CTR-tools users' manual

Y. Wakabayashi
Tohoku University

4th Mar. 2022

Contents

1 What is CTR-tools 1

2 Required environment 1

3 How to use 1

3.1 out_xyz.py 1

3.2 check_ctr.py 2

1 What is CTR-tools

This software supports to make the input files for CTR-structure. "out_xyz.py" converts the substrate and film structure files to xyz- or cif-format. "check_ctr.py" makes a list of the atoms too close to other atoms. (Other files are called from them. Do not delete.)

2 Required environment

The software is working under Python3.9 using numpy 1.21.2 and periodictable 1.6.1.

3 How to use

3.1 out_xyz.py

`out_xyz.py` converts *substrate.dat* and *surf.dat* to xyz- or cif-format.

Here, we assume that the "src" folder is located at the home directory.

`> python /src/out_xyz.py` *substrate.dat surf.dat* `>` *whole.xyz*

Here, the file *whole.xyz* contains all the atoms in the *substrate.dat* and *surf.dat* files. Unfortunately, xyz-format is not very good at describing occupancy. For this reason, we prepared an

option to make cif-format output.

> python /src/out_xyz.py *substrate.dat surf.dat* --cif > *whole.cif*

You can use the cif file for presentation of the final result.

If you want to make larger structure in the in-plane direction, you can use -x and -y options.

> python /src/out_xyz.py -x m -y n *substrate.dat surf.dat* --cif > *whole.cif*

It provides the cif file expanded m and n times in the a and b directions.

3.2 check_ctr.py

check_ctr.py checks the interatomic distance provided by the two input files, *substrate.dat* and *surf.dat*. The result is provided to the standard output.

> python /src/check_ctr.py *substrate.dat surf.dat* -t x

If there is no atomic pair closer than x Å, you will obtain

```
Checking substrate and surface data...
    Checking distance between substrate and surface...
    OK.
    Checking distance in substrate...
    OK.
    Checking distance in surface...
    OK.
Finished.
```

The default value of x is 1.

If you have pairs that are too close to each other, you will have something like this:

```
Checking substrate and surface data...
    Checking distance between substrate and surface...
    Warning: substrate (1) and surface ([4]) are too close
    Checking distance in substrate...
    Warning: substrate (1) and (2) are too close
    Warning: substrate (1) and (3) are too close
    Warning: substrate (1) and (4) are too close
    Checking distance in surface...
    Warning: surface ([2, 3]) and ([4]) are too close
    Warning: surface ([2, 3]) and ([5]) are too close
(snip)
```

```
    Warning: surface ([79, 80]) and ([82]) are too close
    Warning: surface ([79, 80]) and ([83]) are too close
Finished.
```

In this case, atoms 2 and 3 in the surface file have the same coordinate; they should be expressed by the fractional occupancy *occf*.

The atomic coordination is exported by using `--out_coord` option.

`> python /src/check_ctr.py` *substrate.dat surf.dat* `--out_coord`

You will have this kind of output:

```
Output atomic species and coordinates in substrate.dat
    0: Sr at (-0.00000, -0.00000, -3.90500)
    1: Ti at (1.95250, 1.95250, -1.95250)
    2: O at (1.95250, 1.95250, -3.90500)
    3: O at (0.00000, 1.95250, -1.95250)
    4: O at (1.95250, -0.00000, -1.95250)

Output atomic species and coordinates in surf.dat
    0: ['Sr(0)', 'La(1)'] at (0.00000, 0.00000, 0.00000)
    1: ['Ti(2)', 'Al(3)'] at (1.95250, 1.95250, 1.95250)
    2: ['O(4)'] at (1.95250, 1.95250, 0.00000)
    3: ['O(5)'] at (0.00000, 1.95250, 1.95250)
    4: ['O(6)'] at (1.95250, 0.00000, 1.95250)
          :
          :
```

Acknowledgements

This software development was supported by the MEXT Elements Strategy Initiative to Form Core Research Center (Grant No. JPMXP0112101001).