AIRSS Manual

Yang Li
lyang.1915@gmail.com

2020.10.24 — 2020.10.28

Contents

1 About AIRSS

AIRSS(**A**b **I**nitio **R**andom **S**tructure **S**earching) is a first-principles structure searching software developed by professor Chris Pickard at Cambridge University. It is an open-source software protected by the **GPL2 license**. You can visit its official website, `https://www.mtg.msm.cam.ac.uk/Codes/AIRSS`, to get the source package.

The so-called "structure searching" refers to such a process: for a system with unknown atomic structure but under certain physical conditions (such as atomic distance, distribution density, element composition, element ratio, etc.), we widely guessed its configuration, relax them using some DFT software, calculating its energy, and finally obtaining the most stable atomic structure. Obviously, the manual guessing or blind traversal search is extremely awkward, time-consuming, and even difficult to achieve. Therefore, it is necessary to use a set of mature structure search software, systematically and cleverly capture the stable configuration of the system.

AIRSS is exactly such kind of software! Professor Chris's introduction about this software can refer to the relevant video. There are also another two more commonly used structure searching software: USPEX and CALYPSO.

Unlike the genetic algorithm used in USPEX or CALYPSO, AIRSS is based on a completely random structure searching strategy, and the different structures it produces are completely random and independent. Such an algorithm is very conducive to the parallel implementation of search tasks.[1] The effectiveness of this random method is also discussed in Professor Chris' articles on AIRSS.[2] The most powerful feature of this random method of AIRSS lies in its many flexible adjustable parameters, which can be said to be the most outstanding feature of this project. The user experience of AIRSS gives a feeling of "point-and-shoot camera vs professional SLR camera". The former is simple and practical, you can submit various tasks with one click and get some results not bad. The latter has many buttons, adjustable parameters, and can be customized to produce a variety of more professional and refined results. The former sacrifices flexibility for ease of operation; the latter is the opposite. AIRSS can highly customize the characteristics of research objects according to user needs. For example, we can even use this software to verify the stability of an atomic structure, or to calculate the potential energy of the lattice surface.

Regrettably, as a powerful structure searching software, AIRSS's official manual is still to be developed. In addition, the relevant documentation or tutorials on the Internet are also very scarce, so most people are not familiar with such an extremely excellent program in the material design.

In order to make up for this shortcoming, the author decided to write this article. The following content is not a guide or tutorial, but only some records of learning AIRSS. Most of the conclusions of this article are derived from analyzing the source code, actually running the program, and trying to fumble. [3] **Limited by various factors, the errors in understanding and interpretation may be inevitable. If you find any questions or errors, please feel free to correct me.**

[1] For example, we can use AIRSS to randomly generate 1000 structures, and then relax them independently

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

[3] Although the official does not give a complete user manual, it does provide a large number of usage examples, located in the example directory of the program, for users to learn from.

2 Prepare Work

2.1 Linux Operating System

Before reading this record, you need to have a certain understanding of the Linux operating system. For example, to understand the meaning of the following commands:

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

And catastrophic accidents that can be caused by the following instructions:

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

Another thing that needs special reminder is that certain characters will be converted into symbols that are not recognized by the command line when the PDF document is compiled (such as the 'minus' in ls -1, although they look extremely the same as the 'minus' we typed in the command line). **Please be careful when directly copying commands or characters in this document.**

2.2 Program Installation and Remove

The latest `airss-0.9.1` has made a lot of simplifications in the installation process, you can read the included README file to install this version.

The main running script of AIRSS is written in perl language, and can only be installed in the `*nix` system, used in the command line (Command Line). Before installing this software, you'd better understand the usage of GNU make.

2.3 Run AIRSS

For you first time using AIRSS, you can enter the command '`airss.pl`' in terminal to view the software welcome interface.

```
user@machine_name$ airss.pl

      .o.       ooooo oooooooooo.    .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] [-mpinp]
                [-steps] [-best] [-track] [-keep] [-seed]
  -pressure f  Pressure (0.0)
  -build       Build str. 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)
  -ps4         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 str. (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)
  -mpinp    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` is the main command to perform structure search using AIRSS. The usage of this command has been systematically and briefly explained in the welcome interface. The table in the usage explanation has three columns. The first column is the name of the incoming parameter; the second is The data type of the incoming parameter, '`f`' represents a floating point number, '`n`' represents an integer, '`s`' represents a string, and "empty" represents a logical string `true` or `false`. The third column is a brief description of the corresponding parameter.

3 Custom Random Structure

The core component of AIRSS is named **buildcell**. The function of this component is to generate a series of initial atomic configurations with random structures but meeting the constraints of given physical conditions based on the *.cell file given by the user. In addition, we can also take out this module separately to adapt it to other programs (such as VASP). Learning to write highly customized *.cell files is the foundation of learning AIRSS.

In order to facilitate the subsequent description, we first briefly introduce the basic logic of the buildcell component.

The execution of the buildcell block is roughly divided into the following steps:

1. Read the configuration information in the *.cell file through the cell.f90 module.
2. Generate a structure that meets the requirements through build.f90, opt.f90 etc.
   (a) First determine the reasonable lattice constant according to the given atom and its radius, if there is a lattice tag[4] #FIX, ignore this step.
   (b) Under the premise of meeting the requirements, generated an atom at a random position.
   (c) Choose according to constraints: accept this position, reject this position, or **PUSH** the current atom. **PUSH** means: move two atoms that are too close to each other in the direction of their connection, and move them away Same distance. When there is FIX or NOMOVE in the atom label, ignore this operation.
   (d) Traverse all the atoms in the system until the complete structure is updated.
   (e) Depending on the user's setting, use the phenomenological pair potential (pp3) to simply relax the structure.
3. Output the above random structure that meets the requirements through the buildcell.f90 module.

3.1 File Structure of *.cell

The *.cell file is the "seed file" of the structure search. You can set search constraints in this file. The asterisk (*) before the file name represents a wildcard under the Linux system, that is, it represents any strings. In the AIRSS file system, we refer to the part before the suffix as the "seed name". Seed name can be used to distinguish the same kind of files of different calculations. All wildcards will be seen later in the text. Unless otherwise specified, they all refer to the seed name.

***.cell file has the following characteristics:**

1. Since AIRSS was written by some people who participated in the development of CASTEP, the program is extremely friendly to CASTEP, and the *.cell file here is fully compatible with the *.cell file in CASTEP.
2. When setting parameters in the *.cell file, you need to use certain keywords to indicate the meaning of the set parameters.
3. AIRSS completely reuses the keywords of CASTEP for structural declaration, such as LATTICE_CART, etc.
4. The built-in keywords of AIRSS usually start with #, such as #RASH. (To make the *.cell file of AIRSS fully compatible with CASTEP)
5. *.cell file is mainly composed of two parts: "structure data" and "global parameters". Correspondingly, keywords in the file can also be divided into the above two categories.
6. There is no restriction on the order of writing between any keywords.
7. All keywords set in *.cell need to be written in **capital letters**, and be careful **not to add spaces** between keywords, equal signs and parameter values. At the same time, any blank lines will be automatically ignored.
8. This file uses double pound signs (##) to indicate the content of the comments.
9. Although the program allows multiple keywords to appear in a single line, it is best to write only one keyword per line for the purpose of standardization and readability of the written file.

Next, we will introduce the specific writing method of *.cell file. Whether you can use AIRSS independently and efficiently depends on whether you can master the content of the following pages. Compared with those "about AIRSS", "no one will read it carefully" 'Foreword chapter, here can be said to be the core position of the article, so I want to use this "golden rank" to discuss some issues of the rationality of the AIRSS program design, which is quite beneficial for the better use of the software.

The problem that the structure search wants to solve is actually quite well-defined. We imagine a N-dimensional phase space, where different dimensions of the space represent different variable parameters of the structure (such as atomic ratio, lattice basis vector, atomic position, etc.)), each point in the space represents a different atomic configuration, and each configuration corresponds to an energy value. So there will be a N+1-dimensional energy surface in our system (such as our phase space If it is two-dimensional, then there will be a three-dimensional surface about energy). What the structure search does is to find some/all energy minimum points of this energy surface, and then determine the structure corresponding to the global minimum energy (That is what we call the most stable configuration).

One way to solve this problem is to use some very clever algorithms (such as genetic algorithms or machine learning) to make the system "evolve" based on different metastable configurations, and find the most stable configuration globally. The implementation steps of the algorithm are as follows: Initially generate a series of random mechanism types; then use first-principle calculation software to relax

these configurations, inherit/learn the characteristics of low-energy structures from the relaxation results; then use the information learned from the previous generation Generate a next-generation structure; iterate in this way, and finally expect to find a global most stable state. The advantage of this algorithm is that it can quickly locate the energy minimum point in a large phase space, which does not rely on any prior information , It is only a purely theoretical prediction of a certain structure is very effective. But at the same time, due to the causal relationship between the previous generation and the next generation structure, the structure of different generations can only be calculated in sequence (serial). In addition, use This type of algorithm will also face a big challenge, that is, the so-called "phase space collapse" (that is, due to the limitations of the initial selection with the organization type, the subsequent new structures are all localized in certain sub-regions. Near steady state). In order to solve this problem, we have to introduce an adjustable parameter: "mutation probability" (that is, it is believed that the structure inherits the parent structure, and there is a certain probability that it will change in certain positions). But in some Under circumstances (such as setting the mutation probability too high or too low), this parameter is not as effective as we thought.

AIRSS uses another structure searching strategy. Its *.cell file provides a large number of adjustable parameters, and each parameter corresponds to a number of constraints, which can achieve **precise restrictions on the phase space**. The system is meeting these The initial configuration is **randomly** set in the phase space of the constraints, and then a (meta)stable state near the initial configuration is found through the relaxation of the first-principles calculation software, which is regarded as a "structure search". Through large-scale (such as thousands or tens of thousands of times) "search", combined with the **structural duplicate check algorithm**, we can finally find a global most stable state in the constrained phase space. Such a steady state probably has two characteristics: **the energy is lower (the lowest)**, and **the number of repetitions is more (the most)**.

The strategy of "random selection" of the initial structure at first sounds unreasonable and extremely inefficient, but if the phase space obtained after "precision constraints" is not very large, then even if it is not used, it is very complicated The genetic algorithm, which relies solely on simple random dots, can also obtain good results. At the same time, the random setting of the initial structure ensures the **efficiency of parallel computing** while avoiding the occurrence of "phase space collapse".

Perhaps seeing this, you will argue that if we set the first generation in the genetic algorithm as many as in the AIRSS, such as 10,000, the final result is not much better than the AIRSS algorithm, after all, AIRSS only It is equivalent to the first step of the genetic algorithm? But the fact is that in a small phase space, we use 10,000 configurations per generation to iterate for 100 generations, and directly use 1,000,000 random structure relaxation calculations. The difference is actually Not big, and sometimes the latter is more accurate. And just saying "10,000 configuration iterations for 100 generations per generation" would feel ridiculous, because this kind of calculation requires massive computing power. Generally speaking. In other words, we will take the number of configurations in each generation on the order of 100. So I prefer to think that these two algorithms are complementary, and one is suitable for serially finding some low-energy extreme points in a super large phase space. , The other is suitable for finding all extreme points in a small phase space that is tightly restricted in parallel. And according to my experience, the performance of AIRSS in a larger phase space is not as bad as we thought Be-

cause for the phase space that actually meets the physical constraints, there are a small number of extreme points (otherwise the structure may have a large number of metastable states in the experiment), and it can completely rely on the relaxation process calculated by first principles. Find these final extreme points from an initial point that looks random but is actually within the range of DFT relaxation ability.

Using a strategy search structure that **accurately limits the size of the phase space** plus **random search** can be said to be a very strange but effective idea. Because most of the time, we do not need to predict a structure completely theoretically, but cooperate with experiments (such as Experiments have already known the element ratio and crystal shape, and sometimes even the approximate atomic position can be known through electron microscopy), and jointly study the atomic structure of a certain material. At this time, the variable phase space of atoms may not be very large, so "How to define the phase space under study more precisely" has become a more important issue than "how to optimize the algorithm to get the minimum point faster". Of course, the size of the phase space is also relative. For example, if we choose 10 random organization types, it is obviously not possible to cover any so-called "small phase space".Only when this number increases to, for example, 10,000, can we reluctantly say that for some systems. But here comes another benefit of randomly generating structures, that is, it is inherently suitable for parallel computing. We know that almost all the time in structure search is spent on structure relaxation. If we have 8000 cores in our hands, use 8 Each core optimizes a configuration, then at the same time we can search for 1000 structures at the same time. If there are 80,000 cores (of course, this is almost impossible, face smile.jpg), then 10,000 structures can be calculated at the same time This simple and rude and efficient parallel method is unmatched by other algorithms. AIRSS also combines the two characteristics of **precise limit phase space size** and **random structure generation**. In many cases (especially when working with experiments to search for atomic structures), It is very efficient.

At present, the functions of the parameters available in the *.cell file can be divided into the following categories:

1. **Declare the initial lattice structure.** For example, declare the initial unit cell basis vector, initial atomic position, etc.
2. **Constraints on unit cell parameters.** Such as fixing the initial lattice base vector, constraining the unit cell system, constraining the degree of change of the unit cell, constraining the crystal system of the unit cell, adding a vacuum layer to the unit cell, whether to expand the cell, etc.
3. **Requirements for the chemical formula of the system.** For example, how many yuan the system is, whether the electrons can be balanced (whether there are dangling bonds), etc.
4. **Restrictions on the position of atoms.** Such as the shortest distance between two atoms, the maximum/minimum distance of an atom from the initial position, the coordination number of an atom, etc.
5. **Restrictions on the symmetry of the system.** For example, the system has several symmetry operations, in what space group, the degree of fineness of symmetry search, whether to introduce symmetry breaking disturbances, etc.
6. **Adjustment to the search algorithm of the program itself.** For example, accept the upper limit of the number of search failures, whether to introduce PSUH, whether to introduce TPSD, whether to introduce RASH, etc.
7. (Less commonly used) Add a force field covering the entire unit cell in the crystal. For example, add a spherical force field, an ellipsoidal force field, a strip force field, a plane force field, etc.

The following is the `Al.cell` file used to predict the structure of metal aluminum:

```
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$
```

The structure of the above *.cell file can be summarized as follows:

1. The first 9 lines are data reading blocks defined by the format %BLOCK [keywords], this format is very common in CASTEP. These blocks define the basic **structure data** of the crystal.
2. The [keywords] of the 1 to 5 rows of data is LATTICE_CART, which declares the unit cell base vector defined by the Cartesian absolute coordinate system.
3. The [keywords] used in lines 7 to 9 is POSITIONS_FRAC. The meaning of this in CASTEP is "the position of the atom defined in fractional coordinates". In AIRSS, the data module can not only specify the initial position of the atom, but also Define the constraint conditions for a single atom in the search process. If the specific position of the atom is not known initially, it can be set to any value, such as (0, 0, 0), and the program will find the atomic position that meets the requirements. Atom position More specific details of the settings will be given later.
4. #MINSEP on line 11 is **global parameter** in AIRSS. #MINSEP=1.5 indicates that the distance between any two atoms must not be less than 1.5 $\mathring{(}$A).

As we can see, the structure of the *.cell file is roughly divided into two parts: **structure data** and **global parameters**.

3.2 Parameter Details of *.cell File

3.2.1 structure data

This part follows the data pattern defined in the CASTEP structure file. The structure data consists of two parts: the lattice parameter data block and the atomic position data block.

The specific mode of the data block is as follows:

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

The data block keywords that can be used in AIRSS are listed in **Table 1**.

Table 1: AIRSS Cheat Sheet – Data BOLCK

Parameters	Description
LATTICE_CART	Use the Cartesian coordinate system to define the base vector of **unit cell** with a matrix of 3×3. By appending the following tags to the block: #FIX,#CFIX, #ABFIX, can be implemented separately: fix the entire lattice shape, fix the lattice constant c, and fix the lattice constant ab. In addition, it should be emphasized that if you want to ensure that the lattice parameters are unchanged, except in the lattice Add #FIX to the parameter data block to ensure that it is maintained as a constant value when guessing the lattice structure, and it is also necessary to make corresponding settings in the DFT relaxation software to ensure that it is also in the structure relaxation. constant.
LATTICE_ABC	Use a 2×3 matrix to define the lattice parameters of the unit cell. The first row of the matrix is the parameter abc, and the second row is the three angles. In the same .cell, it is the same as |LATTICE_CART|Chooseone of the two. By adding the following tags to the block: #FIX,#CFIX,#ABFIX, can be achieved separately: fix the entire lattice shape, Fixed lattice constant c, fixed lattice constant ab.
POSITIONS_FRAC	The atomic position is defined by the fractional coordinates in the lattice base vector coordinate system.
POSITIONS_ABS	Define the atomic position with absolute numerical coordinates in Cartesian coordinate system. Choose one from this or POSITIONS_FRAC.
SYMMETRY_OPS	Define the symmetry operations that the generated unit cell must meet, a group of four, the first three lines define rotation operations, and the fourth line defines translation operations. For specific usage, please refer to CASTEP:SYMMETRY OPS.

The following briefly introduces the detailed writing method of **lattice param-**

eter data block and **atomic position** data block.

Lattice parameter First introduce the setting of lattice parameters, take `LATTICE_CART` as an example.
Example 1.

```
%BLOCK LATTICE_CART
20 0 0
0 20 0
0 0 20
#FIX
%BLOCKEND LATTICE_CART
```

The above fields construct a cube of $20 \times 20 \times 20$ $(\mathring{A})^3$ as the unit cell of the crystal. Here `#FIX` is called **lattice tag**.

It states that the lattice constant cannot be changed during the search (generating the initial random structure). If you want to ensure that the lattice constant of the system remains unchanged during DFT relaxation, you need to set it in the input file of the corresponding calculation software .

Atomic position Then introduce the setting method of atomic position, take `POSITIONS_FRAC` as an example.
Example 2.

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

As can be seen from this example, the basic format of atomic position structure data is:

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

The meaning of the elements inside each line is as follows:

1. The first item is the element name. If **the element name is set to Z**, it means that the atom is a void. The void has the same volume and PUSH properties as a real atom, but it will not be output in the final result.
2. The terms two, three, and four are the coordinates of the atomic position.
3. The first item after the pound sign (#) is the name of the atom set where the atom is located. The name of the atom set can be set to any character. **Atoms with the same atom set name form an atom set, and an atom set consists of one or more A basic structural unit composed of two atoms. The atoms in the same atomic set move the same randomly and are not affected by PUSH**. If there is no special requirement, in order to achieve the relative maximum degree of freedom, it is generally not recommended to put different atoms into the same atom concentrated.
4. The content after the percent sign (%) is **atomic tag**. The same line of atoms can specify multiple atom tags, separated by spaces, they jointly specify several constraints that the atom should satisfy condition.

Table 2 is the specific description of all atom tags available in AIRSS.

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

Parameters	**Type**	**Defualt**	**Description**
`NUM=n` `NUM=`$n_{min} - n_{max}$	`int`	1	Define the number of atoms actually represented by the row of atoms. Repeated atoms defined with `NUM` will be individually assigned to different **atom sets** one by one.
`POSAMP=d`	`float`	-1.0	Define the maximum distance that the atom(s) can deviate from the initial position in the structure search process. If it is a negative value, there is no limit.
`MINAMP=d`	`float`	0.0	Define the minimum distance that the atom(s) deviates from the initial position during the structure search process.
`XAMP=d`	`float`	-1.0	定义该行原子在结构搜索过程中, 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

Continued on next page...

Table 2 – *Continued from previous page*

Parameters	Type	Defualt	Description
YAMP=d	float	-1.0	定义该行原子在结构搜索过程中，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	Define the maximum distance that the row of atoms can move in the X axis direction during the structure search. If it is a negative value, there is no limit. Setting this item will invalidate POSAMP and MINAMP
ANGAMP=θ	float [0, 360]	-1.0	The maximum value of the rotation angle of the atom around the center of **atom set** where it is located (independent of POSAMP, separate functions). When there is a lattice label #FIX, ANGAMP=0, POSAMP=0 is shared with NOMOVE(\FIX) to ensure that the atom does not move in the original position. If all atom sets in the cell file have only 1 atom, this parameter is invalid, ANGAMP Forced to 0. If it is negative, there is no limit.
RAD=d	float	0.0	Set the radius of the atom, used to judge the minimum distance between two atoms, MINSEP. Its priority is lower than directly setting the global parameter #MINSEP=.
FIX	void	off	This atom is not affected by PUSH when it generates a structure. Any movable atom close to this atom will be PUSH back with 2 times the PUSH step. At the same time, write instructions to the cell file to make it immobile when CASTEP relaxes. This command is only valid when the lattice is #FIX. FIX in the atom tag and #FIX, #CFIX, # in the lattice tag ABFIXSimilar names, but different functions.

Continued on next page...

Table 2 – *Continued from previous page*

Parameters	**Type**	**Defualt**	**Description**
NOMOVE	void	off	This atom is not affected by PUSH when it produces a structure. Any movable atom close to this atom will be PUSH back with 2 times the PUSH step. This instruction is only valid when the lattice is #FIX.
COORD=n, COORD=$n_{min} - n_{max}$	int	-1	Constrain the coordination number (number of nearest neighbors) of atoms in the row. If it is negative, there is no limit.
NN=$\pm$elm	string	null	Specify the type of the nearest neighbor element of the atom, '+' means that the element must be adjacent to the element, and '−' means that the element cannot be adjacent to the element. Is read once). If it is empty, it means unlimited.

Continued on next page...

Table 2 – *Continued from previous page*

Parameters	Type	Defualt	Description
OCC=p	float [0, 1]	1.0	The probability of the point occupying an atom. The sum of the probability of occupying the same element in different positions must be an integer greater than 0. Accept input in the form of fractions, such as OCC=1/3. If the value is set to less than 0 The number of the atom, the OCC of the atom is forcibly set to 1, and other atom positions that may be derived from the symmetry of the atom are no longer output. If the atom also has the FIX or NOMOVE label, then |OCC|is forcibly set to 1. If the atom OCC is greater than 1, set it to a space (the same effect as setting the element name to Z). When the system has symmetry, using a smaller OCC can force an atom Located at a point with higher symmetry. But a more natural usage is to use MULT instead of OCC to set. If the global parameter #NFORM=n(n is greater than 0) is set: In the absence of symmetry, the OCC parameter is invalid; in the case of symmetry, please refer to the description in MULT.

Continued on next page...

Table 2 – *Continued from previous page*

Parameters	Type	Defualt	Description
MULT=m	float	symm num	Set the multiplicity of the atomic point.[5] The multiplicity of the grid tells us that if we place an atom at this position, then the symmetry How many atoms will be generated. Using this parameter, you can limit the number of atoms in a row of atoms actually generated by the symmetry of the crystal, and the level of symmetry of the position. The default size of this parameter is the number of group operations, that is, the default is only Look for the general position. If the atom in the line has the label FIX or NOMOVE, the MULT will be invalid, and OCC will be forced to -1. If the global parameter #NFORM=n(n is greater than 0), it is equivalent to setting the value of MULT of all atoms to n, and not setting #NFORM=. Regarding lattice multiplicities and symmetry For more information, please refer to PDF: Space Group and Multiplicities.
PERM	void	off	After completing the selection of atomic positions, rearrange (exchange with each other with a certain probability) the types of elements between the specified atomic positions. It needs to be used with the global parameter #PERMUTE, otherwise the parameter will be turned off.
ADATOM	void	off	Indicates that the atom in the row is equal, and the atom that is added after the generation of all atom positions without this label.

Continued on next page...

[5]This value is positive, it will force the value of the OCC parameter to be overwritten to MULT/SYMM_NUM, where SYMM_NUM is the number of symmetry groups in the lattice. In AIRSS, multiplicities is achieved by setting the score OCC, so the same effect can be achieved by directly setting the score OCC without setting MULT.

Table 2 – *Continued from previous page*

Parameters	Type	Defualt	Description
ATHOLE	void	off	Indicates that the atom is located in the hole that was dug out, and the atom will be automatically deleted from the output structure. It is often used in conjunction with the global variables #HOLE=, #HOLEPOS=.

3.2.2 Global parameters

The conditions that the entire system should comply with in the structure search process are specified by global parameters. The global parameters in AIRSS all start with a pound sign (#), which affect the lattice and all atoms in the system. In addition, some global parameters and Atomic tags have the same effect.At this time, the priority of atomic tags setting is higher than the global parameters.

The functions and detailed usage methods of all global parameters available in AIRSS are show in **Table 3**.

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

Parameters	Type	Defualt	Description
#CELLCON=a b c α β γ	int	off	define the conditions that the unit cell should meet: a, b, c are the lattice constant; α, β, γ are the three crystal angles. The optional values for the lattice constant are $\{0, -1\}$, The optional values for crystal angle are $\{0, -1, \theta\}$. 0 means no constraint, -1 means equal. For example, #CELLCON= -1-1 -1 90 90 90 means cubic crystal system.
#SYSTEM=sys	string	off	Set the crystal system where the structure is located. The optional inputs are [Tric, Mono, Hexa, Rhom(/Trig), Orth, Tetr, Cubi]. This parameter is actually a number of special combinations of '#CELLCON=' parameters.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#CONS=p	float $(0, 1]$	0.4	Constrain the side length of the unit cell. Close to 0 means no constraint at all, and equal to 1 means do everything possible to constrain the unit cell abc to have the same length on the three sides.
#ACONS=p	float $(0, 1]$	0.5	Constrain the crystal angle of the unit cell to keep it away from a situation where the lattice parameter is extremely small and the unit cell is flat. Close to 0 means no restriction at all, and equal to 1 means doing everything possible to constrain the three bond angles to be strictly equal to 90 degrees.
#CELLAMP=p	float	-1.0	Within a certain limit, the lattice shape is changed randomly. When set to a negative value, the lattice shape will change completely randomly. The smaller the p, the smaller the deviation of the lattice from the given value in the cell file. To enable this parameter , It is recommended not to set a value exceeding 1.0.[6] After using this parameter, the grid label #ABFIX and #CFIX, atomic tags #CONS and #ACONS will all be deprecated. If it is a negative number, turn off this parameter setting.
#VACUUM=d	float	0.0	Add a vacuum layer of d(Å) in the lattice (Z direction).
#NFORM=n	int	-1	Declare that the actual number of atoms in the unit cell is %BLOCK POSITIONS_* or n times defined in #SPECIES. This option cannot be combined with #NATOM. If it is negative, this parameter is turned off.

Continued on next page...

[6] For details on how to use this parameter, please refer to the usage of cellamp variable on line 100 of airss-0.9.1/src/buildcell/src/build.f90.

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#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	Define the size of the super cell. You can use the number of unit cells in the super cell n, the value of the base vector of the super cell lattice in three directionsn_a n_b n_c, or the super cell The transformation matrix between the base vector of the cell and the unit cell $[[a_x, a_y, a_z], [b_x, b_y, b_z], [c_x, c_y, c_z]]$ defines a new super cell. Super cell The construction strategy is: first use the unit cell to construct the qualified atomic configuration, and then copy and paste the super cell according to the expansion cell parameters and directly output it.
#SLAB	void	off	Declare that the structure is a two-dimensional planar structure. When taking the super cell, the cell will no longer expand in the c direction, and the c direction will be ignored when checking the symmetry of the structure.
#SURFACE	void	off	Declare that the structure is a surface structure, the surface structure inherits all the characteristics of SLAB, and at the same time adds a symmetry restriction condition related to the surface. [7]
#MOLECULES	void	off	Stated that the atoms in the same atomic set constitute a basic molecular configuration, and the atoms in the same atomic set will no longer be checked when checking the coordination..
#CLUSTER	void	off	Stated that the predicted structure is acyclic cluster.

Continued on next page...

[7] See source code airss-0.9.1/src/buildcell/src/cell.90 row 3216.

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#FOCUS=n	int	0	The final structure of the constraint must be n components (consisting of n different elements). This parameter is generally used for variable component analysis. Less than or equal to 0 means no constraint.
#OCTET	void	off	Check whether the chemical ratio is electronically balanced (whether all electrons are divisible by 8).
#POSAMP=d	float	-1.0	Same as the definition in the atom tag, the maximum position of the atom from the initial point in the structure search. Negative values are unlimited.
#MINAMP=d	float	-1.0	Same as the definition in the atom tag, the minimum position of the atom from the initial point in the structure search. Negative values are unlimited.
#XAMP=d	float	-1.0	The same as the definition in the atom tag, the maximum amplitude of the atom X-direction deviation from the initial point in the structure search. Negative value is unlimited. Setting this option will invalidate POSAMP and MINAMP.
#YAMP=d	float	-1.0	Same as the definition in the atom tag, the maximum amplitude of the atom Y-direction deviating from the initial point in the structure search. Negative value is unlimited. Setting this item will invalidate POSAMP and MINAMP.
#ZAMP=d	float	-1.0	Same as the definition in the atom tag, the maximum amplitude of the atom Z-direction deviating from the initial point in the structure search. Negative value is unlimited. Setting this item will invalidate POSAMP and MINAMP.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#ANGAMP=θ	float $[0, 360]$	-1.0	Same as the definition in the atom tag, the maximum value of the rotation angle of the atom around the center of **atom set** where it is located. Negative values are unlimited.
#MINBANGLE=θ	float $(0, 360]$	0.0	Set the minimum value of the atom bond angle in the search structure.
#MAXBANGLE=θ	float $(0, 360]$	180.0	Set the maximum value of the atom bond angle in the search structure.
#MINSEP=d #MINSEP=d X-X=d_{XX} X-Y=d_{XY} ...	float	0.0	The minimum distance between two atoms can also be used to define the fixed distance between two atoms. For example, #MINSEP= 2.0 Li-Li=2.6 Ge-Ge=2.51
#RAD=d	float	0.0	Same as the definition in the atom label, define the radius of the atom.
#COORD=n	int	-1	Same as the definition in the atom label, set the coordination number limit of the atom. Negative value has no limit.
#SPIN=S_{real} S_{mod} #SPIN=S_{real} S_{mod} "elm_1 elm_2... "	float, string	off	Randomly set the value of the collinear spin on each atom of the system. And require $S_{t,real}/N_{ions} = S_{real}$, $S_{t,mod}/N_{ions} = S_{mod}$, where $S_{t,real}$ is the sum of the spins of all specified atoms, $S_(t, mod)$ is the sum of the absolute values of the spins of all specified atoms. The spins of unspecified atoms remain at 0. The element symbols with spaces enclosed in double quotes can be used to indicate which are specified atoms. If no atom is specified , All atoms are specified by default. For example, #SPIN=0 5 "Fe Co " or #SPIN=1 4.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#SPECIES=elm_1%$tags_1$, elm_2%$tags_2$, ... #SPECIES=elm_1, elm_2...	string	off	Use simplified notation to define the atomic components of the system. For example, #SPECIES=Si%NUM=1 COORD=2,O%NUM=2. This parameter cannot be combined with BLOCK POSITIIONS_* appears at the same time.
#NATOM=n #NATOM=$n_{min} - n_{max}$	0	int	Used in conjunction with #SPECIES to define the total number of atoms in a unit cell, which is generally used for variable component analysis (metamorphic prediction). And using this global parameter will make #SPECIES included in the command All atomic tags after % are invalid. [8]For detailed reasons, please refer to the difference between line 493 and line 514 of the source code "airss-0.9.1/src/buildcell/src/cell.f90".) If there are many elements in #SPECIES, the random number of each element is kept, and the total is kept as NATOM. This parameter is automatically invalid when it is set to 0.
#TARGVOL=V #TARGVOL=$V_{min} - V_{max}$	float	init. cell vol.	Fixed lattice volume to V, or a random value between $V_(min)$ and $V_(max)$. This parameter is invalid when there is a lattice label #FIX. The default is the volume of the initial given primitive cell. This parameter is often used when the BLOCK POSITION_* block is not introduced.
#VARVOL=V	float	init. cell vol.	It has the same effect as #TARGVOL=, this parameter will override the setting of #TARGVOL=. The default is the volume of the initial given primitive cell.

Continued on next page...

[8](

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#SLACK=p	float $[0, 1)$	0.0	Using this parameter can reduce the system's restrictions on the bonding between atoms (the relative position of atoms) as a whole, the larger the p, the lower the requirements for the distance and angle between atoms. The default is 0, and the recommended value is 0.1–0.3.
#AUTOSLACK=p	float $[0, 1)$	off	Using this parameter can reduce the system's restrictions on the bonding between atoms (the relative position of atoms) as a whole. The larger the p, the lower the requirements for the distance and angle between atoms. If the given initial value p is not appropriate, the SLACK is incremented in steps of 0.01 until the appropriate SLACK value is found.
#FLIP	void	off	When searching for the structure, a random flip operation is introduced to **atomic set**. If there is only one atom in all atom sets in the system, this parameter is invalid.
#REMOVE	void	off	(After PUSH,) Delete (one of) the overlapping atom. Can be used for structure prediction with high initial atom density.
#TIGHT	void	off	Make the generated structure more compact.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#SYMMOPS=n #SYMMOPS=~n #SYMMOPS=$n_{min}-n_{max}$	int	off	State that the generated structure must contain n symmetry operations. If the system is a periodic crystal structure, it is recommended to select from the following integers: 1,2,3,4,5,6,8,12,16,24,48. If the input contains a tilde (~), it means that only atoms will be placed on general positions when searching for the structure. For the more symmetrical but quantity [9]Less special positions are not considered. If you use MULT to specify multiplicity by atom, the tilde (~) is ignored, but if #NFORM=n(n is greater than 0) is used, it will conflict with the tilde (~). Also note that, The method to achieve lock symmetry in AIRSS is to copy n atoms according to the symmetry. For example, if the atom has C_4 symmetry, the total number of atoms output will be 4 times the set. But if you combine atoms The use of tags OCC, MULT can realize the operation of placing fewer atoms at higher symmetry points. It is not recommended to use #NFORM=n to limit the number of atomic symmetric copies under the mode containing tilde (~).

Continued on next page...

[9]here "quantity" Refers to the minimum number of atoms of the same kind required for the symmetry operation, that is, the multiplicities of the point.

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#SYMM=spg #SYMM=~spg	string	off	The generated structure must be in the spg space group, spg is the name of the space group. If the input contains a tilde (~), it means that the structure search will only be in general positions Place atoms on the upper side. Special positions with higher symmetry but fewer numbers are not considered. In this mode, #NFORM=n is not recommended to limit the number of symmetric copies of atoms.
#SYMMNO=n #SYMMNO=~n	int	off	The generated structure must be in the space group No. n, n is the number of the space group. If the input contains a tilde (~), it means that the structure search will only be in Atoms are placed on general positions, and special positions with higher symmetry but fewer numbers are not considered. In this mode, it is not recommended to use #NFORM=n to limit the number of symmetric copies of atoms.
#SYMMORPHIC	void	off	Only check the point symmetry operation.
#SGRANK=n	int	230	Set the upper limit of the sequence number for space group n. When this value is set to 230, symmetry locking of any space group is accepted.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#ADJGEN=n #ADJGEN=$n_{min} - n_{max}$	int	0	Adjust the number of general positions (GP) used in the unit cell. When this value is 0, the GP points will be used to the greatest extent. Increasing the value will gradually use more Special positions (SP) points. If you try After finding that it is difficult to generate a qualified structure, the program will dynamically increase this value. For the relationship between SP/GP and crystal space group, please refer to: Wiki: Wyckoff Position.
#BREAKAMP=d	float	0.0	Moving atoms randomly in the direction of the lattice vector a destroys the original symmetry, and the moving distance of the atomic fractional coordinates is calculated with: $d_a^{(frac)} = (random(0,1) \times d)^{1/3}$
#NOPUSH	void	off	For atoms that are too close, do not introduce a PUSH step, directly reject the configuration. This keyword will not close the PUSH step of the crystal center potential field introduced by keywords such as #SPHERE.
#PUSHSTEP=p	float	0.25	The scale parameter of each step of PUSH moving distance (stepsize).
#PUSHMAX=n	int	100	Set the maximum number of PUSH steps (in the output of buildcell, the number of * |: - symbols between two Xs).
#OVERLAP=cov	float	off	After PUSH is over, TPSD[10] is additionally used to perform simple relaxation of the potential pair crystal structure. cov is the convergence criterion. The smaller the value, the higher the limit on the lattice convergence. The recommended value is 0.1–0.2.

Continued on next page...

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

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#RASH	float	off	After using TPSD to relax the potential, introduce random displacement and rotation between the atom sets (SHAKE step). This parameter is only effective after setting #OVERLAP.
#RASH_POSAMP=d	float	1.0	Set the maximum distance of the SHAKE step introduced by RASH to move the atomic set, similar to POSAMP. This item must be a small positive value, **cannot be set to a negative number**.
#RASH_ANGAMP=θ	float $(0, 360]$	30.0	Set the maximum angle of rotation of the SHAKE step atom set introduced by RASH around its midpoint, similar to ANGAMP. This item must be a small positive value, **cannot be set to a negative number**.
#CELLADAPT	void	off	In the case of setting #OVERLAP, you can set this additionally to force the relaxation of the simple structure of TPSD and try to change the shape of the unit cell while keeping the volume unchanged. By default, the system will not optimize the TPSD structure. Change the shape of the unit cell. If there is a lattice mark #FIX or #CELLAMP=0 is set, this parameter is invalid.
#THREE=p	float	TODO	Use three-body potential energy instead of TPSD relaxation structure, **The function corresponding to this parameter has not been implemented in airss-0.9.1**.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#COMPACT	void	on	Perform a niggli reduce operation on the final unit cell. When the lattice shape is not locked (the lattice label #FIX, #CFIX, #ABFIX is not introduced), this item is turned on by default. R. W. Grosse-Kunstleve, N. K. Sauter and P. D. Adams, Acta Cryst., A60, 1-6 (2004)
#NOCOMPACT	void	off	Forcely close COMPACT operation.
#PERMUTE	void	off	After completing the selection of the atomic positions, rearrange (exchange with each other with a certain probability) the types of elements between the specified atomic positions. You can use it in conjunction with the atomic label PERM.
#PERMFRAC=p	float $[0, 1]$	1.0	Set the probability of rearrangement.
#HOLE=d	float	-1.0	Set the radius of the hole to be cut on the lattice. When set to a negative number, no processing will be done on the formed structure.
#HOLEPOS=f_a f_b f_c	float	random	Set the position (fractional coordinates) of the cut hole on the lattice. The default is a random position. Can be used with ATHOLE in the atomic tag.
#VACANCIES=n@elm	float, string	off	After waiting for the structure to be generated, select n atoms whose element type is elm and replace them with vacancies. [11]
#MAXTIME=t	float	1.0	Set the upper limit of the time used for the PUSH step of a guess structure, after which the program will stop PUSH and re-guess the new structure. Default $t = 1$ second.

Continued on next page...

[11] This parameter originally designed an input without the @ symbol, but it is currently in airss-0.9.1 , This input has some bugs in the subsequent processing, so it is not explained here.

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
`#NFAILS=n`	`int`	0	The number of failures allowed for each structure (the number of occurrences of `X` in the output of the `buildcell` command). If its value is 0, there is no limit.
`#SPHERE=r`	`float`	off	Introduce a spherical potential energy at the center of the unit cell. Set the radius of attraction of the spherical potential energy to `r`. When the distance between the atom and the center of the lattice is greater than `r`, it will be PUSH directed to the center of the lattice.
`#ELLIPSOID=r` ε	`float`	off	An ellipsoidal attractive potential is introduced at the center of the unit cell. `r` is the length of the semi-major axis of the ellipsoid potential energy, and ε is the degree of deformation. When the distance between the atom and the center of the lattice is greater than `r` , Will be subjected to PUSH pointing to the center of the lattice. $\varepsilon = 0$ is spherical, the larger the ε, the more serious the distortion.
`#PANCAKE=r` ε	`float`	off	Introduce a pie-shaped attraction potential at the center of the unit cell. `r` is the radius of the disc, ε is the degree of deformation. When the distance between the atom and the center of the lattice is greater than `r`, it will be affected PUSH pointing to the center of the lattice. $\varepsilon = 0$ is a flat round pie, $\varepsilon = 1$ is the potential energy close to the sphere.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	Type	Defualt	Description
#CIGAR=r ε	float	off	A cigar-like attractive potential is introduced at the center of the unit cell. r is the length of the cigar, ε is the degree of deformation. When the distance between the atom and the center of the lattice is greater than r, it will be directed to the crystal The PUSH in the center of the lattice. $\varepsilon = 0$ is the tip-like potential energy, and $\varepsilon = 1$ is the potential energy close to the spherical shape.
#CYLINDER=r	float	off	Introduce a cylindrical potential energy at the center of the unit cell (the atom is not subject to force in the Z direction). r is the radius of attraction of the cylindrical potential energy. When the distance between the atom and the center of the lattice is greater than r, it will receive PUSH in the XY direction pointing to the center of the lattice.
#CORE=r	float	off	Attach a repulsive core to the defined spherical (ellipsoid, round pie, cigar, cylindrical) attraction potential. Set the repulsive core radius (major axis length, radius, cigar length, cylinder radius) to r. When the distance between the atom and the center of the lattice is less than r, it will be PUSH away from the center of the lattice.
#WIDTH=l	float	off	Use planar potential energy (atoms are not subject to force in the X and Y directions), and additionally calculate the distance of the PUSH step. l is the length of the planar potential energy attraction. When the distance between the atom and the origin is greater than l When, it will be PUSH in the Z direction pointing to the origin.

Continued on next page...

Table 3 – *Continued from previous page*

Parameters	**Type**	**Defualt**	**Description**
`#SHIFT=h`	`float`	0.0	Move the plane potential to the position of Z=h (the position of origin), by default $h = 0$.

Now you should be able to easily understand the following:

```
%BLOCK LATTICE_CART
20 0 0
0 20 0
0 0 20
#FIX
%ENDBLOCK LATTICE_CART

%BLOCK POSITIONS_FRAC
Al 0.0 0.0 0.0 # Al1 % NUM=7-13 COORD=4
%ENDBLOCK POSITIONS_FRAC

#MINSEP=1.5
#CLUSTER
#OVERLAP=0.2
#RASH
#POSAMP=3.0
#MINBANGLE=80
#MAXBANGLE=120
```

In addition, the data block corresponding to the structure data may not appear in the *.cell file at all. For example, you only have some vague understanding of the approximate structure of the crystal (for example, you only know the atomic composition of the crystal, the approximate volume of the crystal lattice, etc.), you can still use AIRSS for structure search. Here are two AIRSS structure seed files that conform to the specification and are very concise.

例 *4.1*

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

例 *4.2*

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

Here, `#VARVOL` is used to replace the lattice parameter data block `BLOCK LATTICE_CART`, and `#SPECIES` is used to replace the atomic position data block `BLOCK POSITIONS_FRAC`.

4 Structural relaxation and energy calculation

4.1 Relax with airss-pp3 module

When using AIRSS combined with `airss-pp3` to calculate the prediction structure of the module, in addition to preparing a *.cell file, you also need to prepare a file named *.pp.

`airss-pp3` is the pp3 pair potential calculation module that comes with AIRSS. Its function is to use the classical molecular potential field (such as 6-12 potential) to relax the atomic structure and output the energy of the system. This module uses the crystal energy model of the dimensional image and does not involve any first-principle complex calculations, so its implementation is simple, the effect is stable, and the calculation speed is extremely fast. It is suitable for small systems with simple components. More specifically, `airss-pp3` uses the following potential energy to calculate the force on the atom and the total energy of the system:

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

$$E = \sum_{i<j}^{\text{all ions}} E_{ij} \tag{1b}$$

Where, i, j calibrates different atomic positions. When the types of elements are different, variables such as ε_{ij}, σ_{ij} correspond to different values.

When using this module, you need to configure a parameter file named *.pp first. This file stores the relevant parameters of the potential, and its internal writing form is as follows:

```
n_spec m n range
specs
# Epsilon
eps_11 eps_12 ...
eps_22 ...
...
# Sigma
sgm_11 sgm_11 ...
sgm_11 ...
...
# Beta
beta_11 beta_12 ...
beta_22 ...
...
```

Among the above parameters,

1. The items in the first line are: the number of elements; the value of the exponent m, n in the potential; the distance from the center of the atom at the minimum energy value (where the force is 0) in the potential d_{min} and the ratio of σ, that is, $d_{min}^{(ij)} = \sigma_{ij} * d_{range}$.
2. The second line specifies the types of elements in the system, and different elements are separated by spaces.
3. Line 3 is a comment line, meaningless in the program, but must exist.
4. Lines 4 to $(n_{spec}+3)$ declare the matrix corresponding to the ε value between the elements.
5. The line $(n_{spec}+4)$ is a comment line, meaningless in the program, but must exist.
6. Lines $(n_{spec}+5)$ to $(2n_{spec}+4)$ declare the matrix corresponding to the value of σ between the elements.
7. The line $(2n_{spec}+5)$ must be a comment containing the character "Beta". If this character does not appear, the system will force all β values to 1.
8. Lines $(2n_{spec}+6)$ to $(3n_{spec}+5)$ declare the matrix corresponding to the β value between the elements.

例如,

```
2 12 6 5
A B
# Epsilon
1.00 1.50
0.50
# Sigma
2.00 1.60
1.76
```

Let's use an example to demonstrate the calculation process of AIRSS combined with pp3.

```
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 -1
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$
```

4.2 Relaxed with CASTEP

It is officially recommended that AIRSS be used in combination with CASTEP, and a very complete interface between AIRSS and CASTEP is constructed.

To use CASTEP combined with AIRSS calculation, first you need to copy the successfully installed CASTEP executable file `castep.serial` or `castep.mpi` to the `bin` directory of AIRSS, and rename it to `castep`.

When using CASTEP combined with AIRSS calculation, in addition to *.cell, you also need to prepare *.param files. *.param is the configuration file of CASTEP, in which you can define the necessary configuration parameters during the calculation of CASTEP, including, the type of calculation (Structure optimization, self-consistency, calculation of optical properties, band calculation, etc.), charge, spin orientation, truncation energy, convergence criteria, etc. The file usually consists of several lines, each line contains a `keyword` and its corresponding assignment.

***.param files mainly have the following characteristics:**

1. There is no restriction on the writing order between any two keywords.
2. You can use # or; or! Or the word COMMENT to add comments.
3. All keywords and data set in *.param are not case-sensitive. At the same time, any punctuation (except for symbols that indicate the content of the comment), extra spaces and any blank lines will be automatically ignored.
4. At most one keyword and its corresponding parameter can appear in any line of the file.

The basic format of each line of the *.param file is:

```
[keywords] : [value]
```

Among them, ':' is added for beautiful writing and easy to distinguish content. It will be automatically ignored when the program is actually executed. You can also skip this symbol at all and replace it with a space.

For the definition and usage of `[keywords]` in CASTEP, please refer to: CASTEP cell keywords and data blocks.

AIRSS combines CASTEP calculation by default, so run the following command to start the structure search.

```
1 user@machine_name$ airss.pl -max 3 -seed Al
```

4.3 Relax with VASP

AIRSS comes with the `airss.pl -vasp` option for joint VASP relaxation calculation. However, the official interface needs to be improved in terms of parallel computing. Based on AIRSS, the author rewritten AIRSS with a set of bash commands The interface to VASP is named airss4vasp(a4v). Although I have considered this command set to be rewritten in python, but AIRSS itself cannot be used on Windows However, the replication project is too large and has little effect, so a4v is still mainly based on bash.[12]

a4v can be regarded as a revised version of AIRSS. It can run independently without installing original AIRSS. It is mainly based on AIRSS's `buildcell` module, and it also embeds PBS, NSCC, Slurm and other job submission system commands, which truly achieves one Key to submit the function of AIRSS+VASP task, and it also has better support for parallel calculation. For specific usage, please refer to the file `README.md` inside the project. [13]

In the *.cell file settings: a4v adds `SD-XYZ`, `SD-XY`, `SD-X` and other atom labels that correspond to the fixed atomic position relaxation; at the same time deletes the original Set the global parameters of the collinear spin value in buildcell, `#SPIN=`. In addition, because VASP itself is not very friendly to cluster calculations, a4v does not integrate cluster calculation components (such as symmetry judgment components, etc.).

[12] If you compare C++ to classical Chinese, python is equivalent to modern vernacular, and shell languages such as bash are spoken. Therefore, it is generally recommended to use python to write large-scale scripts.

[13] In fact, the instruction manual is also in this project In. (laughs

5 Data post-processing

5.1 *.res file structure

The calculation results of AIRSS are all stored in the *.res file. This *.res structure file was first used in SHELX. Due to some historical reasons, it was reused by CASTEP and AIRSS. Because SHELX itself is a Windows-friendly program, the writing format of its input and output files also follows the characteristics of some Windows documents, such as the tendency to use uppercase letters, `REM` stands for comment lines, and the file ends with `END`.

```
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
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
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$
```

The meaning of each line in the *.res file output by AIRSS is as follows:

1. The first item in the first line of TITL is the name tag assigned by the software to the structure, the second item is the system's additional hydrostatic pressure (GPa), the third item is the unit cell volume, and the fourth item is the total of each unit cell Enthalpy (energy), the fifth term is the average value of the atomic spin value, the sixth term is the average value of the absolute value of the atomic spin, the seventh term is the total number of atoms in the system, and the eighth term is the name of the space group where the system is located. The last item is the fixed character n-1.
2. Afterwards, several lines beginning with REM are comment lines, which record the basic information generated by the file, and will not have any effect after deletion.
3. The line beginning with CELL records the basic unit cell information. Among them, the first item is a meaningless decimal, which is used to record the wavelength of the diffraction spectrum used to obtain the relevant structure in the original SHELX program. In AIRSS The middle is locked as a meaningless decimal. The second to fourth terms are lattice constants a b c, and the next five to seven terms are crystal anglesα β gamma.
4. The bottom line is LATT -1. This line is used in SHELX to calibrate the symmetry of the lattice, and it is locked to a fixed value of -1 in AIRSS.
5. The next line starting with SFAC records the names of all the elements that make up the system, and different elements are separated by a space.
6. The last few rows mark the positions of the atoms in the unit cell. The first column of these rows is the element symbol, the second column indicates the order in which the element appears in the SFAC row, and the third to fifth columns are the rows where the atoms are stored Fractional coordinates, the last column is the number of atoms occupied, generally set to 1.
7. The file ends with an END line.

5.2 Data batch processing

Once you have the calculation data *.res file, you can use the `ca` command to process the data. `ca` is the package of the basic analysis suite `cryan` in AIRSS.

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

`cryan` is used as follows:[14]

```
user@machine_name$ cryan

Usage: cryan [OPTIONS]
The str. are read from STDIN, for example:
```

[14]For the sake of brevity, the description of the parameters has been simplified a little, please run the above command yourself to view more detailed information.

```
     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 str.
 -s,  --summary                         Summary
 -e,  --enthalpy <length_scale>         Plot enthalpy vs. pressure
 -f,  --formula <formula>               Select str. of a given com.
 -fc,  --formula_convert <formula>      Attempt to convert.
 -t,  --top [num]                       Output top few results
 -u,  --unite <thresh>                  Unite similar str.
 -dr, --distance <rmax>                 Distance threshold
 -de, --delta_e <energy>                Ignore str. above energy
 -sd, --struc_dos <smear>               Plot a structural DOS
 -p,  --pressure <pressure>             Additional pressure
 -m,  --maxwell                         Extract the stable com.
 -ph, --pressure_hull                   Ext. the stable str. with P
 -<n>                                   Component <n>
 -xg, --xmgrace                         Plot output with xmgrace
 -c,  --compare <thresh> <structure>    Compare structures
      --delete                          Delete unwanted str.
 -g,  --geometry [thresh]               Calculate the atomic geometry
 -n,  --num_units                       Report n separate str.
 -d,  --dimensionality                  Report dD str.
 -cl, --cluster                         No periodic boundary
 -bl, --bondlength                      Maximum bond length
 -bs, --bondscale                       Bond length scaling
 -dm, --deltamodularity                 Modularity bias parameter
 -wt, --weight                          Weight the adjacancy matrix
 -ns, --notsymm                         Clusters  point group off
 -sc, --struct_comm <thresh>            Determine the community str.
 -cm, --community                       Output the community str.
 -am, --adjacancymatrix                 Output the adjacancy matrix
 -x,  --xyz                             Output clusters in XYZ format
 -o,  --off                             Output polyhedra in OFF
 -al, --alpha                           Construct alpha shapes
 -l,  --long                            Long names for str.
 -h,  --help, -?                        Print usage information
user@machine_name$
```

Use ca to analyze the previous calculation results.

```
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$
```

In the above output result,

1. The first column is the name tag assigned by the AIRSS software to the structure.
2. The second column is the hydrostatic pressure value of the system(GPa).
3. The third column is the volume of each chemical formula structural unit.
4. The first row of the fourth column is the total energy (enthalpy, eV/f.u.) of the corresponding structure, and the following rows are the energy difference relative to the first row under different structures.
5. The fifth column is the total number of chemical formula structural units in the unit cell.
6. The sixth column is the chemical formula of the structural unit.
7. The seventh column is the space group name.
8. The eighth column is the number of times the structure appears in all search results.

If you think there are too many results listed, you can use the `-u` option, but note that `-u` must be used before `-r`. [15] If you read the help information of `cryan`, it is not difficult to find such a reminder: "note-the order matters".

```
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$
```

The number following the command `ca -u` is a dimensionless ratio. This parameter can be simply understood as follows: It calibrates the lattice similarity threshold. The larger the value, the higher the tolerance. In the end, there are fewer different structures displayed. In more detail, the distance calibrated by this parameter is a multiple of the distance between the two closest atoms inside the atomic structure. For example, the two existing structures are very similar, and the shortest in the crystal lattice The bond length is 1.5 (Å), then `ca -u 0.1` means that the pairwise distances of the corresponding atoms of the two structures are compared in turn, if it is not found that these distance differences are greater than 0.15 (Å) In this case, the two structures are the same as the calibration. [16] This value can be adjusted according to needs, and it is recommended to choose between 0.1–0.01.

[15] In fact, all rows The parameters after the rank (-r) task are automatically ignored.

[16] This paragraph description is just a superficial and intuitive explanation. The algorithm actually used is more complicated and stable. For details, see the source code "airss-0.9.1/src/cryan/src/ cryan.f90" line 2133.