



```

      AA      DDDDD      AA      II      SSSS
    A  A  DD  DD  A  A  II  SS  SS
  A  A  DD  DD  A  A  II  SSS
AAAAAA DD  DD  AAAAAA II  SSS
AA  AA DD  DD  AA  AA II  SS  SS
AA  AA DDDDD  AA  AA II  SSSS  V1.0
AUTOMATIC DERIVATION OF ANISOTROPIC IDEAL STRENGTH

```

MANUAL

Version 1.0.0, June 16, 2018

Edited by

Shihao Zhang

Ruifeng Zhang

<https://sites.google.com/site/zrfbuaa/softwares/adais>

DSASM Lab
DESIGN AND SIMULATIONS OF ADVANCED STRUCTURAL MATERIALS

Content

Content.....	I
1. Features and history of ADAIS.....	1
1.1 Overview.....	1
1.2 Features of ADAIS	1
1.3 Version history.....	2
2. Getting started.....	4
2.1 How to obtain ADAIS	4
2.2 Necessary citations.....	4
2.3 How to install ADAIS.....	4
2.3.1 Install of ADAIS	4
2.3.2 Install of slightly modified VASP.....	5
2.4 How to run ADAIS	6
2.4.1 All input modes in ADAIS	6
2.4.2 Input modes in ADAIS	7
2.5 VASP&ADAIS calculation	18
3. Theoretical background	19
4. Overview of input and output files	20
4.1 All input and output files	20
4.2 Input and output files	21
5. Example of shell file	28

5.1 Affine deformation.....	28
5.2 Alias deformation.....	29
5.3 High-throughput calculation	31
References.....	34

1. Features and history of ADAIS

1.1 Overview

ADAIS is an efficient open source command-line program for Automatic Derivation of Anisotropic Ideal Strength via high-throughput first-principles computations. It is designed by Dr. S. H. Zhang and Prof. R. F. Zhang at Beihang University. ADAIS runs on LINUX and MAC operating systems. Its compilation needs free or commercial Fortran compiler. It is contributed free of charge for non-commercial users.

1.2 Features of ADAIS

- The tensile and shear ideal strengths of 3D materials via affine pure and simple deformations
- The uniaxial and biaxial tensile ideal strengths of 2D materials via affine tensile deformation
- Universal binding energy via alias tensile deformation
- Generalized stacking fault energy (γ -surface) via alias shear deformation
- Any crystal system of 3D and 2D materials
- Projection operation on crystal for specific crystallographic orientation parallel to three axes
- Rotation operation on crystal to specific orientation along three axes
- Redefinition of new lattice vector to specific crystallographic orientation
- Force convergence judgement for ideal strength calculation via affine deformation
- The standardized unit cell (IEEE-format)
- Automatic determination of the possible slip system according to space group
- Automatic setups of K-points density according to the reciprocal space or atomic number

- Automatic setups of INCAR file based on the input pressure
- The variation of bond length as a function of strain for affine deformation
- Supercell building for any crystal
- Primitive cell for any crystal

1.3 Version history

V_1.0.0

(2018.6.16)

The tensile and shear ideal strengths of 3D materials via affine pure and simple deformations

The uniaxial and biaxial tensile ideal strengths of 2D materials via affine tensile deformation

Universal binding energy via alias tensile deformation

Generalized stacking fault energy (γ -surface) via alias shear deformation

Any crystal system of 3D and 2D materials

Projection operation on crystal for specific crystallographic orientation parallel to three axes

Rotation operation on crystal to specific orientation along three axes

Redefinition of new lattice vector to specific crystallographic orientation

Force convergence judgement for ideal strength calculation via affine deformation

The standardized unit cell (IEEE-format)

Automatic determination of the possible slip system according to space group

Automatic setups of K-points density according to the reciprocal space or atomic number

Automatic setups of INCAR file based on the input pressure

The variation of bond length as a function of strain for affine deformation

Supercell building for any crystal

Primitive cell for any crystal

Further development

Disturbed method for the ideal strength calculation via affine deformation

Surface energy of any crystallographic plane

Analysis of phonon instability mode based on atomic displacement

Twin structure building

More available computer packages including VASP, QE, WIEN2K ...

2. Getting started

2.1 How to obtain ADAIS

ADAIS is an open source code and can be downloaded at:

<https://sites.google.com/site/zrfbuaa/software/adais>

2.2 Necessary citations

Whenever using ADAIS in all publications and reports, you must cite the original papers, for example, in the following way:

The ideal strength via affine pure and simple deformations for three-dimensional (3D) materials, the ideal strength via affine uniaxial and biaxial tensile deformations for two-dimensional (2D) materials, the universal binding energy via alias tensile deformation, and the generalized stacking fault energy (GSFE) via alias shear deformation were calculated using the ADAIS code [1], which is an automatic derivation of anisotropic ideal strength via high-throughput first-principles computations.

2.3 How to install ADAIS

2.3.1 Install of ADAIS

The compiler “cmake” is needed before installing ADAIS, and more information can be found via:

<https://cmake.org/>

After expanding source code, and go into the source code directory:

```
tar zxvf ADAIS-x.x.x.tar.gz
cd ADAIS-x.x.x
```

The current directory is ADAIS_SOURCE_DIR.

```
cd spglib-x.x.x
mkdir _build; cd _build
cmake ..
make
make install
```

The libraries are installed at ADAIS_SOURCE_DIR/spglib-x.x.x/lib and the header file is installed at ADAIS_SOURCE_DIR/spglib-x.x.x/include

```
cd ../../src
make -f makefile
```

Then the executable file ADAIS will be created in the folder of ADAIS_SOURCE_DIR/bin

2.3.2 Install of slightly modified VASP

To calculate the ideal strength via affine pure and simple deformations for three-dimensional (3D) materials, both the atomic coordinates and the atomic basis vectors are relaxed by modifying the five relevant strain components while remaining the applied strain component unchanged. The relaxation will be terminated when the five conjugate Hellmann-Feynman stresses reaches negligible values. While to calculate the ideal strength via affine uniaxial and biaxial tensile deformations for two-dimensional (2D) materials, besides the applied strain component remains unchanged, the components related to z-axis (perpendicular to the atomic layers of 2D materials), *i.e.*, ϵ_{zz} , ϵ_{yz} and ϵ_{zx} , also remain unchanged during relaxation.

In order to implement such relaxation automatically, one need to slightly modify the VASP optimizer with cell constraints. In ADAIS, four modified `constr_cell_relax.F` files are provided in ADAIS_SOURCE_DIR/src/ to meet different conditions: *i.e.*, `constr_cell_relax_3d_ten.F` and

`constr_cell_relax_3d_shear.F` for the calculation of the tensile and shear ideal strengths of 3D materials via affine pure and simple deformations, and `constr_cell_relax_2d_uniax.F` and `constr_cell_relax_2d_biax.F` for the calculation of the uniaxial and biaxial tensile strengths for 2D materials via affine tensile deformation, respectively.

Before compiling the modified VASP, the original VASP source `constr_cell_relax.F` needs to be replaced by the one `constr_cell_relax_xxx_xxx.F`, by renaming it to `constr_cell_relax.F`. Then, please compile the VASP package and get the executable file `vasp` or `vasp_std`. Finally, rename the executable file to `vasp_xxx_xxx`. Note that the original VASP is also necessary for the calculation of other properties via alias deformation obtained in ADAIS code.

2.4 How to run ADAIS

2.4.1 All input modes in ADAIS

A short overview of all input modes currently supported in ADIAS code, together with a short description.

Mode Name	Description
affine	Applying affine deformation to the structure of 3D and 2D crystalline materials
alias	Applying alias deformation to the structure of 3D and 2D crystalline materials
proj	Projecting the specific orientation of the given crystal to be parallel to three axes x, y and z
newlat	Redefining new right-handed lattice vectors of the current crystal
ieee	Transforming the structure to IEEE-format for 3D and 2D crystalline materials
super	Building the supercell of the current crystal
output	Reading the energy and stress from the VASP output file OUTCAR after first-principles calculation
fconv	Judging whether the relaxation of the structure is convergent or not
symm	Determining automatically the symmetry of structure and the possible slip system according to space group

bond	Analyzing the variation of bond length as a function of the strain value for affine deformation
incar	Creating INCAR file with a given pressure value
kpt	Creating the KPOINTS file used in VASP
help	Showing the help message, including version information, license information and user-friendly options of ADAIS
clean	Cleaning all the input and output files of ADAIS code

2.4.2 Input modes in ADAIS

- *Mode:* **affine**

Syntax:

ADAIS --**affine** <**affoption**> ten_str shear_str istr0 step_length step_num

Description:

This mode applies affine pure or simple deformation to the structure of 3D materials, and uniaxial or biaxial tensile deformation to the structure of 2D materials. A structure file named AFFPOS_[istr0].vasp or AFFPOS0 (only for istr0=0.0000) is needed, and then a structure file named AFFPOSI is created after applied affine pure or simple deformation for 3D materials, or uniaxial or biaxial tensile deformation for 2D materials.

In ADAIS code, affine tensile deformation to the structure of 3D materials will be applied to ϵ_{yy} , and affine pure or simple shear deformation will be applied to ϵ_{xy} . Therefore, before applying affine deformation, the specific orientation of the given crystal must be projected to be parallel to y-axis for the calculation of ideal tensile strength, and the slip plane and slip direction must be projected to be perpendicular to x-axis and parallel to y-axis for the calculation of ideal shear strength, respectively. For 2D materials, the uniaxial tensile deformation to the structure will be applied to ϵ_{yy} , and the biaxial tensile deformation will be applied to both ϵ_{xx} and ϵ_{yy} . Therefore, before applying affine deformation, the specific orientation of the given crystal for 2D materials must be projected to be parallel to

y-axis for the calculation of ideal tensile strength. To implement such projection automatically, a mode of `proj` is including in ADAIS code.

In order to ensure that the strain path is continuous, the starting position at each strain step is taken from the relaxed coordinates of the previous step. Therefore, except the first time, the CONTCAR file outputted after first-principles calculation via VASP, will be copy to AFFPOS0, then the affine pure or simple deformation will be applied to the structure of AFFPOS0, and a file named AFFPOSI will be output. Meanwhile, the CONTCAR will be also copied to the file of AFFPOS_[istrvalue].vasp as the relaxed structure of the previous step. A file named LOOPCAR, which includes the information of the calculation loop for affine or alias deformation, is also created. For more information of this file, one may refer to [Sec. 4](#).

<affoption>: `-pure3d` and `-simp3d`, which are corresponding to the affine pure and simple deformations of 3D materials, and `-unia2d` and `-biax2d`, which are corresponding to the uniaxial and biaxial tensile deformations of 2D materials, respectively.

`ten_str` and `shear_str` are the coefficient of the tensile and shear strain values, *i.e.*, the tensile and shear strain values per step applied to the crystal is `ten_str×step_length` and `shear_str×step_length`, respectively. Especially, `ten_str=0.00` is corresponding to the shear deformation, and `shear_str=0.00` is corresponding to the tensile deformation.

`istr0`, `step_length` and `step_num` are the initial strain value, the step length and the step number of affine deformation, respectively.

Examples:

```
ADAIS --affine -pure3d 0.00 1.00 0.00 0.02 21
```

```
ADAIS --affine -pure3d 1.00 0.00 0.00 0.02 10
```

```
ADAIS --affine -unia2d 1.00 0.00 0.00 0.01 42
```

```
ADAIS --affine -biax2d 1.00 0.00 0.00 0.01 42
```

- *Mode:* **alias**

Syntax:

ADAIS --**alias -dxy3d/-dxy2d** istartx iendx ispaddingx istarty iendy ispaddingy [zvalue]

ADAIS --**alias -dz** istartz iendz ispaddingz [zvalue]

Description:

This mode applies alias deformation to the structure of 3D and 2D crystalline materials. A structure file named ALIPOS0 is needed, and then a structure file named ALIPOS1 will be created after applied alias deformation. In addition, except the first time, the CONTCAR file output after first-principles calculation via VASP, will be copied to ALIPOS_[dx]_[dy].vasp or ALIPOS_[dz].vasp as the relaxed structure of the previous step. A file named LOOPCAR, which includes the information of the calculation loop for affine or alias deformation, is also created. For more information of this file, one may refer to [Sec. 4](#).

In the structure file ALIPOS1, the selective dynamics of all atoms is set as “F F T” to relax only the atomic positions perpendicular to the slip plane for affine shear deformation. While for the affine tensile deformation, to relax the positions of atoms nearby the cleavage plane only, the selective dynamics of the atoms away from the cleavage plane $\leq 4 \text{ \AA}$, is set as “T T T”, and for the others, it is set as “F F F”.

-dxy3d and -dxy2d are corresponding to the alias shear deformations in the x-y plane for calculating the generalized stacking fault energy (GSFE) of 3D and 2D crystalline materials, respectively. -dz is corresponding to the alias tensile deformation along the z axis for calculating the universal binding energy of 3D or 2D materials.

istartx(y), iendx(y) and ispaddingx(y) are the initial alias shear strain value, the end alias shear strain value and the shear strain spacing value for alias shear

deformation along $a(b)$ -axis, respectively (UNIT: 1), *i.e.*, the alias shear strain value of $i\text{startx}+i\times\text{spacingx}$ and $i\text{starty}+j\times\text{spacingy}$ will be applied, and $i=0, 1, 2, 3, \dots$ and $j=0, 1, 2, 3, \dots$, until $i\text{startx}+i\times\text{spacingx} > i\text{endx}$ and $i\text{starty}+j\times\text{spacingy} > i\text{endy}$.

$i\text{startz}$, $i\text{endz}$ and $i\text{spacingz}$ are the initial alias tensile displacement value, the end alias tensile displacement value and the tensile displacement spacing value for alias tensile deformation along z -axis, respectively (UNIT: Å), *i.e.*, the alias tensile strain value of $i\text{startz}+k\times\text{spacingz}$ will be applied, and $k=0, 1, 2, 3, \dots$, until $i\text{startz}+k\times\text{spacingz} > i\text{endz}$.

$z\text{value}$: the alias tensile and shear deformations will be applied to the atoms, whose z coordinate $> z\text{value}$. If the value of $z\text{value}$ is not specified by users, a default value of $z\text{value}$ will be determined automatically (UNIT: Å), for which there is a maximum layer spacing. It is because the slip plane or cleavage plane with a large interlayer distance is always prone to glide or cleavage due to the weak interaction between the layers.

Examples:

```
ADAIS --alias -dxy3d 0.0000 1.0000 0.1000 0.0000 1.0000 0.0666
10.0000
```

```
ADAIS --alias -dxy2d 0.0000 1.0000 0.1000 0.0000 0.0000 0.0000
10.0000
```

```
ADAIS --alias -dz 0.0000 0.1000 4.0000 10.0000
```

○ *Mode:* **proj**

Syntax:

```
ADAIS --proj -rot angx angy angz
```

```
ADAIS --proj -ind pvh pvk pvl uvu uvv uvw
```

```
ADAIS --proj -mat mat11 mat12 mat13 mat21 mat22 mat23 mat31 mat32 mat33
```

Description:

This mode projects one specific orientation (*i.e.*, the imposed strain direction) of a given crystal to be parallel to one axis vector (e.g. y) for affine tensile deformation, and projects one axis vector (e.g. x) to be perpendicular to the slip plane and another one (e.g. y) to be parallel to the slip direction on that plane for the affine shear deformation. A structure file named INPOS is necessary, and after projection, a structure file named PROJPOS will be created.

`-rot` is corresponding to the rotation method. For rotation method, three angles $\text{angx}(y, z)$ are corresponding to the contra-rotating angles along x , y and z axes, respectively (UNIT: $^\circ$).

`-ind` is corresponding to the crystallographic index method. For crystallographic index method, a reciprocal lattice vector $[pvh\ pvk\ pvl]^*$, perpendicular to the lattice plane $(pvh\ pvk\ pvl)$, and a lattice vector $[uvu\ uvv\ uvw]$ are necessary. And pvh , pvk , pvl , uvu , uvv and uvw must satisfy the condition: $pvh \cdot uvu + pvk \cdot uvv + pvl \cdot uvw = 0$, to ensure that the lattice vector $[uvu\ uvv\ uvw]$ lies on the $(pvh\ pvk\ pvl)$ lattice plane. Then the reciprocal lattice vector $[pvh\ pvk\ pvl]^*$ will be projected to be parallel to x -axis, and the lattice vector $[uvu\ uvv\ uvw]$ will be projected to be parallel to y -axis. Note that pvh , pvk , pvl , uvu , uvv and uvw must be integer.

`-mat` is corresponding to a generalized method of projection. mat_{ij} is the projection matrix (3×3), which is orthogonal normalized, and is determined as the initial atomic basis vector matrices \mathbf{R} transforming to the projected ones $\mathbf{R}^{proj} : \mathbf{R} = \mathbf{R}^{proj} \cdot [\text{mat}_{ij}]$. This matrix can be obtained from the 3D visualization program VESTA [2].

Examples:

```
ADAIS --proj -rot 30.00 0.00 30.00
```

```
ADAIS --proj -ind 1 1 1 1 1 -2
```

```
ADAIS --proj -mat 0.5774 0.5774 0.5774 0.4082 0.4082 -0.8165
-0.7071 0.7071 0.0000
```

- *Mode:* **newlat**

Syntax:

```
ADAIS --newlat vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32 vect33
```

Description:

This mode redefines new right-handed lattice vectors vect_{ij} of the current structure for 3D and 2D crystalline materials, then the new lattice vectors a_{new} and b_{new} will be put to be parallel to x -axis and to lie in x - y plane, respectively. A structure file named INPOS is necessary, and after redefining new lattice vectors, a structure file named REDPOS will be created. Note that the new lattice vectors vect_{ij} must be the existing lattice. However, in lattices with centering, the new lattice vectors can refer to centering points. In such lattices, the new lattice vectors can be specified using non-integer values.

A supercell structure will be also created using the structure file REDPOS, for which its component of the lattice vector c along z axis is larger than 15.0 Å. This supercell structure will be output to a file named SUPPOS, then this supercell can be used to calculate the GSFE and universal binding energy to eliminate any spurious interactions between the interface (*i.e.* the glide and cleavage plane) and its periodic images.

Examples:

```
ADAIS --newlat 0.5 -0.5 0.0 0.5 0.5 -1.0 1.0 1.0 1.0
```

This will redefine the new lattice vectors a , b and c along to the [1-10], [11-2] and [111] crystal directions for FCC structure, respectively.

- *Mode:* **ieec**

Syntax:

ADAIS --**ieee** -3d/-2d

Description:

This mode transforms the structure to IEEE-format as defined in Ref. [3]. -3d and -2d are corresponding to the 3D and 2D crystalline materials, respectively. A structure file named INPOS is necessary, and after transforming to IEEE-format, a structure file named IEEEPOS will be created.

- *Mode:* **super**

Syntax:

ADAIS --**super** sup_1 sup_2 sup_3

Description:

This mode builds the supercell of the current crystal. A supercell has lattice vectors which are integral multiples of their equivalents in the original lattice. And a structure file named INPOS is necessary, and after creating the supercell, a structure file named SUPPOS will be created. sup_1, sup_2 and sup_3 are the supercell ranges for the lattice vectors a , b and c , respectively. Note that sup_1, sup_2 and sup_3 must be integer and ≥ 1 .

Examples:

ADAIS --super 2 2 2

A $2 \times 2 \times 2$ supercell will be created.

- *Mode:* **output**

Syntax:

ADAIS --**output**

Description:

This mode reads the energy and stress from the VASP output file OUTCAR after first-principles calculation. In addition, the VASP output file CONTCAR is also necessary for the calculation of ideal strength via uniaxial and biaxial tensile deformation for 2D materials. Then the results will be output to the file RDAIS, with different information. For more information of this output file, one may refer to [Sec. 4](#).

- *Mode:* **fconv**

Syntax:

ADAIS --**fconv**

Description:

This mode judges whether the relaxation of the structure is convergent, *i.e.*, the conjugate Hellmann-Feynman stresses reach negligible values. The VASP output file OUTCAR after first-principles calculation is needed. If convergent, the ADAIS code will output 1, if not, 0 will be output. This mode is obtained only for affine deformation.

- *Mode:* **symm**

Syntax:

ADAIS --**symm** [symprec]

Description:

This mode determines the symmetry of structure. A structure file of INPOS is needed, and an output file SYMMCAR, including the symmetry information and the automatically determined possible slip system according to space group, will be created. Meanwhile, the primitive cell of the input structure will also be determined, and will be output to the structure file named PRIMPOS.

`symprec` is corresponding to the tolerance of the distance between atomic positions and the lengths of lattice vectors to be tolerated in the symmetry finding (UNIT: Å). A default value of 0.01 for `symprec` will be used if it is not specified by users.

Examples:

```
ADAIS --symm 0.0100
```

○ *Mode:* **bond**

Syntax:

```
ADAIS --bond bond_cut
```

Description:

This mode analyzes the variation of bond length as a function of the strain value for affine deformation. The output structure file `AFFPOS_[istrvalue].vasp` of `affine` mode is necessary. Two output files of `BONDCAR` and `RBOND` will be created. For more information of these two files, one may refer to [Sec. 4](#). This mode is obtained only for affine deformation.

The bond information is firstly obtained by analyzing the structure file `AFFPOS_[istr0].vasp` with a given value of `bond_cut`, the results will be output to the file `BONDCAR`. Then, the variation of bond length as a function of strain will be obtained using this bond information, and the results will be output to the file `RBOND`.

`bond_cut` is corresponding to the maximum value for bond length (UNIT: Å). In ADAIS code, a maximum value of 10.0 Å is allowed for `bond_cut`. Note that if the `bond_cut` is so small that no bond with length $\leq \text{bond_cut}$ exists between atoms, an error information will be output to the file `OUTFILE`.

Examples:

```
ADAIS --bond 1.8000
```

- *Mode:* **incar**

Syntax:

```
ADAIS --incar -rlx/stc [pressure]
```

Description:

This mode creates INCAR file with `PSTRESS=pressure` (UNIT: GPa), and a default value of 0.0000 will be used for `pressure` as it is not specified by users. `-rlx` and `-stc` are corresponding to the structure-relaxed mode and the structure-unrelaxed mode, respectively, and an INCAR file named `incarx` will be created. Note that when calculating the ideal strength, GSFE or universal binding energy at high pressure, the crystal structure needs to be fully relaxed at the corresponding pressure.

Examples:

```
ADAIS --incar -rlx
```

```
ADAIS --incar -stc 100
```

- *Mode:* **kpt**

Syntax:

```
ADAIS --kpt -kppra [kppra] [kscheme]
```

```
ADAIS --kpt -kspac [kspacing] [kscheme]
```

```
ADAIS --kpt -kmesh k_1 k_2 k_3 [kscheme]
```

Description:

This mode creates the KPOINTS file used in VASP. In ADAIS code, there are two automatic methods to specify the KPOINTS file: the *k*-points per reciprocal atom (KPPRA) [3] and the smallest allowed spacing between *k*-points (KSPACING). The KPPRA method is a way to automatically set the *k*-point mesh while keeping the *k*-point density constant

along the three axes despite of the variation of the unit cell. The number of mesh points along a given reciprocal lattice vector a_1 is set proportional to $|a_1 \cdot (a_2 \times a_3)| / |a_2 \times a_3|$, where a_2 and a_3 are the other two remaining reciprocal lattice vectors. The KSPACING method is another automatic method that makes the mesh as uniform as possible. According to this method, the number of k -points in the direction of three reciprocal lattice vectors is determined by the equation: $|a_i| / KSPACING$. A generalized way to determine the KPOINTS file is also included in ADAIS, by manually inputting the numbers of the subdivisions k_1 , k_2 and k_3 .

`-kppra` is corresponding to create the NEWKPT file with KPOINTS format according to the K-Point Per Reciprocal Atom of `kppra`. `kscheme` is the type mesh to use: Gamma shift (“Gamma”) or Monkhorst-Pack (“MP”). A default value of 1000 for `kppra`, and a default value of “Gamma” for `kscheme` will be used if they are not specified by users. Note that `kppra` must be integer.

`-kspac` is corresponding to create the NEWKPT file in KPOINTS format according to the KSPACING method of `kspacing`. A default value of 0.5 for `kspacing` will be used if it is not specified by users.

`-mesh` is corresponding to create the NEWKPT file with KPOINTS format according to the user’s input. Note that k_1 , k_2 and k_3 must be integer and ≥ 1 .

Note that a structure file INPOS is needed for `-kppra` and `-kspac` modes.

Examples:

```
ADAIS --kpt -kppra 1000 Gamma
```

```
ADAIS --kpt -kspac 0.5 MP
```

```
ADAIS --kpt -kmesh 10 10 10 Gamma
```

- *Mode:* [help](#)

Syntax:

ADAIS --**help**

Description:

This mode shows the help message, including version information, license information and user-friendly options of ADAIS.

- *Mode:* **clean**

Syntax:

ADAIS --**clean**

Description:

This mode cleans all the input and output files of ADAIS code, including the result files, *i.e.*, RDAIS, RBOND and OUTFILE.

2.5 VASP&ADAIS calculation

Two bash shell codes (*i.e.*, *example/example_affine.sh* for the calculation of ideal strength via affine deformation, and *example/example_alias.sh* for the calculation of GSFE and universal binding energy via alias deformation) have been provided for the pre-processing, first-principles calculations and post-processing, and no human intervention is required until the final results are obtained. A bash shell code (*i.e.*, *example/BatchCal.sh*) is also provided to calculate the ideal strength, GSFE and universal binding energy in batches. These three files are also shown in [Sec. 5](#).

3. Theoretical background

For this section, one may refer to our publication: S. H. Zhang, Z. H. Fu and R. F. Zhang, ADAIS: Automatic Derivation of Anisotropic Ideal Strength via high-throughput first-principles computations, in preparation.

4. Overview of input and output files

4.1 All input and output files

Here is a short overview of all input and output files currently included in ADAIS code, together with a short description.

File Name	Description
AFFPOS0	Input structure file of <code>affine</code> mode
AFFPOSI	Output structure file of <code>affine</code> mode after applying affine deformation
AFFPOS_[istrvalue].vasp	Relaxed structure file after first-principles calculation under affine deformation with a strain value of [istrvalue]
ALIPOS0	Input structure file of <code>alias</code> mode
ALIPOSI	Output structure file of <code>alias</code> mode after applying the alias deformation
ALIPOS_[dx]_[dy].vasp	Relaxed structure file after first-principles calculation under alias shear deformation with displacements of [dx] and [dy]
ALIPOS_[dz].vasp	Relaxed structure file after first-principles calculation under alias tensile deformation with a displacement of [dz]
INPOS	Input structure file used in ADAIS
PROJPOS	Projected structure via <code>proj</code> mode
REDPOS	Structure after redefining new lattice vector via <code>newlat</code> mode
SUPPOS	Structure of supercell via <code>super</code> mode
IEEEPOS	Structure with IEEE-format via <code>ieee</code> mode
PRIMPOS	Structure of primitive cell via <code>symm</code> mode
incarx	INCAR file with or without ionic relaxation, created via <code>incar</code> mode
NEWKPT	Created KPOINTS file via <code>kpt</code> mode
RDAIS	Results of <code>output</code> mode, including the calculated energy and stress as a function of strain via affine deformation, the calculated GSFE as a function of dx and dy, or universal binding energy as a function of dz via alias deformation for 3D and 2D crystalline materials.
BONDCAR	Results of <code>bond</code> mode, including the bond information of the structure file AFFPOS_[istr0].vasp with a given <code>bond_cut</code> value
RBOND	Results of <code>bond</code> mode, including the variation of bond length as a function of strain value under affine deformation
OUTFILE	Information of the calculated progress, as well as the error information if an error happens
LOOPCAR	Information of the calculation loop for affine or alias deformation
SYMMCAR	Information of the symmetry and the possible tensile crystal direction and shear slip system based on space group via <code>symm</code> mode

4.2 Input and output files

- File **AFFPOS0**

Meaning: The input structure file of `affine` mode.

Format: Standard POSCAR file used in VASP.

- File **AFFPOSI**

Meaning: The output structure file of `affine` mode after applying affine deformation.

Format: Standard POSCAR file used in VASP.

- File **AFFPOS_[istrvalue].vasp**

Meaning: The relaxed structure file after first-principles calculation under affine deformation with a strain value of [istrvalue].

Format: Standard POSCAR file used in VASP.

- File **ALIPOS0**

Meaning: The input structure file of `alias` mode.

Format: Standard POSCAR file used in VASP.

- File **ALIPOSI**

Meaning: The output structure file of `alias` mode after applying alias deformation.

Format: Standard POSCAR file used in VASP.

- File **ALIPOS_[dx]_[dy].vasp**

Meaning: The relaxed structure file after first-principles calculation under alias shear deformation with displacements of [dx] and [dy] along *x* and *y* axes, respectively.

Format: Standard POSCAR file used in VASP.

- File **ALIPOS_[dz].vasp**

Meaning: The relaxed structure file after first-principles calculation under alias tensile deformation with a displacement of [dz] along z axis.

Format: Standard POSCAR file used in VASP.

- File **INPOS**

Meaning: The input structure file used in ADAIS.

Format: Standard POSCAR file used in VASP.

- File **PROJPOS**

Meaning: The projected structure via `proj` mode.

Format: Standard POSCAR file used in VASP.

- File **REDPOS**

Meaning: The structure after redefining new lattice vector via `newlat` mode.

Format: Standard POSCAR file used in VASP.

- File **SUPPOS**

Meaning: The structure of supercell via `super` mode.

Format: Standard POSCAR file used in VASP.

- File **IEEEPOS**

Meaning: The structure with IEEE-format via `ieee` mode.

Format: Standard POSCAR file used in VASP.

○ File **PRIMPOS**

Meaning: The structure of primitive cell via `symm` mode.

Format: Standard POSCAR file used in VASP.

○ File **incarx**

Meaning: The INCAR file with or without ionic relaxation, created via `incar` mode.

Format: Standard INCAR file used in VASP.

○ File **NEWKPT**

Meaning: The created KPOINTS file via `kpt` mode.

Format: Standard KPOINTS file used in VASP.

○ File **RDAIS**

Meaning: The results of `output` mode, including the calculated energy and stress as a function of strain via affine deformation, the calculated GSFE as a function of `dx` and `dy`, or universal binding energy as a function of `dz` via alias deformation for 3D and 2D crystalline materials.

Format:

>>> Affine tensile and shear deformations for 3D materials

strain_value	Eng (eV)	Volume (Ang^3)	Stress (GPa)
0.000000	-80.9806125300	44.21000000	0.00000000
0.020000	-80.9530136400	44.21000000	10.00933100
0.040000	-80.8731017900	44.22000000	19.49519600
0.060000	-80.7457971800	44.24000000	28.45222900

>>> Uniaxial and biaxial tensile deformations for 2D materials

strain_value	Eng (eV)	Stress (N/m)
0.000000	-80.9806125300	0.00000000
0.020000	-80.9530136400	0.00350361
0.040000	-80.8731017900	0.00675691
0.060000	-80.7457971800	0.00976638

>>> Alias shear deformation for 3D and 2D crystalline materials

dx (Ang)	dy (Ang)	Eng (eV)	GSFE (J/m2)
0.000000	0.000000	-470.9515353100	0.00000000
0.033300	0.000000	-470.9146257500	0.01365303
0.066600	0.000000	-470.8086523200	0.05285314
0.099900	0.000000	-470.6318748700	0.11824401

>>> Alias tensile deformation for 3D and 2D crystalline materials

dz (Ang)	Eng (eV)	Ebinding (J/m2)
0.000000	-501.4387005000	0.00000000
0.100000	-501.1706935900	0.34504053
0.200000	-500.5278965300	1.17259770
0.300000	-499.6894706800	2.25201354

○ File **BONDCAR**

Meaning: The results of bond mode, including the bond information of the structure file AFFPOS_[istr0].vasp with a given bond_cut value.

Format:

The first row is the atomic serial number, the second row is the number of bond for the atom listed in the first row with a given bond_cut value, the remaining rows are the serial number (in the brackets) of atoms bonding with the atom listed in the first row, and the serial number of this atom in the 3×3×3 supercell.

ATOM(001)	4	(005)	24	(006)	14	(007)	18	(008)	26
ATOM(002)	4	(005)	23	(006)	14	(007)	14	(008)	23
ATOM(003)	4	(005)	15	(006)	14	(007)	15	(008)	14
ATOM(004)	4	(005)	14	(006)	14	(007)	17	(008)	17
ATOM(005)	4	(001)	4	(002)	5	(003)	13	(004)	14
ATOM(006)	4	(001)	14	(002)	14	(003)	14	(004)	14
ATOM(007)	4	(001)	10	(002)	14	(003)	13	(004)	11
ATOM(008)	4	(001)	2	(002)	5	(003)	14	(004)	11

○ File **RBOND**

Meaning: The results of bond mode, including the variation of bond length as a function of strain value under affine deformation.

Format:

For each atom, the first line is the atomic serial number and the number of bond, respectively. Next, the variation of bond length as a function of strain value is listed. The serial number of atoms bonding with this atom, is also listed in the brackets in the first line.

```

ATOM(001)      4
Strain  (005)  (006)  (007)  (008)
0.0000  1.5311  1.5311  1.5311  1.5311
0.0200  1.5275  1.5315  1.5275  1.5383
0.0400  5.3077  3.9240  5.3077  6.4325
0.0600  5.3489  3.9590  5.3489  6.4922

ATOM(002)      4
Strain  (005)  (006)  (007)  (008)
0.0000  1.5311  1.5311  1.5311  1.5311
0.0200  1.5383  1.5275  1.5315  1.5275
0.0400  4.6556  2.9601  2.9436  4.6169
0.0600  4.6866  2.9760  2.9495  4.6300

ATOM(003)      4
Strain  (005)  (006)  (007)  (008)
0.0000  1.5311  1.5311  1.5311  1.5311
0.0200  1.5275  1.5383  1.5275  1.5315
0.0400  1.5241  1.5453  1.5241  1.5328
0.0600  1.5209  1.5523  1.5209  1.5348

```

- File **OUTFILE**

Meaning: Information of the calculated progress, as well as the error information if an error happens.

- File **LOOPCAR**

Meaning: Information of the calculation loop for affine or alias deformation.

Format:

>>> Affine deformation

```
AFFINE    <affoption>
```

```
-----
| TEN_STR      : 0.0000
| SHEAR_STR    : 1.0000
| ISTR0        : 0.0000
| STEP_LENGTH  : 0.0100
| STEP_NUM     : 20
| LOOP_NUM     : 0020
```

>>> Affine shear deformation

```
ALIAS     dxy3d
```

```
-----
| ISTARTX      : 0.6000
| IENDX        : 1.0000
| ISPACINGX    : 0.2000
| ISTARTY      : 0.0000
| IENDY        : 0.0000
| ISPACINGY    : 0.0000
| ZVALUE       : 7.5000
| LOOP_NUM     : 0003
```

>>> Affine tensile deformation

```
ALIAS     dz
```

```
-----
| ISTARTZ      : 1.1000
| IENDZ        : 2.0100
| ISPACINGZ    : 0.1000
| ZVALUE       : 7.5000
| LOOP_NUM     : 0010
```

○ File **SYMMCAR**

Meaning: Information of the symmetry and the possible tensile crystal direction and shear slip system based on the symmetry via `symm` mode.

Format: First line lists the crystal system of the structure, second line lists the space group number, the international space group symbol and the schoenflies symbol, respectively. And the possible tensile crystal direction and shear slip system based on the space group are listed next.

```
Cubic system
227  Fd-3m    Oh^7
-----
| ! TENSILE CRYSTAL DIRECTION
|   1   0   0   ![100]
|   1   1   0   ![110]
|   1   1   1   ![111]
| ! SHEARING SLIP SYSTEM
|   0   0   1   1   0   0   ! (001) [100]
|   1   1   0   0   0   1   ! (110) [001]
|   1   1   0   1  -1   0   ! (110) [1-10]
|   1   1   1   1  -1   0   ! (111) [1-10]
|   1   1   1   1   1  -2   ! (111) [11-2]
|   1   1   1  -1  -1   2   ! (111) [-1-12]
```

```

rm AFFPOSI
${ADAIS_DIR}/bin/ADAIS --affine -pure3d ten_str shear_str istr0 step_length step_num
${ADAIS_DIR}/bin/ADAIS --affine -simp3d ten_str shear_str istr0 step_length step_num
${ADAIS_DIR}/bin/ADAIS --affine -unia2d ten_str shear_str istr0 step_length step_num
${ADAIS_DIR}/bin/ADAIS --affine -biax2d ten_str shear_str istr0 step_length step_num

if [ -s AFFPOSI ]; then
    cp AFFPOSI POSCAR
else
    break
fi

# To set up the iteration
for Niter in {1..6} # 6-maximum iteration number
do

    # To run VASP
    if [ -s INCAR ] && [ -s POSCAR ] && [ -s KPOINTS ] && [ -s POTCAR ]; then
        ${VASP_DIR}/vasp_3d_shear >> olog.vasp
        ${VASP_DIR}/vasp_3d_ten >> olog.vasp
        ${VASP_DIR}/vasp_2d_uniax >> olog.vasp
        ${VASP_DIR}/vasp_2d_biax >> olog.vasp
    else
        echo "Necessary files: INCAR POSCAR KPOINTS or POTCAR is not ready!!"
        continue
    fi

    # To check the convergence for termination

    fconv_idx=${${ADAIS_DIR}/bin/ADAIS --fconv}
    if [ ${fconv_idx} -eq 1 ] || [ ${Niter} -gt 5 ]; then

        # To generate results
        if [ -s OUTCAR ]; then
            ${ADAIS_DIR}/bin/ADAIS --output
        else
            echo "OUTCAR does not exist or is empty"
        fi
        break
    else
        cp CONTCAR POSCAR
    fi

# To end up the iteration
done
done

${ADAIS_DIR}/bin/ADAIS --bond bond_cut

done

#rm POSCAR KPOINTS* INCAR* POTCAR
#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR XDATCAR IBZKPT vasprun.xml

#>>> make clean
${ADAIS_DIR}/bin/ADAIS --clean

#END

```

5.2 Alias deformation

```

#!/bin/bash

#
# Purpose:
#
#
# Record of vevisions:
#   Data      Programmer      Description of change
#   ====      =====
#   2017/03/17  Shihao Zhang   Original code

```

```
# Variables:
#
#   ADAIS_DIR : The path of the source code of ADAIS
#   VASP_DIR  : The path of the compiled VASP
#   POT_DIR   : The path of the POTCAR file
#
#
#
#
ADAIS_DIR=''
VASP_DIR=''
POT_DIR=''

for SYS_name in structure_file
do
#>>> create POTCAR file

    cat ${POT_DIR}/POT_XXX > ./POTCAR

#>>> create INCAR file using ADAIS and the file incar_rlx and incar_stc are created

#>>> the pressure in 0 GPa (default value)
${ADAI$DIR}/bin/ADAI$ --incar -rlx
${ADAI$DIR}/bin/ADAI$ --incar -stc

#>>> and the pressure is 100 GPa
${ADAI$DIR}/bin/ADAI$ --incar -rlx 100
${ADAI$DIR}/bin/ADAI$ --incar -stc 100
cp incarX INCAR

#>>> redefine the lattice

cp ./structure/${SYS_name}.vasp ./INPOS
${ADAI$DIR}/bin/ADAI$ -ieee -3d
${ADAI$DIR}/bin/ADAI$ -ieee -2d
#cp IEEEPOS INPOS

${ADAI$DIR}/bin/ADAI$ --newlat vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32 vect33

#cp REDPOS INPOS
${ADAI$DIR}/bin/ADAI$ --super sup_1 sup_2 sup_3

cp SUPPOS ALIPOS0

#>>> create KPOINTS file using ADAIS and the file NEWKPT are created

cp SUPPOS INPOS
#>>> kmesh=10x10x10 with Gamma-centered ('G':default value) or Monkhorst-Pack ('M')
${ADAI$DIR}/bin/ADAI$ --kpt -kmesh 10 10 10 Gamma

#>>> KPPRA=1000(default value) with Gamma-centered ('G':default value) or Monkhorst-Pack ('M')
${ADAI$DIR}/bin/ADAI$ --kpt -kppra 1000 Gamma

#>>> KSPACING=0.5(default value) with Gamma-centered ('G':default value) or Monkhorst-Pack ('M')
${ADAI$DIR}/bin/ADAI$ --kpt -kspac 0.5 Gamma
cp NEWKPT KPOINTS

while ((1))
do

rm ALIPOSI
${ADAI$DIR}/bin/ADAI$ --alias -dxy3d istartx iendx ispacingx istarty iendy ispacingy zvalue
${ADAI$DIR}/bin/ADAI$ --alias -dxy2d istartx iendx ispacingx istarty iendy ispacingy zvalue
${ADAI$DIR}/bin/ADAI$ --alias -dz istartz iendz ispacingz zvalue

if [ -s ALIPOSI ]; then
    cp ALIPOSI POSCAR
else
    break
fi

# To run VASP
if [ -s INCAR ] && [ -s POSCAR ] && [ -s KPOINTS ] && [ -s POTCAR ]; then
    ${VASP_DIR}/vasp >> olog.vasp
```

```

        else
            echo "Necessary files: INCAR POSCAR KPOINTS or POTCAR is not ready!!"
            continue
        fi

        # To generate results
        if [ -s OUTCAR ]; then
            ${ADAIS_DIR}/bin/ADAIS --output
        else
            echo "OUTCAR does not exist or is empty"
        fi

done
done

#rm POSCAR KPOINTS* INCAR* POTCAR
#rm CHG* CONTCAR DOSCAR OSZICAR OUTCAR EIGENVAL PCDAT WAVECAR XDATCAR IBZKPT vasprun.xml

#>>> make clean
#${ADAIS_DIR}/bin/ADAIS --clean

#END

```

5.3 High-throughput calculation

```

#!/bin/bash

#
# Purpose:
#
#   A bash shell is used to ideal strength, GSFE or universal binding energy in batches,
#   three examples are including
#
# Record of vevisions:
#
#   Data          Programmer          Description of change
#   ====          =====
#   2017/03/17    Shihao Zhang        Original code
#
# Variables:
#
#
#
#
# Note:
#
#   A directory named "FILE" is needed including "${file}" file and "structure" directory. And the
#   structure files to calculate are including in the "structure" directory
#
#
#
# The file name of bash shell that realizes the pre-processing, first-principles calculations
# and post-processing. Such as /example/example_affine.sh for affine deformation and
# /example/example_alias.sh for alias deformation

file='example_affine.sh'
#file='example_alias.sh'

#>>> EXAMPLE 1

# The name of structure file to calculated

structure_file='diamond'

# The slip system, rotation angle or new lattice vector to calculate

for loop in '1 1 1 1 1 -2' '1 1 1 -1 -1 2'
#for loop in '10.00 0.00 0.00' '20.00 0.00 0.00'
#for loop in '0.5 -0.5 0.0 0.5 0.5 -1.0 1.0 1.0 1.0'
do

```

```

# Create directories and copy the structure file to the corresponding directory

loop_non=$(echo ${loop} | sed 's/\s\+/_/g')
mkdir ${loop_non}
mkdir ./${loop_non}/structure
cp ./FILE/structure/${structure_file}.vasp ./${loop_non}/structure/

# Substitute the character of "slip system", "rotation angle" or "new lattice vector"
#   in file "./FILE/${file}" using "${loop}"

sed "s/pvh pvk pvl uvu uvv uvw/${loop}/" ./FILE/${file} > ./${loop_non}/${file}
#sed "s/angx angy angz/${loop}/" ./FILE/${file} > ./${loop_non}/${file}
#sed "s/vect11 vect12 vect13 vect21 vect22 vect23 vect31 vect32
vect33/${loop}/" ./FILE/${file} > ./${loop_non}/${file}

# To submit job

cd ${loop_non}
chmod +x ${file}
qsub ${file}
cd ..

done

#>>> END of EXAMPLE 1

#>>> EXAMPLE 2

# The name of structure file to calculated

for loop in diamond graphene
do

    # Create directories and copy the structure file to the corresponding directory

    mkdir ${loop}
    mkdir ./${loop}/structure
    cp ./FILE/structure/${loop}.vasp ./${loop}/structure/

    # Substitute the character "structure_file" in file "./FILE/${file}" using "${loop}"

    sed "s/structure_file/${loop}/" ./FILE/${file} > ./${loop}/${file}

    # To submit job

    cd ${loop}
    chmod +x ${file}
    qsub ${file}
    cd ..

end

#>>> END fo EXAMPLE 2

#>>> EXAMPLE 3

# The name of structure file (from POS_001 to POS_100) to calculated

for loop in `seq -w 001 001 100`
do

    # Create directories and copy the structure file to the corresponding directory

    mkdir POS_${loop}
    mkdir ./POS_${loop}/structure
    cp ./FILE/structure/POS_${loop}.vasp ./POS_${loop}/structure/

    # Substitute the character "structure_file" in file "./FILE/${file}" using "${loop}"

    sed "s/structure_file/POS_${loop}/" ./FILE/${file} > ./POS_${loop}/${file}

    # To submit job

    cd POS_${loop}
    chmod +x ${file}

```

```
qsub ${file}
cd ..
```

done

```
#>>> END of EXAMPLE 3
```

References

- [1] S. H. Zhang, Z. H. Fu and R. F. Zhang, ADAIS: Automatic Derivation of Anisotropic Ideal Strength via high-throughput first-principles computations, in preparation.
- [2] K. Momma. VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data, Journal of Applied Crystallography 44 (2011) 1272-1276.
- [3] S. H. Zhang, R. F. Zhang. AELAS: Automatic ELAStic property derivations via high-throughput first-principles computation, Computer Physics Communications 220 (2017).