-n_{max}$	搜寻晶体结构的对称群范围
#POSAMP = $n$	与原子标记中的定义相同, 结构搜寻中原子偏离初始点的最大位置.
#MINAMP = $n$	与原子标记中的定义相同, 结构搜寻中原子偏离初始点的最小位置.
#ZAMP = $n$	与原子标记中的定义相同, 结构搜寻中原子 $z$ 方向偏离初始点的最大振幅.
#ANGAMP = $\theta$	(需要进一步解读源码)
#MINSEP = $n$ or $n\ X-X = n_{X-X}\ X-Y = n_{X-Y} \dots$	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 Li-Ge=2.81
FIX_ALL_CELL:true	声明 CASTEP 结构优化晶体时不得改变其晶格常数.
KPOINTS_MP_SPACING 0.07	使用 MP 方法获取 $k$ 点时, $k$ 点间隔 0.07Å
#VARVOL	定义原子堆垛 (ap) 的体积, 如果与LATTICE_*同时出现, 则优先考虑#VARVOL所定义的体积.
#TARGVOL = $v$ or $v_{min} - v_{max}$	定义原子堆垛的体积, 使用该方法定义的体积结构搜寻过程中, 会保持所设的体积不变.
#SPECIES=[symbol1]#[Tag1] [Tag2] ..., [symbol2]#[Tag1] [Tag2] ..., [symbol3]...	使用简化记号定义体系原子组分, 不能与POSITIONS_*同时出现.
#NATOM = $n$ or $n_{min} - n_{max}$	与#SPECIES联用, 定义一个化学式结构组 (fug) 中有总共多少原子, 一般用于变组分分析 (变胞预测). 且使用此编译指示会使#SPECIES指令包含的%后的原子标记全部失效. <sup>5</sup> 同时, 使用此指令会使之前定义的原子堆垛体积自动变为单个原子的平均体积. <sup>6</sup>
#SLACK	(需要进一步解读源码)
#OVERLAP	(需要进一步解读源码)
#COMPACT	(需要进一步解读源码)
SYMMETRY_GENERATE	使用 CASTEP 结构优化前, 首先探测该体系的对称性.
SNAP_TO_SYMMETRY	CASTEP 结构优化过程中, 强制晶格参数和原子位置变化符合晶体对称性.



```

6 | %ENDBLOCK LATTICE_CART
7 | %BLOCK POSITIONS_FRAC
8 | A1 0.0 0.0 0.0 # A11 % NUM=7-13
9 | %ENDBLOCK POSITIONS_FRAC
10 | FIX_ALL_CELL : true
11 | #MINSEP=1.5
12 | #CLUSTER
13 | #POSAMP=3.0

```

对于只存在编译指示的结构文件, 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=[symbol1]%NUM=[n1], [symbol2]%NUM=[n2] 相当于一个简化的 POSITIONS\_FRAC 原子位置数据区块.

至此, 本文的内容已相当繁杂了, 但仍然有大量没有被讨论编译指示. 他们被列在附录章节 5.1 中.

最后再说明一点, 书写 AIRSS 的 .cell 文件时, 您可以根据个人喜好采取字母的大写或大写, 但是为了尽量避免不必要的运算和转换, 仍然推荐您使用大写字母.<sup>7</sup>

### 2.3.2 .param 文件的结构

\*.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 ! 甚至是单词 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.
- (5) 未出现在 \*.param 文件中的 Keywords 都将被设为默认值.

<sup>7</sup>详细原因见源码 “airss-0.9/src/buildcell/src/cell.f90”3732 行, “function up(string)”

(6) 您可以自主设置所设数值的单位 (具体细节稍后提及). 如果没有设置相应参数的单位, 则该参数将保持其默认的单位.

\*.param文件每一行的基本格式均为:

[keywords] : [value]	1
----------------------	---

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.

下述列表是\*.param文件中全部可用的Keywords.  
总计 255 个关键字.

COMMENT	VERBOSITY	1
IPRINT	CONTINUATION	2
REUSE	CHECKPOINT	3
TASK	CALCULATE_STRESS	4
CALCULATE_DENSDIFF	CALCULATE_ELF	5
CALCULATE_HIRSHFELD	RUN_TIME	6
BACKUP_INTERVAL	NUM_BACKUP_ITER	7
PRINT_CLOCK	PRINT_MEMORY_USAGE	8
WRITE_NONE	WRITE_FORMATTED_POTENTIAL	9
WRITE_FORMATTED_DENSITY	WRITE_FORMATTED_ELF	10
WRITE_ORBITALS	WRITE_CIF_STRUCTURE	11
WRITE_CELL_STRUCTURE	WRITE_BIB	12
WRITE_OTFG	WRITE_CST_ESP	13
WRITE_BANDS	WRITE_GEOM	14
WRITE_MD	WRITE_CHECKPOINT	15
CALC_MOLECULAR_DIPOLE	CML_OUTPUT	16
CML_FILENAME	LENGTH_UNIT	17
MASS_UNIT	TIME_UNIT	18
CHARGE_UNIT	SPIN_UNIT	19
ENERGY_UNIT	FORCE_UNIT	20
VELOCITY_UNIT	PRESSURE_UNIT	21
INV_LENGTH_UNIT	FREQUENCY_UNIT	22
FORCE_CONSTANT_UNIT	VOLUME_UNIT	23
IR_INTENSITY_UNIT	DIPOLE_UNIT	24
EFIELD_UNIT	ENTROPY_UNIT	25
PAGE_WVFNs	RAND_SEED	26
DATA_DISTRIBUTION	OPT_STRATEGY	27
OPT_STRATEGY_BIAS	NUM_FARMS	28
NUM_PROC_IN_SMP	NUM_PROC_IN_SMP_FINE	29
MESSAGE_SIZE	STOP	30
XC_FUNCTIONAL	XC_DEFINITION	31
XC_VXC_DERIV_EPSILON	RELATIVISTIC_TREATMENT	32
SEDC_APPLY	SEDC_SCHEME	33
SEDC_SR_TS	SEDC_D_TS	34
SEDC_S6_G06	SEDC_D_G06	35
SEDC_LAMBDA_OBS	SEDC_N_OBS	36
SEDC_SR_JCHS	SEDC_S6_JCHS	37
SEDC_D_JCHS	PAGE_EX_POT	38
NLXC_PAGE_EX_POT	PPD_INTEGRAL	39
NLXC_PPD_INTEGRAL	PPD_SIZE_X	40
NLXC_PPD_SIZE_X	PPD_SIZE_Y	41
NLXC_PPD_SIZE_Y	PPD_SIZE_Z	42
NLXC_PPD_SIZE_Z	IMPOSE_TRS	43
NLXC_IMPOSE_TRS	EXCHANGE_REFLECT_KPTS	44
NLXC_EXCHANGE_REFLECT_KPTS	K_SCRN_DEN_FUNCTION	45
NLXC_K_SCRN_DEN_FUNCTION	K_SCRN_AVERAGING_SCHEME	46
NLXC_K_SCRN_AVERAGING_SCHEME	RE_EST_K_SCRN	47
NLXC_RE_EST_K_SCRN	NLXC_EXCHANGE_SCREENING	48
NLXC_EXCHANGE_FRACTION	CALC_FULL_EX_POT	49
NLXC_CALC_FULL_EX_POT	NLXC_DIV_CORR_ON	50

NLXC_DIV_CORR_S_WIDTH	NLXC_DIV_CORR_TOL	51
NLXC_DIV_CORR_NPTS_STEP	PSPOT_NONLOCAL_TYPE	52
PSPOT_BETA_PHI_TYPE	SPIN_ORBIT_COUPLING	53
BASIS_PRECISION	FIXED_NPW	54
FINITE_BASIS_CORR	NELECTRONS	55
CHARGE	SPIN	56
NUP	NDOWN	57
SPIN_POLARIZED	SPIN_POLARISED	58
NBANDS	SPIN_TREATMENT	59
ELECTRONIC_MINIMIZER	ELEC_METHOD	60
METALS_METHOD	ELEC_ENERGY_TOL	61
ELEC_EIGENVALUE_TOL	ELEC_FORCE_TOL	62
FIX_OCCUPANCY	DIPOLE_CORRECTION	63
DIPOLE_DIR	ELEC_DUMP_FILE	64
NUM_DUMP_CYCLES	ELEC_RESTORE_FILE	65
MIXING_SCHEME	POPN_CALCULATE	66
POPN_BOND_CUTOFF	POPN_WRITE	67
PDOS_CALCULATE_WEIGHTS	BS_MAX_ITER	68
BS_NBANDS	BS_EIGENVALUE_TOL	69
BS_XC_FUNCTIONAL	BS_XC_DEFINITION	70
BS_WRITE_EIGENVALUES	GEOM_METHOD	71
GEOM_MAX_ITER	GEOM_ENERGY_TOL	72
GEOM_FORCE_TOL	GEOM_DISP_TOL	73
GEOM_STRESS_TOL	GEOM_MODULUS_EST	74
GEOM_FREQUENCY_EST	GEOM_LBFGS_MAX_UPDATES	75
GEOM_TPSD_ITERCHANGE	MD_NUM_ITER	76
MD_DELTA_T	MD_ENSEMBLE	77
MD_USE_PATHINT	MD_NUM_BEADS	78
MD_PATHINT_INIT	MD_PATHINT_STAGING	79
MD_PATHINT_NUM_STAGES	MD_TEMPERATURE	80
MD_THERMOSTAT	MD_BAROSTAT	81
MD_CELL_T	MD_LANGEVIN_T	82
MD_EXTRAP	MD_EXTRAP_FIT	83
MD_XLBOMB	MD_DAMPING_SCHEME	84
MD_OPT_DAMPED_DELTA_T	MD_ELEC_FORCE_TOL	85
MD_SAMPLE_ITER	MD_EQM_METHOD	86
MD_EQM_ION_T	MD_EQM_CELL_T	87
MD_EQM_T	MD_USE_PLUMED	88
MD_HUG_METHOD	MD_HUG_DIR	89
MD_HUG_T	MD_HUG_COMPRESSION	90
OPTICS_XC_FUNCTIONAL	OPTICS_XC_DEFINITION	91
TSSEARCH_METHOD	TSSEARCH_LSTQST_PROTOCOL	92
TSSEARCH_FORCE_TOL	TSSEARCH_DISP_TOL	93
TSSEARCH_ENERGY_TOL	PHONON_CONST_BASIS	94
PHONON_ENERGY_TOL	PHONON_PRECONDITIONER	95
PHONON_USE_KPOINT_SYMMETRY	PHONON_CALCULATE_DOS	96
PHONON_DOS_SPACING	PHONON_DOS_LIMIT	97
PHONON_FINITE_DISP	PHONON_FORCE_CONSTANT_CUTOFF	98
PHONON_FINE_METHOD	PHONON_METHOD	99
SECONDD_METHOD	PHONON_SUM_RULE	100
CALCULATE_BORN_CHARGES	BORN_CHARGE_SUM_RULE	101
CALCULATE_RAMAN	RAMAN_RANGE_LOW	102
RAMAN_RANGE_HIGH	PHONON_WRITE_FORCE_CONSTANTS	103
PHONON_WRITE_DYNAMICAL	EFIELD_ENERGY_TOL	104
THERMO_T_START	THERMO_T_STOP	105
THERMO_T_SPACING	WANNIER_SPREAD_TOL	106
WANNIER_SD_STEP	WANNIER_SPREAD_TYPE	107
WANNIER_MIN_ALGOR	WANNIER_ION_RMAX	108
WANNIER_ION_CUT_FRACTION	WANNIER_RESTART	109
WANNIER_ION_CUT_TOL	MAGRES_TASK	110
MAGRES_METHOD	MAGRES_CONV_TOL	111
MAGRES_XC_FUNCTIONAL	MAGRES_XC_DEFINITION	112

MAGRES_JCOUPLING_TASK	ELNES_XC_FUNCTIONAL	113
ELNES_XC_DEFINITION	ELNES_EIGENVALUE_TOL	114
SPECTRAL_THEORY	SPECTRAL_TASK	115
SPECTRAL_MAX_ITER	SPECTRAL_NBANDS	116
SPECTRAL_EIGENVALUE_TOL	SPECTRAL_XC_FUNCTIONAL	117
SPECTRAL_XC_DEFINITION	SPECTRAL_WRITE_EIGENVALUES	118
TDDFT_NUM_STATES	TDDFT_SELECTED_STATE	119
TDDFT_EIGENVALUE_TOL	TDDFT_XC_FUNCTIONAL	120
TDDFT_XC_DEFINITION	TDDFT_METHOD	121
TDDFT_EIGENVALUE_METHOD	TDDFT_APPROXIMATION	122
TDDFT_POSITION_METHOD	GA_POP_SIZE	123
GA_MAX_GENS	GA_MUTATE_RATE	124
GA_MUTATE_AMP	GA_FIXED_N	125
GA_BULK_SLICE	CUT_OFF_ENERGY	126
GRID_SCALE	FINE_GRID_SCALE	127
MAX_SCF_CYCLES	MIX_CHARGE_AMP	128
MIX_CHARGE_GMAX	MIX_HISTORY_LENGTH	129

### 2.3.3 .pp 文件的结构

.pp文件是 AIRSS 使用自带的 **pp3 对势 (pair potential)** 计算模块进行结构优化<sup>8</sup>时, 需要的参数配置文件. .pp中存储了相应的对势参数. .pp文件的结构十分简单, 如下所示.

```

1 1 12 6 2.5
2 A1
3 # Epsilon
4 1
5 # Sigma
6 2
7 # Beta
8 1

```

但是由于知识水平有限, 各个参数的具体意义仍不明确. 仍需进一步学习和源码解析.

不过由源码<sup>9</sup>已经初步得知, 第一行数据的第一个是原子种类数, 中间两个是对势中排斥势和吸引势的幂次.

## 3 计算指令

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) 2005-2017

```

<sup>8</sup>AIRSS 默认使用 CASTEP 完成此步骤, 需手动设置, 以使用 pp3 模块计算

<sup>9</sup>“.../airss0.9/src/pp3/src/pp.f90” 第 99 行

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] [-pp3] [-gulp] [-lammops]
[-gap] [-psi4] [-cluster] [-dos] [-workdir] [-max] [-num]
[-amp] [-mode] [-minmode] [-sim] [-symm] [-mpinp] [-steps]
[-best] [-track] [-keep] [-seed]
  -pressure f  Pressure (0.0)
  -build      Build structures only (false)
  -pp3        Use pair potentials rather than Castep (false)
  -gulp       Use gulp rather than Castep (false)
  -lammops    Use LAMMPS rather than Castep (false)
  -gap        Use GAP through QUIP/QUIPPY/ASE (false)
  -psi4       Use psi4 (false)
  -cluster    Use cluster settings for symmetry finder (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 vibrational
               modes (false)
  -minmode n   Lowest mode (4)
  -sim        f  Threshold for structure similarity (0.0)
  -symm       f  Symmetrise on-the-fly (0.0)
  -mpinp      n  Number of cores per mpi Castep (0)
  -steps      n  Max number of geometry optimisation steps (400)
  -best       Only keep the best structures for each
               composition (false)
  -track      Keep the track of good structures during relax
               and shake (false)
  -keep       Keep intermediate files (false)
  -seed       s  Seedname ('NONE')
user@machine_name$
```

airss.pl是使用 AIRSS 执行结构搜寻的主要指令. 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.

举一个简单的例子, 使用 AIRSS 的pp3模块作为能量计算软件搜寻 Al 的结构.

```
user@machine_name$ ls
Al.cell Al.pp
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$
```

AIRSS 的计算结果全部储存在了.res文件中.

```
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
/airss-0.9/examples/1.1
REM
REM
REM
REM
CELL 1.54180      2.20339      5.24398      5.24398      86.60117      89.99994
      90.00000
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
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$

```

第一行TITL中的第一个元素是软件分配给该结构的名称标签,  
 第二个元素是压力值 (GPa),  
 第三个元素是单胞的总体积,  
 第五个元素是单胞总的焓,  
 第六个元素是 spin(自旋?),  
 第七个元素是 modspin(??),  
 第八个元素是空间群名称.  
 最后一个元素是n-<#copies>(??)

## 4 计算数据处理

有了计算数据后, 就需要使用ca指令进行数据处理了. 关于此指令的使用, 软件中已经给出了十分详细全面的说明, 这里不再赘述, 只给出查询说明的方法.

首先执行空的ca指令. 因为按照airss.pl的经验, 猜测作者可能会将说明以这种形式贴出.

```

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

```

很遗憾, 关于ca的说明只有短短的一行, 但由这一行我们得知, ca是调用了cryan这个程序, 于是在终端输入空的cryan查询结果.

```

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

```

```

-f, --formula <formula>      each composition
                               Select structures of a
                               given composition
-t, --top [num]               Output top few results
                               (default 10)
-u, --unite <thresh>         Unite similar structures
-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)
...
user@machine_name$

```

由于说明文字过多, 之摘取了其中一小部分.  
如, 对之前结果进行分析.

```

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.

```

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$

```

关于 -u 后所跟数字的意义, 现仍不明确. 但可以判断这是某种与晶体点阵结构相似度有关的参数.<sup>10</sup>

## 5 附录

### 5.1 未说明的编译指示

```

1 ACONS
2 ADJGEN

```

<sup>10</sup>详细见源码 “.../airss-0.9/src/cryan/src/cryan.f90”2118-2122 行

3	AUTOSLACK
4	BREAKAMP
5	CELLADAPT
6	CELLAMP
7	CELLCON
8	CONS
9	COORD
10	CYLINDER
11	FLIP
12	FOCUS
13	MAXBANGLE
14	MAXTIME
15	MINBANGLE
16	MOLECULES
17	NOCOMPACT
18	NOPUSH
19	OCTET
20	PERMFRAC
21	PERMUTE
22	RASH
23	RASH_ANGAMP
24	RASH_POSAMP
25	REMOVE
26	SGRANK
27	SHIFT
28	SPHERE
29	SPIN
30	SURFACE
31	SYMM
32	SYMMNO
33	SYMMORPHIC
34	SYSTEM
35	THREE
36	TIGHT
37	VACANCIES
38	VACUUM
39	WIDTH