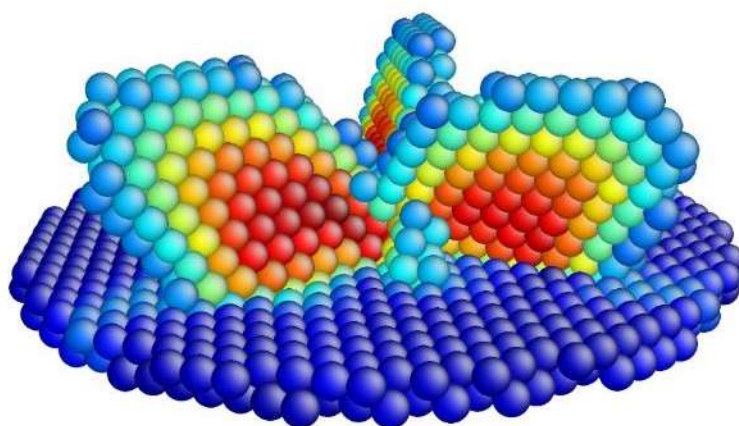


AADIS MANUAL

Version 1.0.0



2018-6-5

Bonan Yao
Ruifeng Zhang

contents

1 Introduction.....	1
1.1 Overview.....	1
1.2 Features of AADIS.....	1
1.3 Version history	1
1.4 Basic Syntax of the program.....	2
1.4.1 How to obtain AADIS.....	2
1.4.2 Necessary citations.....	2
1.4.3 How to run AADIS	2
1.4.4 Requirements	2
2 General command	4
2.1 Function command.....	4
2.2 General command	4
3 Private command	12
3.1 DVA	13
3.2 STA	15
3.3 DDA.....	18
3.4 SVA	21
3.5 IDA	23
3.6 NTA(assigned mode)	26
3.7 NTA (reference mode)	32
Appendices.....	34
A.1 Examples.....	34
A.1.1 Nanoindentation simulations in fcc-Au	35
A.1.2 CuNi interface under uniaxial tension	43
A.1.3 Misfit dislocation in CuNi interface.....	49
A.1.4 Screw dislocations in bcc-Fe.....	51
A.1.5 Edge dislocations in hcp-Mg.....	54
A.2 Precautions	55
A.2.1 O-lattice reference.....	55
A.2.2 coordination	57
References.....	58

1 Introduction

1.1 Overview

It is designed by Dr. B. N. Yao and Prof. R. F. Zhang at Beihang University. More details can be found at:

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

1.2 Features of AADIS

Functionalities:

The program has implemented the following methods

- 1) the displacement vector analysis (DVA)
- 2) the atomic strain tensor analysis (STA)
- 3) the differential displacement analysis (DDA)
- 4) the slip vector analysis (SVA)
- 5) the interplanar disregistry analysis (IDA)
- 6) the Nye tensor analysis (NTA) in assigned or reference mode

1.3 Version history

v_1.0.0

Functionalities:

The program has implemented the following methods

- 1) the displacement vector analysis (DVA)
- 2) the atomic strain tensor analysis (STA)
- 3) the differential displacement analysis (DDA)
- 4) the slip vector analysis (SVA)
- 5) the interplanar disregistry analysis (IDA)
- 6) the Nye tensor analysis (NTA) in assigned or reference mode

Further development

- Advanced mapping system

1.4 Basic Syntax of the program

1.4.1 How to obtain AADIS

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

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

1.4.2 Necessary citations

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

xxx) B. N. Yao and R. F. Zhang. AADIS: Atomistic Analyzer for DISlocations character and distribution, (2018) submitted.

1.4.3 Requirements

The program runs on LINUX, MAC and Windows operating systems. Its compilation needs free or commercial C++ compiler, but the compiled version for each operating system are provided in easy use.

1.4.4 Installation of AADIS

1. Unzip:

```
Star -zxvf AADIS_*.tar.gz
```

One may get the compiled AADIS program as executable file for three operating systems such as Linux, MAC and Windows. This will create an AADIS directory containing those files and several sub-directories:

bin	compiled AADIS program
doc	documentation
exp	simple tutorials
src	source files
AADIS.exe	compiled AADIS program for tutorials
LICENSE	the GNU General Public License (GPL)
makefile	For compile and link
README	Introduction

2. Making AADIS:

If you want to build AADIS yourself, then you may use the makefile file by typing the following lines (in AADIS_* directory):

\$make linux	#for linux system
\$make win32	#for windows system
\$make mac	#for mac system

1.4.3 How to run AADIS

The typical command lines of AADIS for each functionality looks like:

```
>> AADIS <INPAR>
```

(<INPAR> is the parameter file)

```
>> AADIS
```

(The default parameter file name "INPAR" will be used)

2 General command

2.1 Functional command

The following commands must be provided in the first line of the parameter file to activate the corresponding functionality mentioned above.

Note: the following commands must be written in the first line of the command file.

- | | |
|---|--------------|
| 1) The displacement vector analysis (DVA): | #dva |
| 2) The atomic strain tensor analysis (STA): | #sta |
| 3) The differential displacement analysis (DDA): | #dda |
| 4) The slip vector analysis (SVA): | #sva |
| 5) The interplanar disregistry analysis (IDA): | #ida |
| 6) The Nye tensor analysis (NTA) with assigned or reference mode: | #nye or #nta |

For STA, DDA, SVA and NTA method, the grid mesh is generated and the periodic boundary is then applied during running the program. In addition to the NTA method in assigned mode, DVA, STA, DDA, SVA, IDA and NTA method in reference mode require reference configuration. For STA and NTA method, the deformation gradient tensor of each atom will be computed. For NTA method, using #nye to activate the assigned mode without reference configuration, while using #nta to activate the reference mode based on reference configuration.

2.2 General command

These commands can be used in all methods. Most of them are used for input and output and shared for all methods.

read_datafile_def command

Syntax

`read_datafile_def file`

- file = name of the data file to read

Examples

`read_datafile_def disdata.lmp`

Description

Read a data file containing information needed to do an analysis. The file should be ASCII formatted. If no file name is provided, the program will use default file: DISFILE (no filename extension).

read_datafile_ref command

Syntax

```
read_datafile_ref file
```

Examples

```
read_datafile_ref refdata.lmp
```

In addition to the NTA method in assigned mode, all other analysis require reference configuration. Using *read_datafile_ref* to read the reference file. If no file name is provided, the program will use default file: REFFILE (no filename extension).

Requirements for input file:

1. The input file must contain at least the atom id, atom type, atomic coordinates x y z (5 columns required in total), while the atomic sequence must remain unchanged during analysis.
2. When the input file contains more than five columns, e.g. atomic potential energy, additional parameter, i.e. *read_prop_def* is needed. If no number is provided, the program will analyze it automatically. Using *read_prop_ref* command to give the number for the file containing the reference configuration and see below for details.
3. When there is header information in the input file, one may provide the number of header rows by command *read_line_def*. If no number is provided, the program will identify it automatically. Using *read_line_ref* command to give the number for the file containing the reference configuration and see below for details.

read_prop_def command**Syntax**

```
read_prop_def amount
```

- amount = number of additional columns

Examples

```
read_prop_def 2
```

Description

When the input file contains additional columns more than five such as atomic potential energy, the number of additional columns need to be provided. If no number is given, the program will analyze this number automatically. The purpose to keep this command is to handle the failure of automatic identification.

In addition to the NTA method in assigned mode, all other analysis require reference configuration. One may use *read_prop_def* to give the number of additional columns in the defective file, while use *read_prop_ref* to give the number of additional columns in the reference file.

For example, the following reference file should use command:

```
read_prop_ref 2
```

The program will know two additional columns in the reference file.

1	ITEM: TIMESTEP
2	0
3	ITEM: NUMBER OF ATOMS
4	2300
5	ITEM: BOX BOUNDS pp pp pp
6	-77.5423 77.5423
7	-134.629 134.629
8	-44.769 44.769
9	ITEM: ATOMS id type x y z c_eatom c_cnaf
10	1 1 -73.792 90.873 -44.044 -2.841 1
11	2 1 -76.297 90.877 -42.602 -2.849 1
12	3 1 -73.790 90.877 -41.158 -2.849 1
13	4 1 -76.291 90.877 -39.714 -2.849 1
.....

read_line_def command**Syntax**

```
read_line_def lines
```

- *lines* = number of header rows in data file to read

Examples

```
read_line_def 9
```

Description

The command is used to pass by the header information. The number of header rows information is required. If no number is provided, the program will identify it automatically. The purpose to keep this command is to handle the failure of automatic identification.

One may use *read_line_def* to give the number of header rows in the defective file, while use *read_line_ref* to give the number of header rows in the reference file.

For example, the following file should use command:

```
read_line_def 9
```

The program will ignore the front 9 lines and read atom information from the 10th line.

1	ITEM: TIMESTEP
2	0
3	ITEM: NUMBER OF ATOMS
4	2300
5	ITEM: BOX BOUNDS pp pp pp
6	-77.5423 77.5423


```

7      -134.629 134.629
8      -44.769 44.769
9      ITEM: ATOMS id type x y z c_eatom c_cnaf
10     1 1 -73.792 90.873 -44.044 -2.841 1
11     2 1 -76.297 90.877 -42.602 -2.849 1
12     3 1 -73.790 90.877 -41.158 -2.849 1
13     4 1 -76.291 90.877 -39.714 -2.849 1
.....

```

write_datafile command**Syntax:**

```
write_datafile file
```

- file = name of data file to output

Examples:

```
write_datafile data.lmp
```

Description:

Write the analyzed result into a data file in ASCII format. If no file name is provided, the program will use default one, i.e. OUTFILE

decimal_digit command**Syntax**

```
decimal_digit number
```

- number = number of decimal precise

Examples

```
decimal_digit 8
```

Description

It means that the value is in 8 decimal precise when writing a double value into output file. In C++ code, it is equivalent to use:

```
printf("%.8e",.....)
```

The default value is 9 and we use 3 in some examples to reduce the file size for easy downloading. You can comment out this line with # to improve accuracy.

write_datatype command**Syntax**

```
write_datatype fileformat
```

- fileformat= imd or lmp(lammps)

lmp/lammps: File format used by the LAMMPS molecular dynamics code.

Imd: File format used by the molecular dynamics code IMD.

Examples

```
write_datatype lmp
```

Description

The command specifies the format for the output file. No input is equivalent to use the default format, i.e. *write_datatype lmp*. (lmp is a short version in lammps)

boundary command**Syntax**

```
boundary x y z
```

- x, y, z = p or s

p is periodic

s is non-periodic

Examples

```
boundary p p p
```

```
boundary s s p
```

Description

Set the style of boundaries for the global simulation box in each dimension. A single letter assigns the same style to both the lower and upper face of the box. The style p means the box is periodic, so that atoms will interact across the boundary, and they can exit one end of the box and re-enter from the other end. The styles s means the box is non-periodic, so that atoms do not interact across the boundary and do not move from one side of the box to the other.

eliminate_cell command**Syntax**

```
eliminate_cell flag
```

- flag = on or off

Examples

```
eliminate_cell off
```

Description

The default value is on. If activated, this option lets the program first rescale the reference simulation cell including all atomic coordinates to match the cell shape in the defective configuration before calculating. This effectively eliminates any macroscopic, homogeneous deformation of the simulation cell, and the calculation will reflect only the internal, microscopic displacements of atoms.

warpin command**Syntax**

warpin flag

- flag = on or off

Examples

warpin off

Description

The default value is on. If activated, this command maps atoms that are located outside of the simulation cell back into the box by "wrapping" their coordinates around at the periodic boundaries of the simulation box. The modifier acts only in spatial directions for which periodic boundary conditions (PBC) are enabled. The PBC flags are set by *boundary* command.

test_mode command**Syntax**

test_mode

Examples

test_mode

Description

The default value is off. For NTA and DDA method, the command will modify the header information of the output file to facilitate the OVITO[1] to identify those vector if activated. Please note that this command only has effect for the output file with *imd* format. For *lmp* format, using OVITO's modifier can also visualize those vector easily and see the example for more details.

The ideal hcp structures have the configuration ... ABAB ... stacking sequence of the atomic planes. For NTA method in assigned mode, if there are hcp structured atoms, the program will analyze whether each atom locates in atomic planes A or atomic planes B. If this command activated, the program will save this analyzed result in a column named *refid*. (0 for unknown, 1 for atomic planes A and 2 for atomic planes B). If the matching results are abnormal in many atoms that far from any dislocation, you may have entered the wrong lattice type, lattice constant or orientation, especially orientation. And if this happens, you need to check these parameters further. If the matching results are abnormal in atoms near a dislocation core, this is reasonable and no need to worry

neighbor_cutoff command**Syntax**

neighbor_cutoff type radius

- type = atom type (1-Ntypes) of atoms input.
- radius = neighbor cutoff radius

Examples

```
neighbor_cutoff 1 3.7
```

Description

For STA, DDA, SVA and NTA method, the grid mesh is generated and the periodic boundary is then applied during the analysis. This command gives the cutoff radius of neighbors that are taken into account. If no value is provided, the program will determine the average bond length by analyzing the distance to the first nearest neighbor.

For STA and NTA method, the deformation gradient tensor of each atom will be computed. The cutoff controls the range of neighbors that are taken into account to compute the deformation gradient tensor of an atom. To be noted that the radius must be large enough to include all near neighbors for each atom. For face-centered cubic (FCC) and hexagonal close-packed (HCP) structures, the cutoff radius must lie midway between the first and the second shell of neighbors. For body-centered cubic (BCC) structures the cutoff radius should be placed between the second and the third neighbor shell. We recommend a value that is slightly larger than the average of the upper and lower limits of the cutoff radius.

burgers command**Syntax**

```
burgers x y z
```

- $x\ y\ z$ = Components of the Burgers vector.

Examples

```
burgers 0.0 2.0 0.0
```

Description

A screw dislocation can be considered to cut a crystal along a plane and then slip one half across the other by a Burgers vector. For STA, SVA and NTA in reference mode, one may get problem in the selection of reference lattice. If atomic ID is used to map the reference lattice and the defective one without correction, as shown by red vector in Fig. 2.1(left), large deformation will be produced. As a result, a unphysical large strain field will be generated by STA method as shown in Fig. 2.2(left).

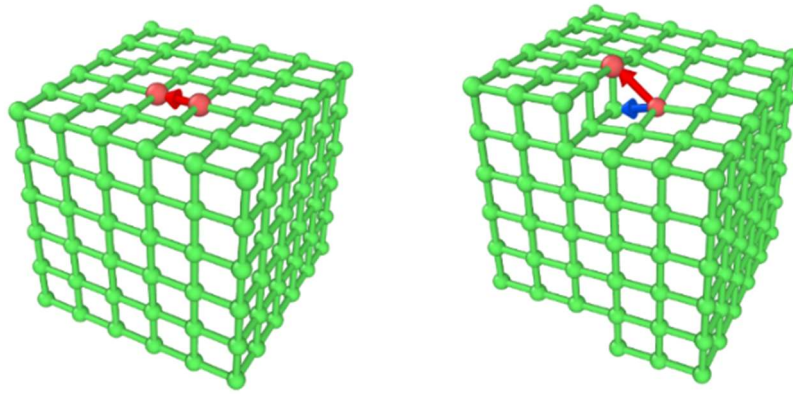


Fig. 2.1 The mapping between the reference and defective lattice according to atomic ID

If one use map with correction, as shown by blue vector in Fig. 2.1(right), the effect of total translation can be eliminated. Then the strain field caused by a screw dislocation can be easily identified with STA method, as shown by Fig. 2.2(right). To activate the correction, one need to provide the Burgers vector and then the program will correct each bond to remove the effects of total translation within slip plane. This feature is implemented by judging whether the bond length can be shortened by increasing or decreasing several Burgers vectors. The correction is not activated by default.

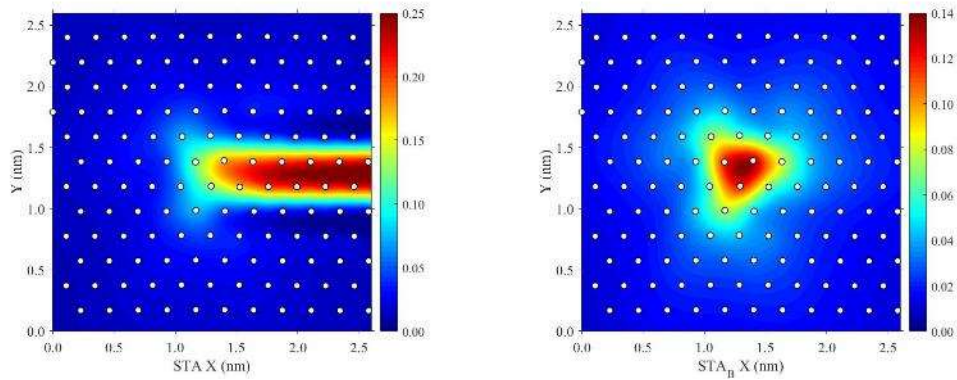


Fig. 2.2 The generated strain filed by STA method without (left) and with (right) map correction

3 Private command

These commands can be used for the setting and output of corresponding methods.

3.1 DVA

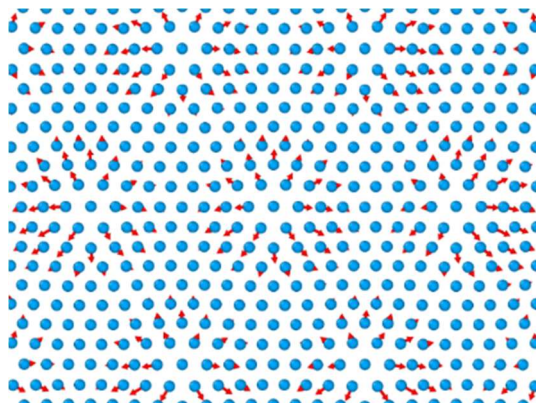


Fig. 3.1.1 The vector plot of atomic displacement revealed by DVA method

This method is used to calculate the displacement vector of each atom based on defective configuration and reference one. The defective atomic configuration is read by command *read_datafile_def*, while the reference one is read by command *read_datafile_ref*.

The displacement vector of each atom is calculated by subtracting its position in the reference file from that in the defective file. This method uses atomic ID to map the correspondence atoms from the reference file to the defective one. If an atom is not found in reference file, the atomic coordinates in defective file will be used as its reference position.

The content of each column in the output file:

column label	type	content
Displacement	Vector	Calculated displacement

Private Command

write_data_ref command

Syntax

```
write_data_ref file
```

- file = name of the second write file

Examples

```
write_data_ref res2.lmp
```

Description

If activated, the program will export the calculated atomic displacement into a data file in which the atomic positions are reproduced from reference file.

eliminate_overall command**Syntax**

eliminate_overall flag

- flag = on or off

Examples

eliminate_overall on

Description

The default value is on. If activated, each atomic displacement will subtract the average displacement of all atoms to remove the effect of translation of the system.

3.2 STA

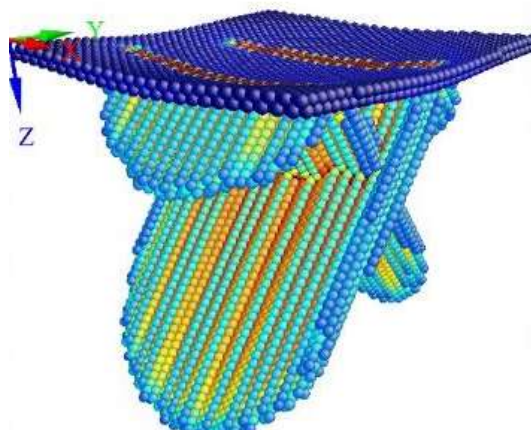


Fig. 3.2.1 The color-coded defective structure revealed by STA method

This method calculates the atomic-level strain tensors based on two atomic configuration files. The defective atomic configuration is read by command *read_datafile_def*, while the reference one is read by command *read_datafile_ref*.

This method calculates the local deformation gradient tensor \mathbf{J}_i for each atom by accounting the relative displacements of neighboring atoms within the given radial cutoff. The neighboring atoms are determined according to the defective atomic coordinates. To be noted that three non-coplanar neighboring atoms are required for the construction of deformation matrix for a central atom, otherwise the calculation makes no sense.

The atomic deformation gradient tensor \mathbf{J}_i can be decomposed into the matrix product: $\mathbf{J}_i = \mathbf{R}\mathbf{U}$, where \mathbf{R} is a rotation matrix and \mathbf{U} is the right stretch tensor. \mathbf{U} is symmetric matrix with positive values, while the local rotation matrix \mathbf{R} will be exported as atomic property termed as “Rotation”, in which contains a quaternion with four components x , y , z , and w . Accordingly, the angle of rotation can be calculated by the formula $2 \cdot \arccos(w)$ in unit of radians.

To calculate deformation gradient tensor for an atom in the defective configuration, the program needs to find the correspondence atom in the reference file. This method uses atomic ID to map the correspondence atoms from the reference file to the defective one. If an atom is not found in reference file, the atomic coordinates in defective file will be used as its reference position.

The content of each column in the output file:

column label	type	content
StrainTensor	Tensor	The deformation gradient tensors
DeformationGradient	Tensor	The deformation gradient tensors
VolumetricStrain	Value	Volumetric part of strain tensor
ShearStrain	Value	Local shear invariant
Rotation	XYZW	The local rotation
StretchTensor	Tensor	The stretch tensors

Private Command

Export command

Syntax

keyword

- *keyword* = *Deform* or *Strain* or *Stretch* *Rotation* or *Volume*.

Strain = Print the strain tensors into the output file

Deform = Print the deformation gradient tensors into the output file

Volume = Print the volumetric (hydrostatic) part of strain tensor into the output file

Rotation = Print the rotations into the output file

Stretch = Print the stretch tensors into the output file

Examples

Strain

Deform

Volume

Rotation

Stretch

Description

Activate those commands to print the following information.

- Strain: Print the six components of atomic strain tensors into output file if activated.
- Deform: Print the nine components of atomic deformation gradient tensors into output file if activated.
- Volume: Print the volumetric (hydrostatic) part of strain tensor into the output file
- Rotation: Print the local rotation obtained from the polar decomposition $F=RU$ of the atomic deformation gradient into output file if activated.
- Stretch: Print the stretch tensor U obtained from the polar decomposition $F=RU$ of the atomic deformation gradient into output file if activated. In convenience of

output, the symmetric matrix U will be converted into a six-component representation.

3.3 DDA

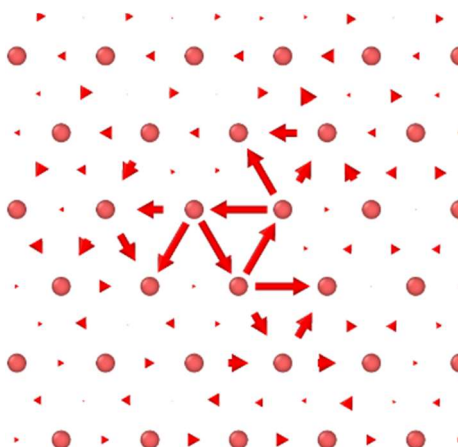


Fig. 3.3.1 The vector plot of dislocation core revealed by DDA method

For an edge dislocation, there are two ways to represent the direction of differential displacement: i) The differential displacements are scaled by corresponding distance between the two atoms and projected along the edge direction; ii) The differential displacement vector is directly projected in the plane normal to the view direction. In AADIS code, both can be switched on according to the selection of the user.

The content of each column in the output file:

column label	type	content
Edge	Vector	Edge part of the differential displacement
Screw	Vector	Screw part of the differential displacement
Mixed	Vector	Mixed part of the differential displacement

Private Command

dda_normal command

Syntax

```
dda_normal x y z
```

- x y z = Vector components normal to the projected plane.

Examples

```
dda_normal 1.0 1.0 0.0
```

Description

Provide the vector normal to the projected plane. The vector does not have to be a unit vector. The default vector is [0 1 0], and if an inappropriate vector such as [0 0 0] is

provided, the program will correct it to the default one.

dda_burgers command

Syntax

```
dda_burgers x y z
```

- x y z = Three components of the Burgers vector.

Examples

```
dda_burgers 2.5 0.0 0.0
```

Description

Provide a Burgers vector for the analysis of atomic configuration. The default value is [0.0 2.0 0.0].

dda_projection command

Syntax

```
dda_projection x y z
```

- x y z = Components of the projection vector.

Examples

```
dda_projection 2.5 0.0 0.0
```

Description

Provide the projection direction for the calculation of mixed part. The default value is [0.0 1.0 0.0]. For the mixed part, it calculates firstly the component of differential displacement **u** along the projection direction, and then those will be projected in the plane specified by *dda_normal* command.

Export command

Syntax

```
keyword
```

- keyword = EDGE or SCREW or MIXED.

EDGE	=	Print the EDGE part of the differential displacement
SCREW	=	Print the SCREW part of the differential displacement
MIXED	=	Print the MIXED part of the differential displacement

Examples

```
EDGE
SCREW
MIXED
```

Description

Use this command to activate the program to print the corresponding information. The SCREW part of the differential displacement will always be printed by default.

edge_bond command**Syntax**

edge_bond

Examples

edge_bond

Description

For an edge dislocation, there are two ways to represent the direction of differential displacement: i) The differential displacements are scaled by corresponding distance between the two atoms and projected along the edge direction; ii) The differential displacement vector is projected in the plane normal to the view direction. In AADIS code, both can be switched on according to the selection of the user. If *edge_bond* command is activated, the former way will be adopted.

center command**Syntax**

center flag

- flag = def or ref

Examples

center def

Description

The differential displacements are projected according to the relative position of each atomic pair. When *def* mode is used, the program will print the positions of arrows according to the defective atomic position. If *ref* mode is used, the program will use reference coordinates instead. The default choice is *ref* mode.

3.4 SVA

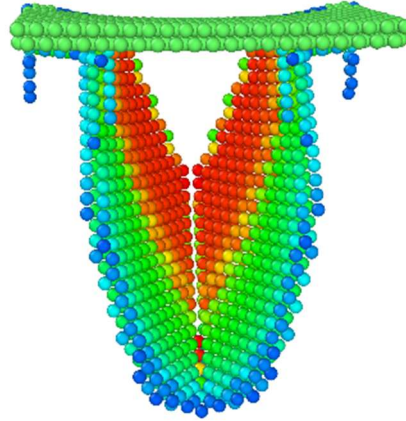


Fig. 3.4.1 The vector plot of dislocation core revealed by SVA method

The slip vector parameter ξ_i is defined as

$$\xi_i = -\frac{1}{N_s} \sum_{j \neq i}^N (R_{ij}^{def} - R_{ij}^{ref})$$

To calculate the slip vector of each atom in the defective configuration, the method needs to find the corresponding atom in the reference configuration. The method use atomic ID to map the correspondence atoms from the reference file to the defective one. If an atom is not found in reference file, the atomic coordinates in defective file will be used as its reference position. Normally, AADIS will output the ξ_i vector, the normal

of ξ_i , and ξ_{ave} that is defined as:

$$\xi_{ave} = -\frac{1}{N_s} \sum_{j \neq i}^N (|R_{ij}^{def} - R_{ij}^{ref}|)$$

The content of each column in the output file:

column label	type	content
SVA	Vector	The slip vector
Average	Value	The average of the slip
SVA	Value	The normal of the slip vector
MAXSVA	Vector	The maximum slip vector
MAXSVA	Value	The normal of the maximum slip vector

Private Command

max_sva command

Syntax

<i>max_sva</i>

Examples

<i>max_sva</i>

Description

This command is used to print the largest differential displacement of all atomic pairs around the center atom, which can be defined as:

$$\xi_{\max} = -\max(R_{ij}^{def} - R_{ij}^{ref}) \quad .$$

Where R_{ij}^{def} and R_{ij}^{ref} are the vector differences of atoms i and its neighbor j in the defective and reference configurations, respectively. It should be noted that there is a certain similarity between ξ_{\max} and the IDA method in some sense. An additional column named *MAXSVA* will be added after the analysis.

3.5 IDA

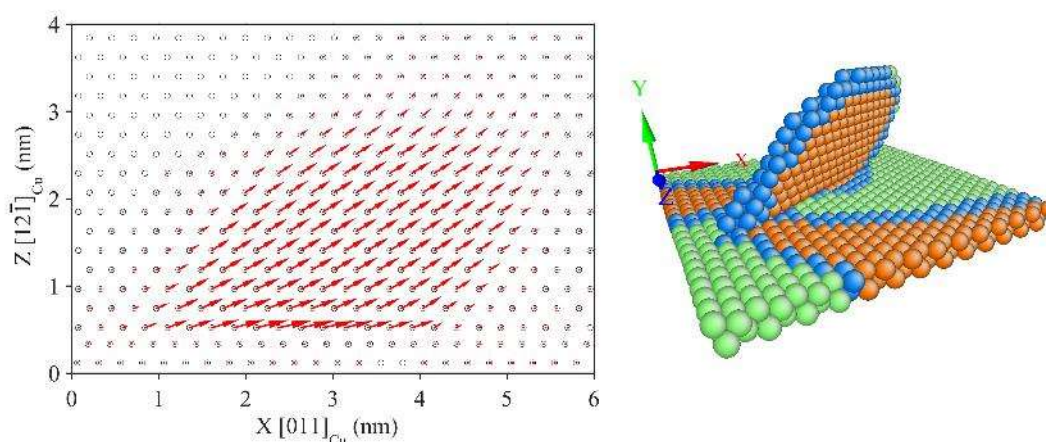


Fig. 3.5.1 The vector plot of dislocation core revealed by IDA method

Atomic pairs will be constructed between the atoms in the first layer and their nearest neighbor in the second layer. Afterwards, this method will calculate the difference of relative displacement of atomic pairs, i.e. disregistry, between defective configuration and reference one.

This method needs to find the correspondence atoms in the defective and reference file according to atomic ID. If an atom is not found in reference file, the atomic coordinates in defective file will be used as its reference position.

The content of each column in the output file:

column label	type	content
Disregistry	Vector	The difference of relative displacement

Private Command

norm command

Syntax

```
norm x y z
```

- x y z = Components of vector normal to projected plane.

Examples

```
norm -0.28 0.94 0.19
```

Description

Provide the vector normal to the projected plane. The vector does not have to be a unit vector. The default vector is [1 0 0], and if an inappropriate vector such as [0 0 0] is provided, the program will correct it to the default one.

layer1 command**Syntax**

layer1 distance half-thickness type

- distance = the (assigned) distance between layer plane from the coordinate system's origin along the direction of the normal vector.
- half-thickness = specifies the half width of the layer to cut from the atomic set.
- type = number of atomic types (1-Ntypes). Using -1 means that the atom type is not distinguished, the atomic type will be ignored during selection.

Examples

layer1 1.2 0.4 1

Description

This command chooses atoms located in the layer I as the first layer. All atoms located in the slice with a certain thickness are selected. Please note that the slice is built based on the reference coordinates since those atoms in the defective configuration may be highly distorted from equilibrium.

layer2 command**Syntax**

layer2 distance half-thickness type

- distance = the (assigned) distance between the layer plane from the coordinate system's origin along the direction of the normal vector.
- half-thickness = specifies the half width of the layer to cut from the atomic set.
- type = number of atom type (1-Ntypes). Using -1 means that the atom type is not distinguished, the atomic type will be ignored during selection.

Examples

layer2 -1.2 0.4 2

Description

This command chooses atoms located in the layer II as the first layer. All atoms located in the slice with a certain thickness are selected. Please note that the slice is built based on the reference coordinates since those atoms in the defective configuration may be highly distorted from equilibrium.

eliminate_overall command**Syntax**

eliminate_overall flag

- flag = on or off

Examples

<i>eliminate_overall on</i>

Description

The default value is on. If activated, this command will subtract the overall average displacement from the average displacement of all atoms to reflect local features.

Note:

Parameter file for DVA analysis must contain the following command at least:

norm
layer1
layer2

3.6 NTA(assigned mode)

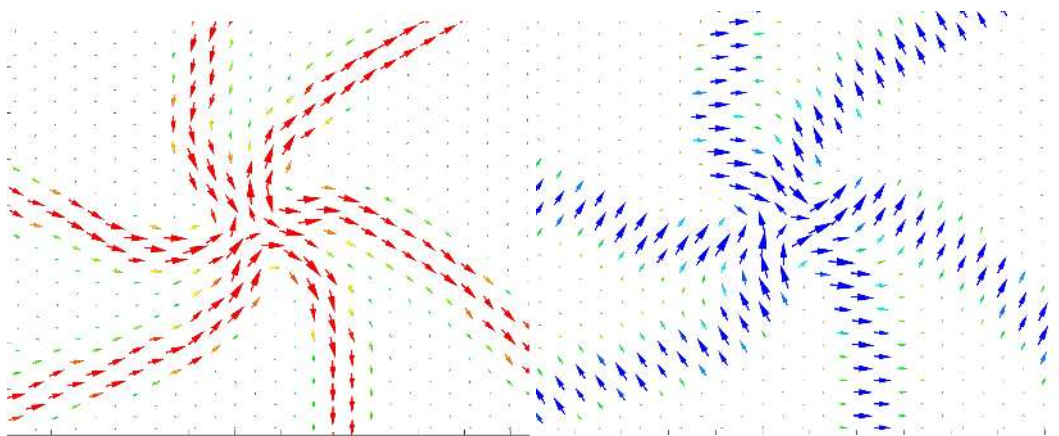


Fig. 3.6.1 The vector plot of misfit dislocation revealed by NTA method in assigned mode

The analyzed results by NTA method depend strongly on the scheme to map the neighboring bonds. So far no universal and efficient mapping scheme is accepted in public. As regards the choice of cutoff, different values are provided for different systems in previous publications. However, a valuable suggestion on appropriate angle can be obtained from the comparison with the distribution of excess potential energy around dislocations, or the analyzed results from CNA or DXA methods.

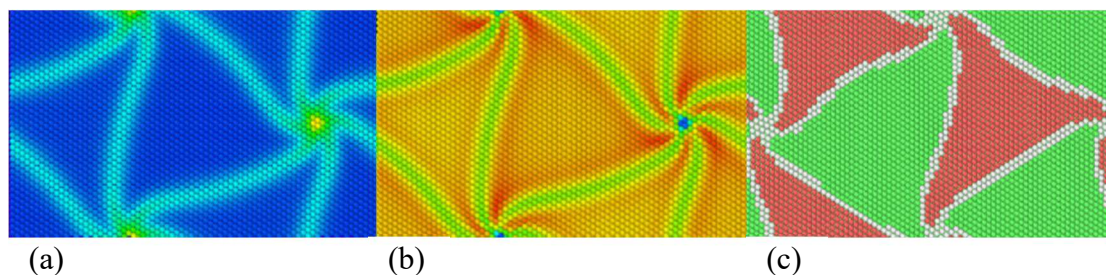


Fig. 3.6.2 Atomic structure at Cu-Ni interface with atomic color coded by (a) excess potential energy, (b) atomic stress and (c) CNA method

The content of each column in the output file:

column label	type	content
s1,s2,s3	Value	The three singular values of the Nye tensor
Line	Vector	Line direction (Left singular vector)
Burgers	Vector	Burgers vector (Right singular vector)

n	Value	Atomic coordination number
refid	Value	The stacking sequence

The content of each column in the other output files:

column label	type	content
D	Tensor	The deformation tensor
s1,s2,s3	Value	The three singular values of the deformation tensor
U	Vector	Left singular vector of the deformation tensor
V	Vector	Right singular vector of the deformation tensor
N	Tensor	The Nye tensor

Private Command

lattice_orient command

Syntax

```
lattice_orient type
xi xj xk
yi yj yk
zi zj zk• type = Atomic type (1-Ntypes).
```

- xi xj xk = Crystallographic direction along x axis..
- yi yj yk = Crystallographic direction along y axis..
- zi zj zk = Crystallographic direction along z axis..

Examples

```
lattice_orient 1
1 -1 0
1 1 -2
1 1 1
```

Description

Use this command to define crystallographic orientation of a certain atomic type. If the orientation is not defined, the program will take the default values: [100] [010] [001]. Fig. 3.6.3 presents the default crystallographic orientation for bcc, fcc and hcp structure:

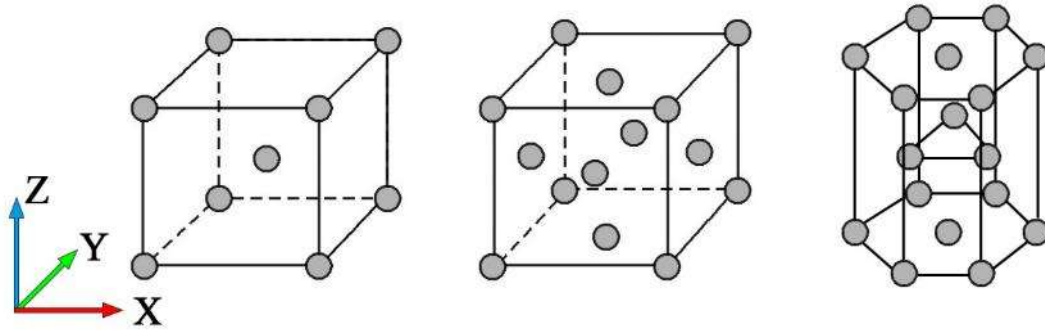


Fig. 3.6.3 The default crystallographic orientation of bcc, fcc and hcp structure

To be noted that for bcc structured atoms, the first and second neighbors will be used for the calculation of local lattice deformation, while for fcc and hcp structured atoms, only the first neighbors will be used for such calculations. Fig. 3.6.4 illustrates the coordination environment of an atom in bcc, fcc and hcp structure with default orientation.

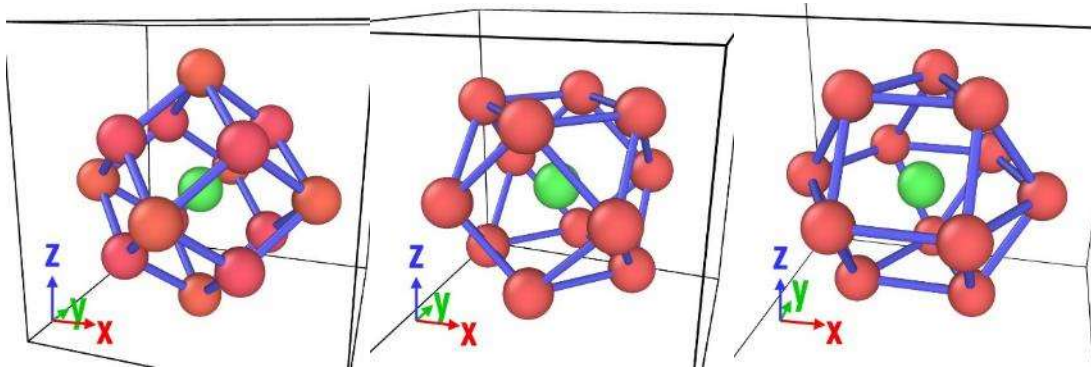


Fig. 3.6.4 The coordination environment of an atom in bcc, fcc and hcp structure with default orientation. Note that the coordination environment of bcc, fcc and hcp structure are provided in the appendix.

Math related:

After providing the crystallographic orientation vector, it will be normalized accordingly.

$$\mathbf{Ori} = \begin{pmatrix} x_i & x_j & x_k \\ y_i & y_j & y_k \\ z_i & z_j & z_k \end{pmatrix} \rightarrow \mathbf{iRot} = \begin{pmatrix} nx_i & nx_j & nx_k \\ ny_i & ny_j & ny_k \\ nz_i & nz_j & nz_k \end{pmatrix}$$

where \mathbf{iRot} is an orthogonal matrix and the inverse matrix of \mathbf{iRot} can be calculated as:

$$\mathbf{Rot} = \mathbf{iRot}^{-1} = \begin{pmatrix} nx_i & ny_i & nz_i \\ nx_j & ny_j & nz_j \\ nx_k & ny_k & nz_k \end{pmatrix}$$

Where **Rot** denotes the rotation matrix and accordingly each atom with the defined orientation in the reference lattice can be obtained by the following formula:

$$R_{ij}^{ref'} = R_{ij}^{ref} \cdot \mathbf{Rot}$$

lattice_type command

Syntax

<i>lattice_type type style</i>

- type = Atomic type (1-Ntypes).
- style = bcc or fcc or hcp or others

Examples

<i>lattice_type 2 hcp</i>

Description

Input a lattice type for each atom, e.g. fcc, bcc, hcp and others. If the lattice type is not provided, the program will take the default one, i.e. fcc. If no hcp type is assigned for any atoms, the program may distinguish fcc and bcc automatically. If the atomic type identified by the program automatically does not meet the one specified by command *lattice_type*, the program will adopt the type specified. To be noted additionally that the atoms defined as others will be excluded from deformation calculations, and their deformation tensor will be set as identity matrix, indicating no deformation near these atoms

lattice_const command

Syntax

<i>lattice_const type value</i>

- type = atomic type (1-Ntypes).
- value = lattice constant

Examples

<i>lattice_const 2 3.232</i>

Description

The program will analyze the average bond length to get the lattice constant automatically. If the calculated lattice constant by the program automatically does not meet the specified one, the program will adopt the input one for further analysis.

mapset command

Syntax

<i>mapset mode value</i>

- mode = angle or distance

- value = cutoff

Examples

mapset angle 15

Description

For angle mode, this command defines an angle cutoff ϕ_{\max} , beyond which the bonds will be excluded from the calculations of deformation gradient tensor.

For distance mode, this command defines a distance cutoff, beyond which the bonds will be excluded from the calculations of deformation gradient tensor.

$$|R_{ij}^{def} - R_{ij}^{ref}| < d_{cutoff}.$$

Where R_{ij}^{def} and R_{ij}^{ref} are the vector differences of atoms i and its neighbor j in the defective and reference configurations, respectively.

The default setting is *mapset angle 25*.

export command

Syntax

keyword

- keyword = deformtensor or deformsvd or nye_tensors or program_infor or mapnum

deformtensor = print the deformation tensors. (Write into a new file with *_deform* suffix)

deformsvd = print the deformation tensors with singular value decomposition. (Write into a new file with *_deform* suffix)

nye_tensors = print the Nye tensors. (Write into a new file)

program_infor = print some generated information by program, such as lattice type, lattice constant. By checking the file one can identify whether the analysis run correctly.

mapnum = Atomic coordination number after mapping

Examples

deformtensor
deformsvd
nye_tensors
program_infor
mapnum

Description

Use this command to activate the program to print the following information.

- deformtensor: Print the deformation gradient tensors into a new file if activated.
- deformsvd: Print the deformation gradient tensors with singular value decomposition into a new file if activated.
- nye_tensors: Print the Nye tensor into a new file if activated.

- `program_infor`: Print the some additional information into a new file during running program if activated.
- `mapnum`: Print the atomic coordination number after mapping if activated. Normally, coordination numbers of an atom around a dislocation core will less than 12 for fcc and hcp structure or 14 for bcc structure. Coordination number of an atom in an ideal region is equal to 12 for fcc and hcp structure or 14 for bcc structure. Otherwise, it indicates that a wrong lattice type, or lattice constant or orientation is provided.

Note:

Command file of NYE must contain the following command at least:

`lattice_orient`

3.7 NTA (reference mode)

This method uses Nye tensor to analyze the dislocation structure by comparing the atomic coordinates in reference and defective files. The functional needs to find the correspondence atoms in the defective and reference file according to atomic ID. If an atom is not found in reference file, the atomic coordinates in defective file will be used as its reference position.

The content of each column in the output file:

column label	type	content
s1,s2,s3	Value	The three singular values of the Nye tensor
Line	Vector	Line direction (Left singular vector)
Burgers	Vector	Burgers vector (Right singular vector)
n	Value	Atomic coordination number

The content of each column in the other output files:

column label	type	content
D	Tensor	The deformation tensor
s1,s2,s3	Value	The three singular values of the deformation tensor
U	Vector	Left singular vector of the deformation tensor
V	Vector	Right singular vector of the deformation tensor
N	Tensor	The Nye tensor

Private Command

mapset command

Syntax

```
mapset mode value
```

- mode = angle or distance
- value = cutoff

Examples

```
mapset angle 15
```

Description

For angle mode, this command defines an angle cutoff ϕ_{\max} , beyond which the bonds will be excluded from the calculations of deformation gradient tensor.

For distance mode, this command defines a distance cutoff, beyond which the bonds will be excluded from the calculations of deformation gradient tensor.

$$\left| R_{ij}^{def} - R_{ij}^{ref} \right| < d_{cutoff} \quad .$$

Where R_{ij}^{def} and R_{ij}^{ref} are the vector differences of atoms i and its neighbor j in the defective and reference configurations, respectively.

The default setting is *mapset angle 25*.

export command

Syntax

keyword

- keyword = deformtensor or deformsvd or nye_tensors or mapnum

deformtensor = print the deformation tensors with singular value decomposition.
(Write into a new file with *_deform* suffix)

deformsvd = print the deformation tensors with singular value decomposition.
(Write into a new file with *_deform* suffix)

nye_tensors = print the Nye tensors. (Write into a new file)

mapnum = Atomic coordination number after mapping

Examples

deformtensor

deformsvd

nye_tensors

mapnum

Description

Use this command to activate the program to print the following information.

- deformtensor: Print the deformation gradient tensors into a new file if activated.
- deformsvd: Print the deformation gradient tensors with singular value decomposition into a new file if activated.
- nye_tensors: Print the Nye tensor into a new file if activated.
- mapnum: Print the atomic coordination number after mapping if activated. Normally, coordination numbers of an atom around a dislocation core will less than 12 for fcc and hcp structure or 14 for bcc structure. Coordination number of an atom in an ideal region is equal to 12 for fcc and hcp structure or 14 for bcc structure. Otherwise, it indicates that a wrong lattice type, or lattice constant or orientation is provided.

Appendices

A.1 Examples

The AADIS includes a sub-directory in which includes several examples. Each example contains a parameter script file (INPAR), two configuration files (DISFILE and REFFILE), and the critical output file that contains the analyzed results.

The output file can be viewed directly by various visualization programs such as Ovito software [\[1\]](#). One can open the Ovito project file (*.ovito) directly for visualization and the angle of view has been adjusted to show the most details. Some details of parameter script file (INPAR) are given in the following sections. The examples are done by the nanoindentation simulation of Au single crystal and the tension simulation of Cu/Ni bilayer models.

A.1.1 Nanoindentation simulations in fcc-Au

Multiple analysis have been performed on the nanoindentation simulations of Au single crystal along $\langle 111 \rangle$ direction after the appearance of plastic deformation by using the EAM potentials developed by Mendelev et al [2].

A.1.1.1 NYE (NTA method in assigned mode)

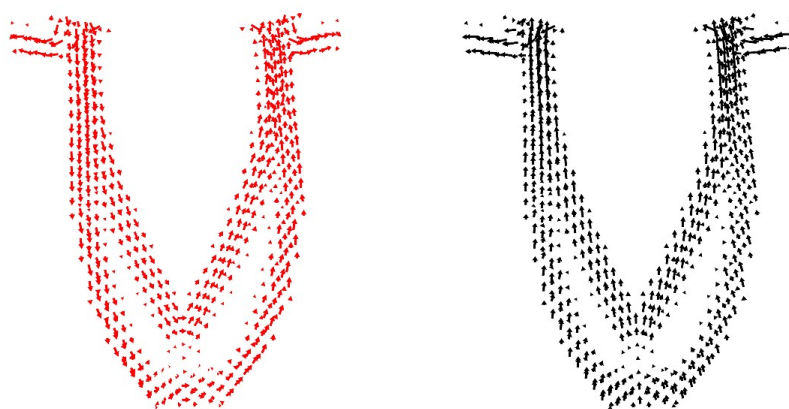


Fig. A.1.1.1 Line sense (left) and Burgers vector (right) of extended dislocations

For NYE analysis, the INPAR looks like:

```
#nye
#to activate the NTA analysis in assigned mode
lattice_orient 1
1 1 -2
-1 1 0
1 1 1
# Define the crystallographic orientation for the type 1 atoms

lattice_type 1 fcc
#Define the lattice type for the type 1 atoms
lattice_const 1 4.078
#Define the lattice parameter for the type 1 atoms
neighbor_cutoff 1 3.48
#Define the cutoff radius of neighbors that are taken into account for the type 1 atoms

boundary s s s
#Set the style of boundaries are non-periodic for the global simulation box in x, y and
z dimension.
```

```

write_datafile NYE.15.lmp
#Give the path of the output file containing analyzed result
read_datafile_def ../DISFILE
# Give the path of the data file containing information needed to do an analysis
mapset angle 15
#Define the cutoff with angle mode
mapnum
#Output the number of coordination

```

A.1.1.2 NYE (NTA analysis with CNA correction)

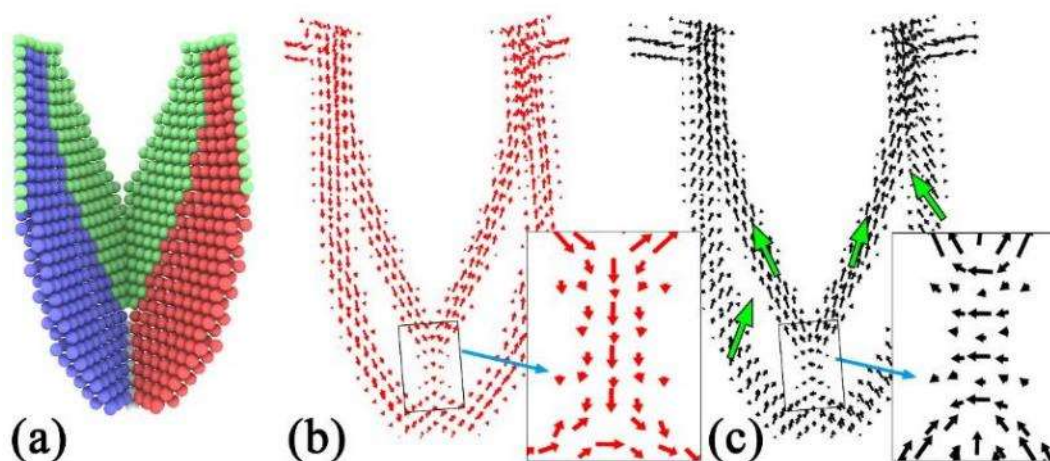


Fig. A.1.1.2 Dislocation structure (a) color coded by CNA method, (b) denoted by line sense and (c) Burgers vector

With CNA correction, one may get an appropriate choice of cutoff, and then the sessile Lomer-cottrell lock with Burgers vector of $\langle 110 \rangle / 6$ can be clearly identified and shown in Fig. A.1.1.2b-c. Please compare with the analyzed results shown in Fig. A.1.1.1 without CNA correction.

For NTA analysis with CNA correction, the INPAR looks like:

```

#nye
#to activate the NTA analysis in assigned mode
lattice_orient 1
1 1 -2
-1 1 0
1 1 1
# Define the crystallographic orientation for the type 1 atoms
lattice_orient 4
1 1 -2

```

```

-1 1 0
1 1 1
# Define the crystallographic orientation for the type 4 atoms
lattice_orient 2
    0.866025403784439    -0.166666666666667    -0.471404520791032
    0.500000000000000    0.288675134594813    0.816496580927726
                0    -0.942809041582063    0.333333333333333
# Define the crystallographic orientation for the type 2 atoms
lattice_orient 3
    0.866025403784439    -0.166666666666667    0.471404520791032
    -0.500000000000000    -0.288675134594813    0.816496580927726
                0    -0.942809041582063    -0.333333333333333
# Define the crystallographic orientation for the type 3 atoms

lattice_type 1 fcc
lattice_type 4 others
lattice_type 2 hcp
lattice_type 3 hcp
#Define the lattice types for all type atoms

lattice_const 1 4.078
lattice_const 4 4.078
lattice_const 2 2.883581454
lattice_const 3 2.883581454
#Define the lattice parameters for all type atoms

neighbor_cutoff 1 3.48
neighbor_cutoff 4 3.48
neighbor_cutoff 2 3.48
neighbor_cutoff 3 3.48
#Define the cutoff radius of neighbors that are taken into account for all type atoms

boundary s s s
#Set the style of boundaries are non-periodic for the global simulation box in x, y and
z dimension.
program_infor
#export lattice information to check
write_datafile NYE.15.lmp
#Give the path of the output file containing analyzed result
read_datafile_def ../DISFILE_c
#Give the path of the data file containing information needed to do an analysis
mapset angle 15

```

```
#Define the cutoff with angle mode
test_mode
#Add for Outputting the AB atomic planes checking results
mapnum
#Output the number of coordination
```

It is easy to specify the orientation for an atom with the fcc and bcc structure, but not direct for an atom with hcp structure. A convenient way to specify the orientation for an atom with the hcp structure is to directly input the rotation matrix. For example, one may easily specify the orientation for an fcc atom as:

$$\begin{pmatrix} 1 & 1 & -2 \\ -1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

The rotation matrix for type 3 atom with hcp structure can be obtained by the following method (See Fig. A.1.1.3 for more details.):

1. The x direction of the crystal lattice \mathbf{l}_x can be easily obtained:

$$\mathbf{l}_x = \left(\frac{\sqrt{3}}{2}, -\frac{1}{2}, 0 \right)$$

2. The z direction of the crystal lattice \mathbf{l}_z is the normal of the slip plane which can be obtained via Ovito slice modifier:

$$\mathbf{l}_z = \left(\frac{\sqrt{2}}{3}, \sqrt{\frac{2}{3}}, -\frac{1}{3} \right)$$

3. The y direction of the crystal lattice \mathbf{l}_y can be obtained with the vector product of \mathbf{l}_x and \mathbf{l}_z :

$$\mathbf{l}_y = -\mathbf{l}_x \times \mathbf{l}_z = \left(-\frac{1}{6}, -\frac{\sqrt{3}}{6}, -\frac{2\sqrt{3}}{3} \right)$$

4. Note that the rotation matrix needs to be transposed:

$$\begin{pmatrix} \frac{\sqrt{3}}{2} & -\frac{1}{2} & 0 \\ -\frac{1}{6} & -\frac{\sqrt{3}}{6} & -\frac{2\sqrt{3}}{3} \\ \frac{\sqrt{2}}{3} & \sqrt{\frac{2}{3}} & -\frac{1}{3} \end{pmatrix} \xrightarrow{\text{Transpose}} \begin{pmatrix} 0.8660 & -0.1667 & 0.4714 \\ -0.5000 & -0.2887 & 0.8165 \\ 0 & 0.9428 & -0.3333 \end{pmatrix}$$

To be noted that the rotation matrix for type 2 atom can also be obtained in the same way.

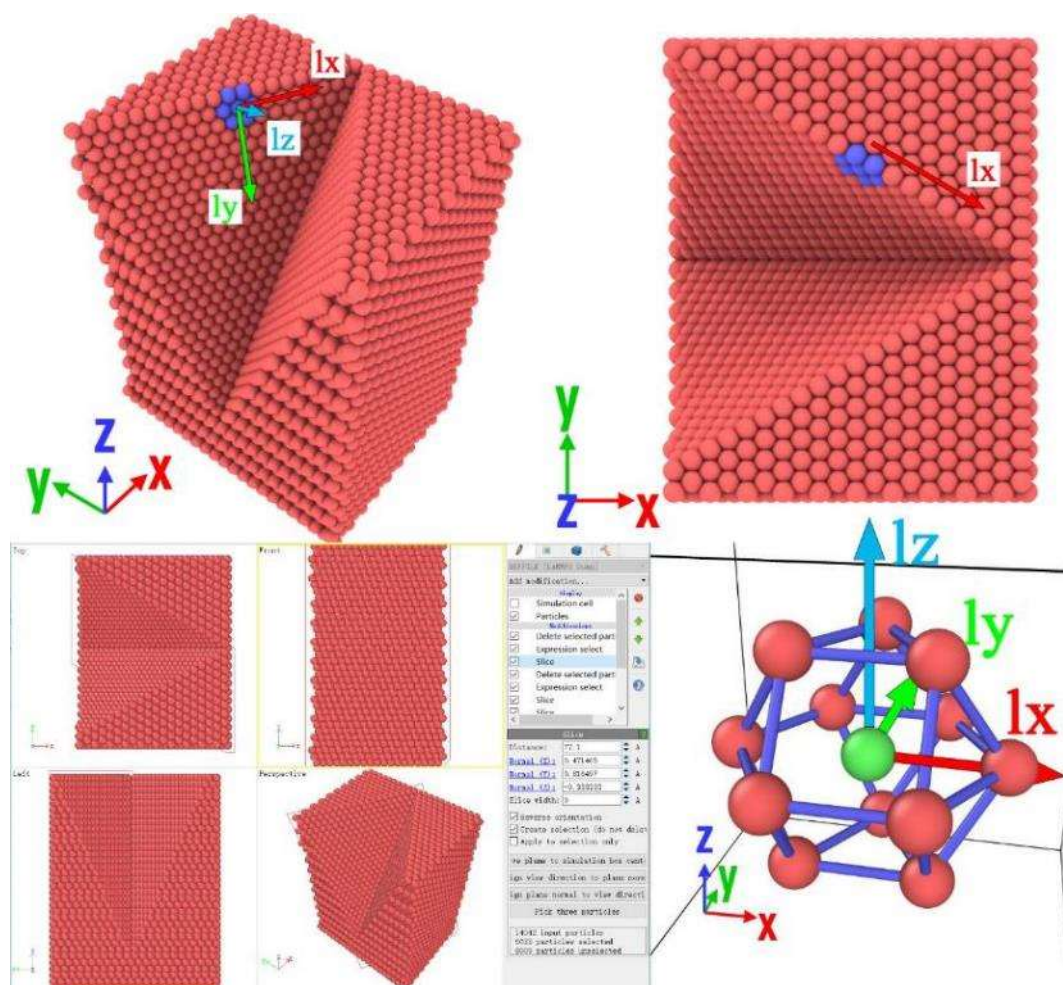


Fig. A.1.1.3 The coordination environments of hcp atom

After analysis, there are two ways to check whether the input orientation is appropriate or not.

1. To check the coordination numbers. For an ideal crystal in hcp or fcc structure, the coordination numbers should be equal to 12, whereas the defective atom would have coordination number less than 12. As revealed in Fig. A.1.1.4a, the atom near dislocation core has coordination number less than 12. For a better visualization the atoms with 12 neighbors are removed.
2. To check the stacking sequence. Fig. A.1.1.4b presents the two layers of hcp atoms that can be well divided into A and B stacking sequence.

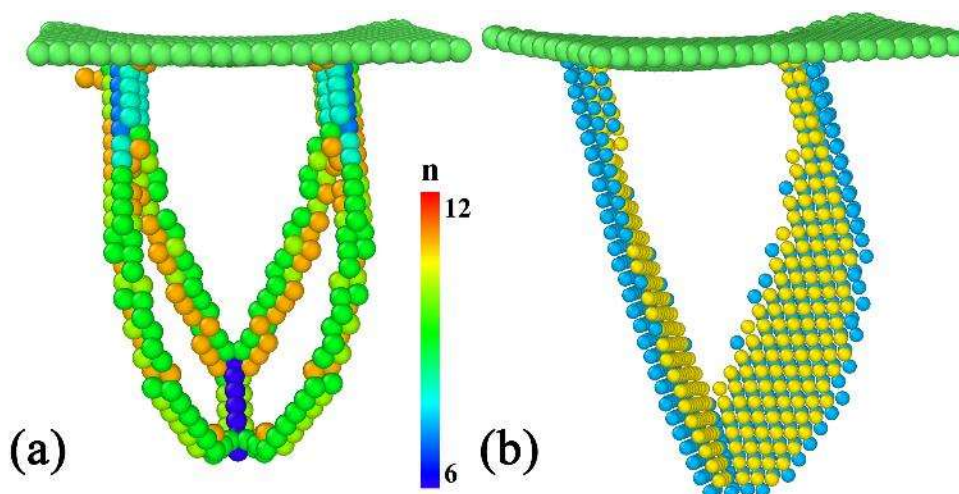


Fig. A.1.1.4 The representation of (a) coordination numbers and (b) AB stacking sequence.

If one exchanges the orientation between type 2 and type 3 atoms as illustrated below, one may get problem in analysis (see Fig. A.1.1.5, for a better visualization the atoms with 12 neighbors are removed).

```
lattice_orient 2
  0.866025403784439   -0.166666666666667   0.471404520791032
 -0.500000000000000   -0.288675134594813   0.816496580927726
          0           -0.942809041582063   -0.333333333333333
# Define the crystallographic orientation for the type 2 atoms
lattice_orient 3
  0.866025403784439   -0.166666666666667   -0.471404520791032
  0.500000000000000   0.288675134594813   0.816496580927726
          0           -0.942809041582063   0.333333333333333
# Define the crystallographic orientation for the type 3 atoms
```

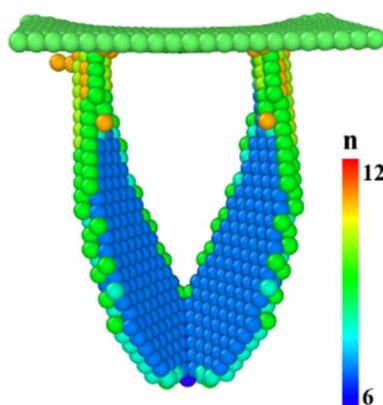


Fig. A.1.1.5 The representation of incorrect coordination numbers by using inappropriate orientation

A.1.1.3 STA (STA analysis)

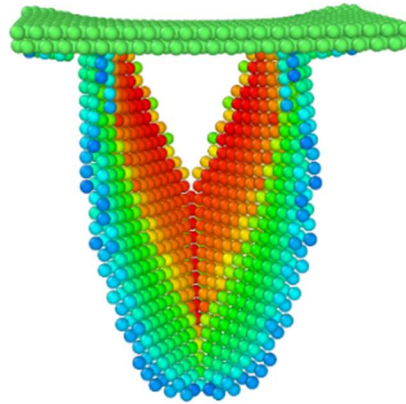


Fig. A.1.1.6 The dislocation structure of Au crystal revealed by STA analysis under nanoindentation

For STA analysis, the INPAR looks like:

```
#sta
#to activate the STA analysis in assigned mode
boundary s s s
#Set the style of boundaries are non-periodic for the global simulation box in x, y and
z dimension.
read_datafile_def ../DISFILE
#Give the path of the data file containing information needed to do an analysis
read_datafile_ref ../REFFILE
#Give the path of the data file containing reference configuration
set neigh_lenth_stan 1 3.48
#Define the cutoff radius of neighbors that are taken into account
write_datafile STA.lmp
#Give the path of the output file containing analyzed result
```

A.1.1.4 SVA (SVA analysis)

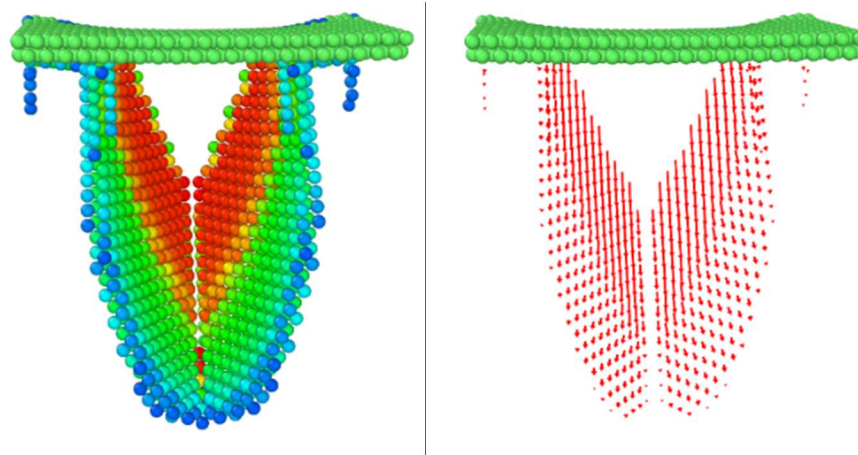


Fig. A.1.1.7 The dislocation structure of Au crystal revealed by SVA analysis under nanoindentation

For SVA analysis, the INPAR looks like:

```
#sva
#to activate the SVA analysis in assigned mode
boundary s s s
#Set the style of boundaries are non-periodic for the global simulation box in x, y and
z dimension.
read_datafile_def ../DISFILE
#Give the path of the data file containing information needed to do an analysis
read_datafile_ref ../REFFILE
#Give the path of the data file containing reference configuration
set neigh_lenth_stan 1 3.48
#Define the cutoff radius of neighbors that are taken into account
write_datafile SVA.lmp
#Give the path of the output file containing analyzed result
```

A.1.2 Cu-Ni interface under uniaxial tension

Multiple analysis have been performed to reveal dislocation nucleation inside Cu side from Cu{111}://{111}Ni semicoherent interface under uniaxial tension loading along $\langle 112 \rangle$ direction. The Cu–Ni interatomic potential developed by Bonny et al [3] is adopted in the present examples.

A.1.2.1 IDA

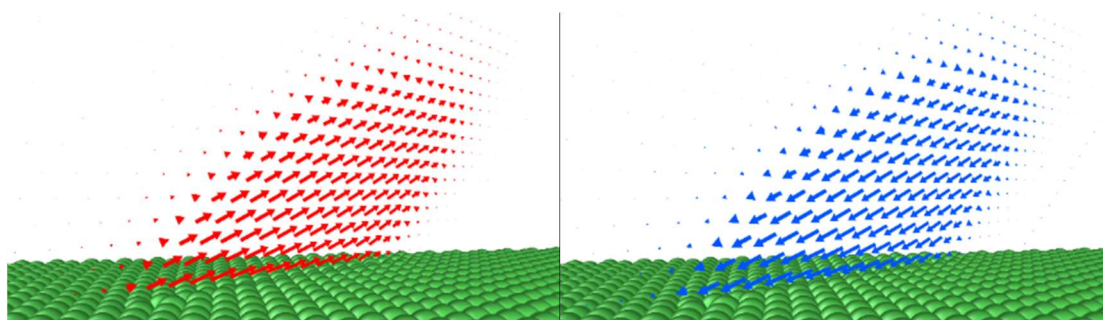


Fig. A.1.2.1 The dislocation nucleation at Cu{111}://{111}Ni interface revealed by IDA method

For IDA analysis, the INPAR looks like:

```
#ida
#to activate the IDA analysis in assigned mode
eliminate_cell
# subtract the overall average displacement from the displacement of all atoms which
can better reflect local features
normal_vector 0.471517 -0.32358 0.820346
#Give the components of the layer plane's normal vector
layer_one -29.5035 1 -1
#Give layer information to select atoms located in the layer I. The last -1 means that
the atom type is not distinguished
layer_two -27.3035 1 -1
#Give layer information to select atoms located in the layer II. The last -1 means that
the atom type is not distinguished

read_datafile_def ../CuNi_model/CuNi_x.lmp
#Give the path of the data file containing information needed to do an analysis
read_datafile_ref ../CuNi_model/CuNi_m.lmp
#Give the path of the data file containing reference configuration
write_datafile CuNi.IDA.lmp
#Give the path of the output file containing analyzed result
```


A.1.2.2 NYE

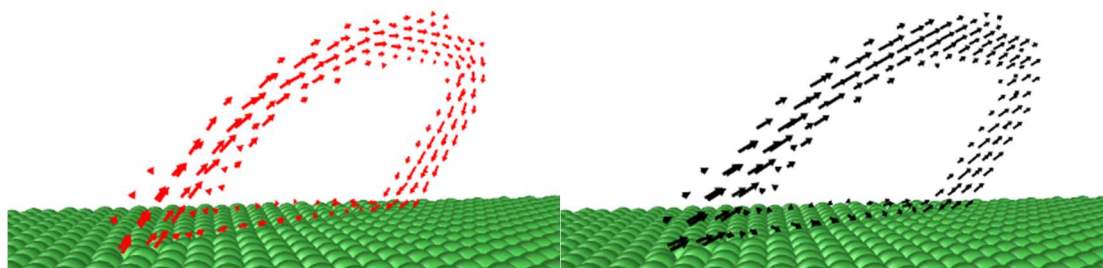


Fig. A.1.2.2 The dislocation nucleation at Cu{111}/{111}Ni interface revealed by NTA method: Line sense representation (left) and Burgers vector representation (right)

For NTA analysis in assigned mode, the INPAR looks like:

```
#nye
#to activate the NTA analysis in assigned mode
lattice_orient 1
1 1 2
1 1 -1
-1 1 0
# Define the crystallographic orientation for the type 1 atoms

lattice_orient 2
1 1 2
1 1 -1
-1 1 0
# Define the crystallographic orientation for the type 2 atoms

lattice_type 1 fcc
lattice_type 2 fcc
#Define the lattice types for all type atoms
lattice_const 1 3.615
lattice_const 2 3.52
#Define the lattice parameters for all type atoms
neighbor_cutoff 1 3.21
neighbor_cutoff 2 3.14
#Define the cutoff radius of neighbors that are taken into account for all type atoms

boundary p s p
#Set the style of boundaries are non-periodic for the global simulation box in y and
periodic in x and z dimension.
```

```

program_infor
#export lattice information to check
mapset angle 15
#Define the cutoff with angle mode
write_datafile CuNi.15.imd
#Give the path of the output file containing analyzed result
read_datafile_def ../../CuNi_model/CuNi_x.lmp
#Give the path of the data file containing information needed to do an analysis

decimal_digit 3
#reduce the file size for easy downloading
write_datatype imd
#specifies the format of output file as imd.
mapnum
#Output the number of coordination
test_mode
#Add this command for more convenient identifying the vector by Ovito with the imd
format

```

A.1.2.3 NYE(Streamlined version)

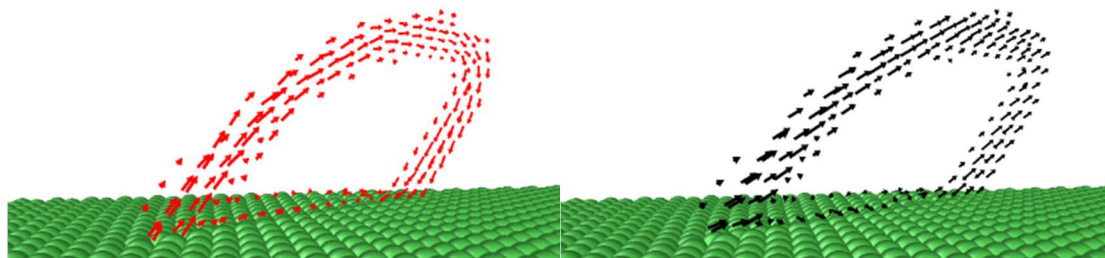


Fig. A.1.2.3 The dislocation nucleation at Cu{111}/{111}Ni interface revealed by NTA method (streamlined version): Line sense representation (left) and Burgers vector representation (right)

For NTA analysis in assigned mode by streamlined version, the INPAR looks like:

```

#nye
lattice_orient 1
1 1 2
1 1 -1
-1 1 0
# Define the crystallographic orientation for the type 1 atoms

```

```

lattice_orient 2
1 1 2
1 1 -1
-1 1 0
# Define the crystallographic orientation for the type 2 atoms

boundary p s p
#Set the style of boundaries are non-periodic for the global simulation box in y and
periodic in x and z dimension.

program_infor
#export lattice information to check

mapset angle 15
#Define the cutoff with angle mode
write_datafile CuNi.15.lmp
#Give the path of the output file containing analyzed result
read_datafile_def ../../CuNi_model/CuNi_x.lmp
#Give the path of the data file containing information needed to do an analysis

decimal_digit 3
#reduce the file size for easy downloading
mapnum
#Output the number of coordination

```

A.1.2.4 STA

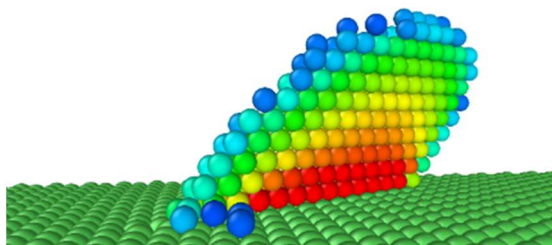


Fig. A.1.2.4 The dislocation nucleation at Cu{111}/{111}Ni interface revealed by STA method.

For STA analysis, the INPAR looks like:

```

#sta
#to activate the STA analysis in assigned mode

```



```

boundary p s p
#Set the style of boundaries are non-periodic for the global simulation box in y and
periodic in x and z dimension.
read_datafile_def ../../CuNi_model/CuNi_x.lmp
#Give the path of the data file containing information needed to do an analysis
read_datafile_ref ../../CuNi_model/CuNi_r.lmp
#Give the path of the data file containing reference configuration
set neigh_lenth_stan 1 3.16
set neigh_lenth_stan 2 3.16
#Define the cutoff radius of neighbors that are taken into account for 1 and 2 type
atoms
write_datafile CuNi.STA.lmp
#Give the path of the output file containing analyzed result

```

A.1.2.5 SVA

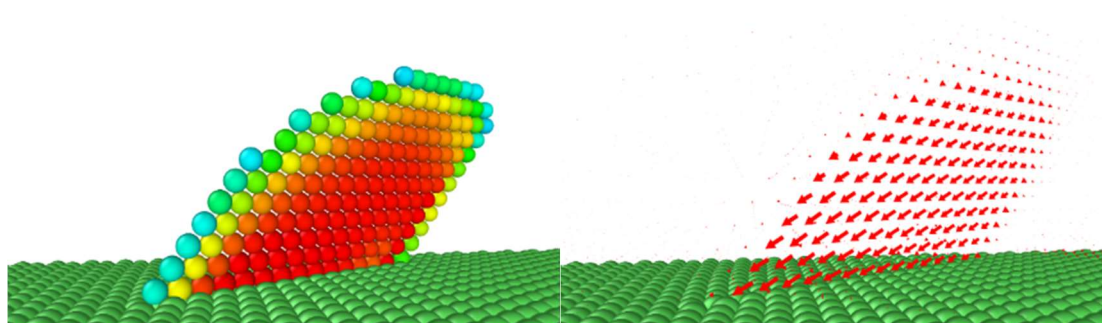


Fig. A.1.2.5 The dislocation nucleation at Cu{111}/{111}Ni interface revealed by SVA method.

For SVA analysis, the INPAR looks like:

```

#sva
boundary p s p
#Set the style of boundaries are non-periodic for the global simulation box in y and
periodic in x and z dimension.
read_datafile_def ../../CuNi_model/CuNi_x.lmp
#Give the path of the data file containing information needed to do an analysis
read_datafile_ref ../../CuNi_model/CuNi_r.lmp
#Give the path of the data file containing reference configuration
set neigh_lenth_stan 1 3.21
set neigh_lenth_stan 2 3.14
#Define the cutoff radius of neighbors that are taken into account for 1 and 2 type
atoms
write_datafile CuNi.SVA.lmp

```

```
#Give the path of the output file containing analyzed result  
decimal_digit 3  
#reduce the file size for easy downloading
```

A.1.3 Misfit dislocation in CuNi interface

This example is used to reveal the importance of the correction by a comparative analysis of misfit dislocation at Cu{111}://{111}Ni interface.

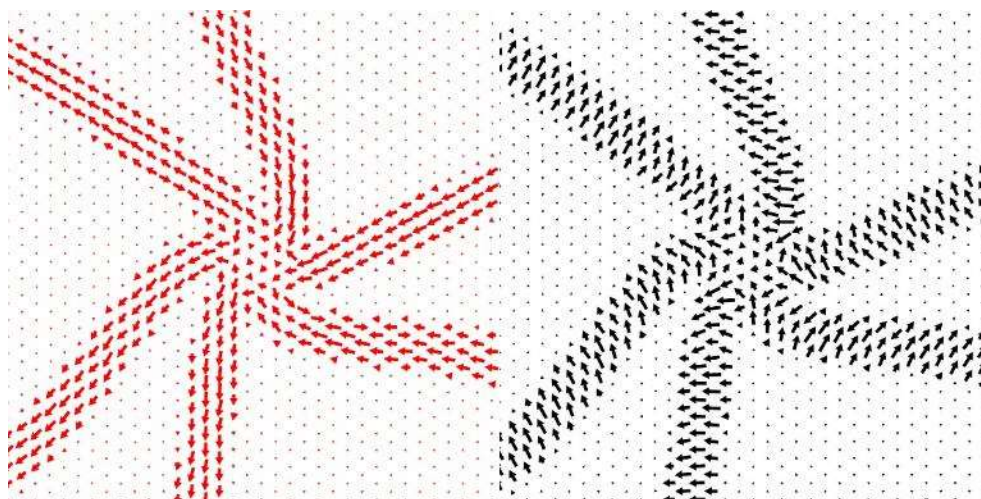


Fig. A.1.3.1 The vector plot of misfit dislocation at Cu{111}://{111}Ni interface revealed by NTA method in assigned mode: representation of line sense (left) and Burgers vector (right)

As described in preceding sections, the choice of angle cutoff does not significantly influence the analyzed line sense and Burgers vector, but it will strongly modify the position of dislocation.

With CNA correction, one may get an appropriate choice of cutoff and then the sessile Lomer-cottrell lock with Burgers vector of $\langle 110 \rangle / 6$ can be clearly identified and shown in Figure A.1.1.2b-c. Please compare with the analyzed results shown in Figure A.1.1.1 without CNA correction.

Fig. A.1.3.2 shows the distribution of misfit dislocations by CNA method, which can identify the position of misfit dislocation at the boundary of fcc and hcp structure. Fig. A.1.3.3 presents the coordination numbers of each atom around a dislocation core, in which the atoms with an ideal 12 neighbors are removed for a better visualization.

If no CNA correction is used, all atoms will be mapped to the fcc structure, thus resulting in the coordination number of atoms in hcp region equal to 9, as shown in Fig. A.1.3.4.

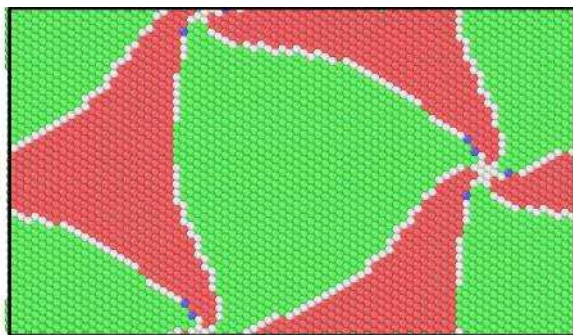


Fig. A.1.3.2 Atomic structures at Cu{111}://{111}Ni interface identified by CNA method. Red for hcp atoms, green for fcc atoms and white for others.

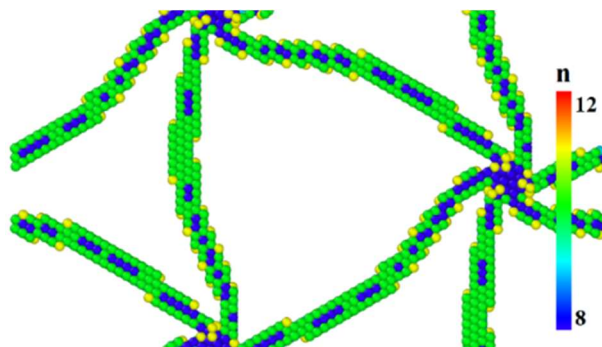


Fig. A.1.3.3 Coordination numbers at Cu{111}://{111}Ni interface revealed by NTA analysis in assigned mode with CNA correction.

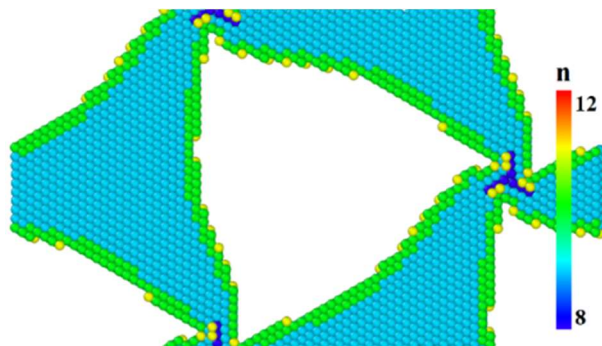


Fig. A.1.3.4 Coordination numbers at hcp region revealed by NTA analysis in assigned mode with

A.1.4 Screw dislocations in bcc-Fe

This example is used to show the importance of mapping correction in DDA method by comparative analysis of $\langle 111 \rangle/2$ screw dislocation in bcc-Fe crystal.

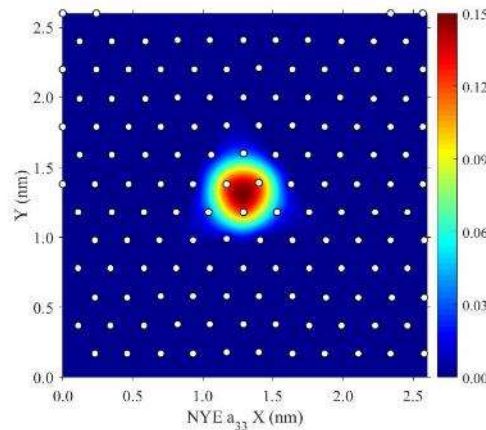


Fig. A.1.4.1 The distribution of σ_1 for $\langle 111 \rangle/2$ screw dislocation analyzed by NTA method in assigned mode

1.4.1 DDA

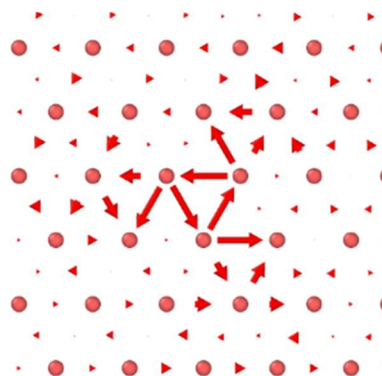


Fig. A.1.4.2 Differential displacement maps of the screw dislocation introduced in bcc-Fe crystal

For DDA analysis, the INPAR looks like:

```
#dda
#to activate the DDA analysis
boundary s s p
#Set the style of boundaries
dda_burgers 0.0 0.0 2.48246182
#Set the style of boundaries
read_datafile_def ../DISFILE
# Give the path of the data file containing information needed to do an analysis
```

```
read_datafile_ref ../REFFILE
#Give the path of the data file containing reference configuration
write_datafile DDA.lmp
#Give the path of the output file containing analyzed result
EDGE
#output the EDGE part of the differential displacement
#The SCREW part of the differential displacement will be output by default and no
need to active
edge_bond
#project the edge part of differential displacements in the edge direction
neighbor_cutoff 1 3.5
#Define the cutoff radius of neighbors that are taken into account
decimal_digit 3
#reduce the file size for easy downloading
```

1.4.2 Mapping Correction

For STA and SVA method, one may use the following command to remove the effect of total translation within slip plane. With the provided Burgers vector, the program will correct all bonds to remove the total translation of a distance of Burgers vector. Please refer to **burgers** command in chapter 2 for more details.

```
burgers 0.0 0.0 2.48246182
#design the Burgers vector for slip
```

Fig. A.1.4.3 illustrates the influence of total translation within slip plane on the analyzed results. Please refer to the example in folder of *example\Fe_screw*

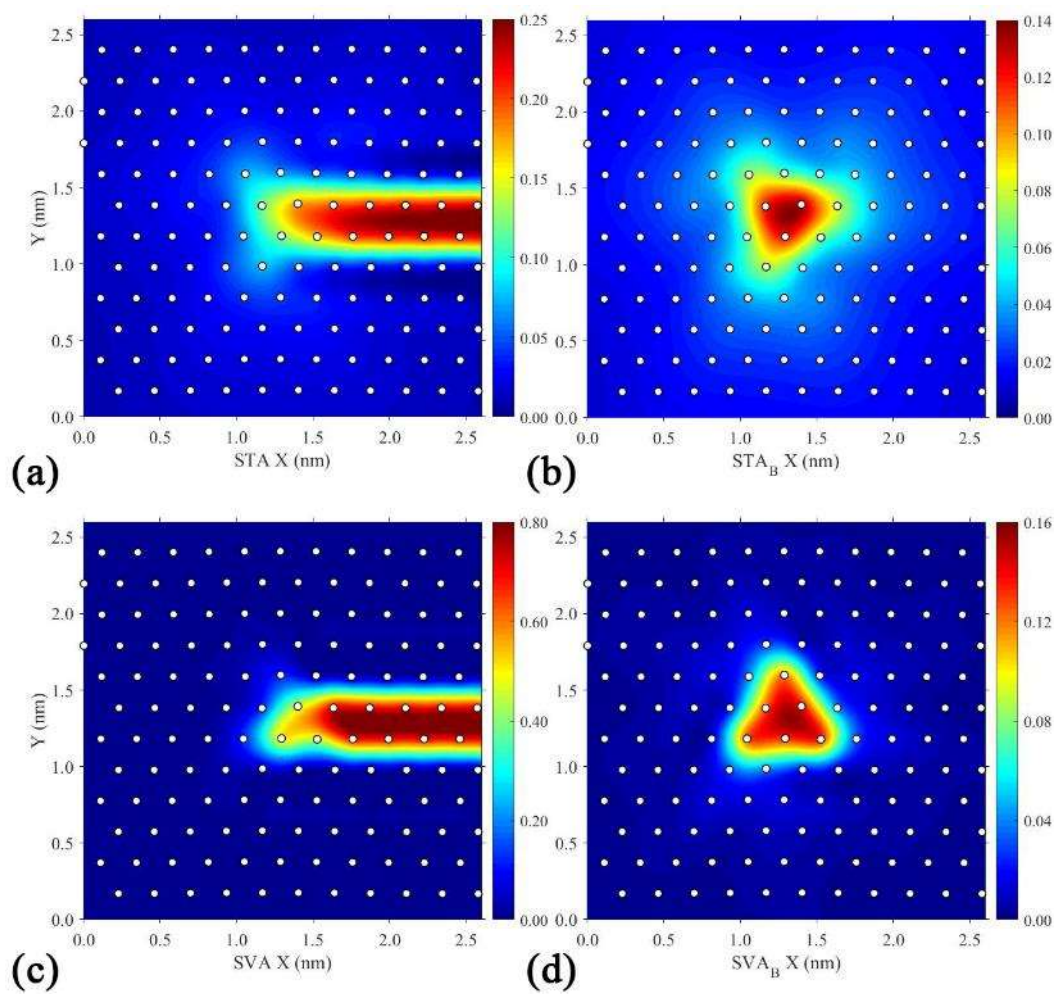


Fig. A.1.4.3 The screw dislocation in bcc-Fe crystal revealed by STA method (a) and (c) without, (b) and (d) with mapping correction.

A.1.5 Edge dislocations in hcp-Mg

Fig. A.1.5.1 shows the core structure of the two partials of the $\langle 11\bar{2}0 \rangle/3$ edge-type dislocation in hcp-Mg crystal. To print out the 9 components of Nye tensor, please issue the following command:

```
nye_tensors
#output the Nye tensors.
```

With this command activated, the program will generate a file named with suffix `_ntensor` which contains 9 components of the Nye tensor. Please refer to the folder in `example\Mg\` for more details.

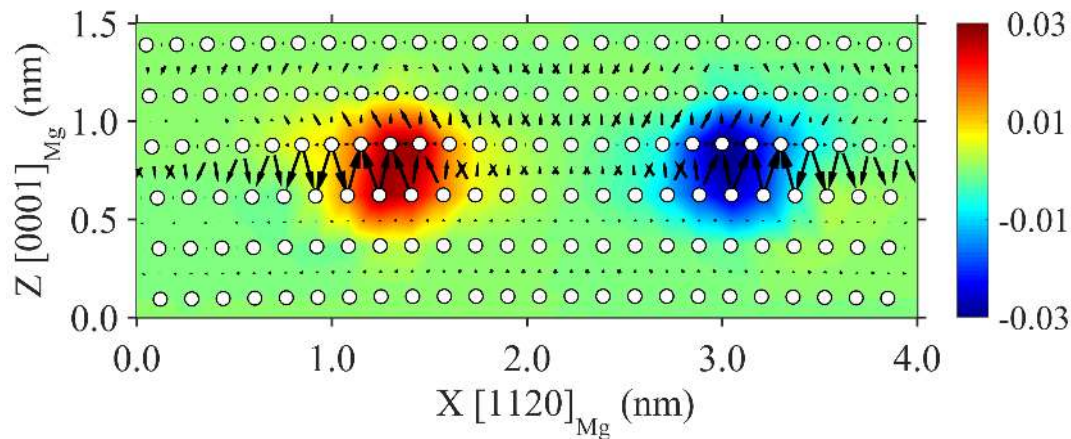


Fig. A.1.5.1 The screw components of $\langle 11\bar{2}0 \rangle/3$ edge dislocations in hcp-Mg crystal revealed by differential displacements (arrows) and Nye tensor distribution (color map) superimposed on atomic positions (circles).

For atoms in hcp structure, the program will analyze the stacking sequence of each atom from which one may get the matching status. Fig. A.1.5.2 shows that two kinds of hcp-structured atoms that are well divided into two layers of A and B.

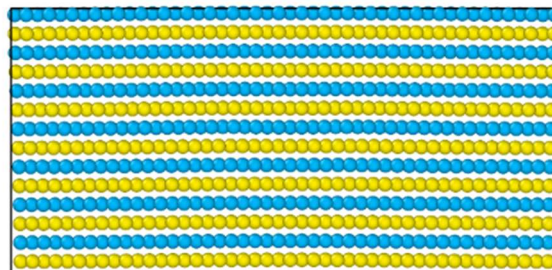


Fig. A.1.5.2 The AB atomic planes in hcp stacking sequence at the edge dislocations in hcp-Mg crystal

A.2 Precautions

A.2.1 O-lattice reference

When appearing large density of misfit dislocations, the analyzed results by the NTA method in assigned mode will be smeared by the strong interaction of dislocation core. By using lattice parameters of Al and Mg ($a_{\text{Al}}=4.0495 \text{ \AA}$ and $a_{\text{Mg}}=3.2094 \text{ \AA}$) for analysis, Fig. A.2.1.1 presents the analyzed misfit dislocations at the Al{111}://{0001}Mg interface. One may apply respective lattice constants for different constitute metals as pasted below:

```
lattice_const 1 4.0495  
lattice_const 2 3.2094
```

Fig. A.2.1.1c-d presents the feature of misfit dislocations at Al{111}://{0001}Mg interface analyzed by NTA method in assigned mode using the respective lattice constants. It is seen that a highly dispersed feature appears along the misfit dislocations. To make the feature of line sense more clear, one may use an average lattice constant of Al and Mg, and the analyzed results are shown in Fig. A.2.1.1a-b. The average lattice constant can be applied by issuing the commands below:

```
lattice_const 1 4.2941  
lattice_const 2 3.0364
```

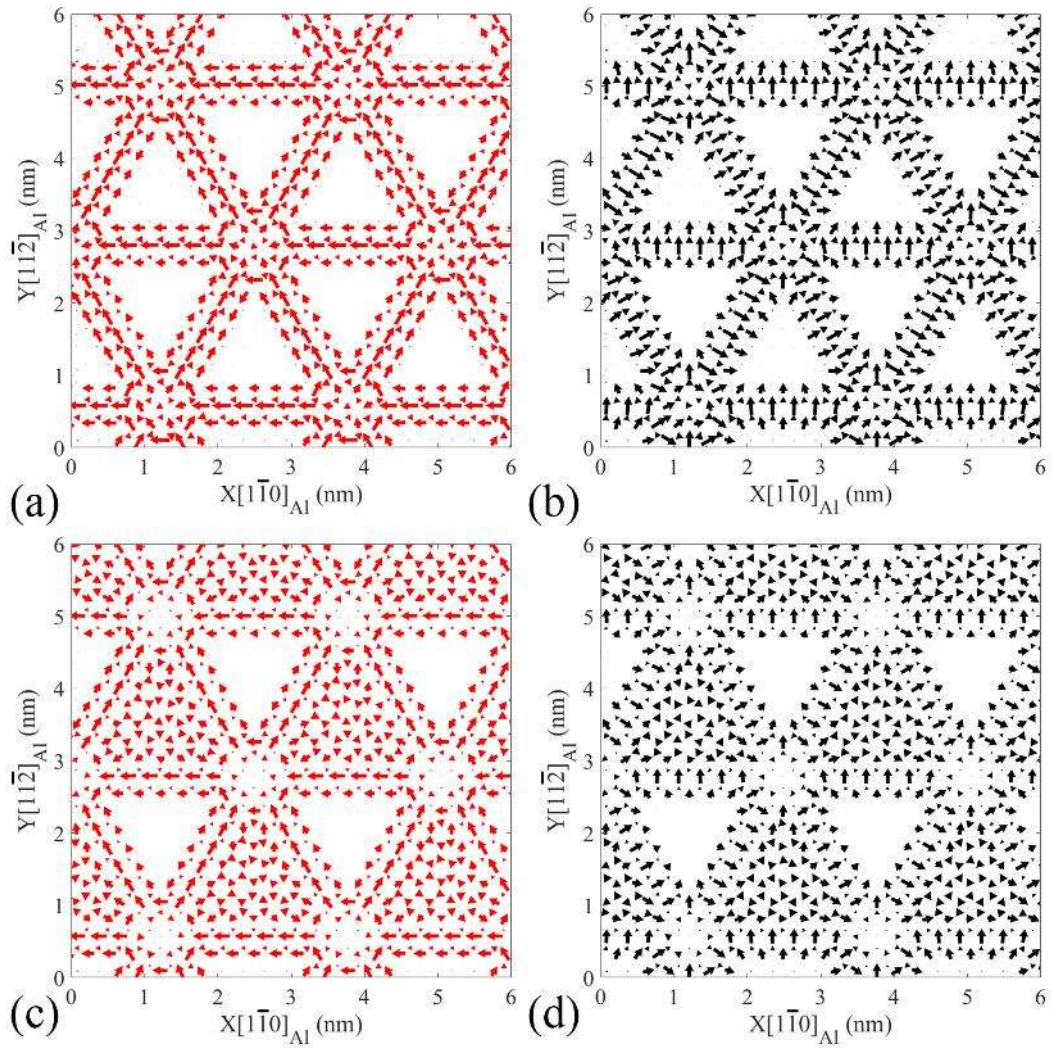


Fig. A.2.1.1 The feature of misfit dislocations at Al{111}//{0001}Mg interface analyzed by NTA method in assigned mode. Represented by (a) Line sense and (b) Burgers vector using the average lattice constants, (c) Line sense and (d) Burgers vector using respective lattice constants.

A.2.2 Coordination

The coordination environment of bcc, fcc and hcp structure are listed as follows:

bcc		
x	y	z
1	0	0
0	1	0
0	0	1
1/2	1/2	1/2
-1/2	1/2	1/2
1/2	-1/2	1/2
1/2	1/2	-1/2
-1	0	0
0	-1	0
0	0	-1
-1/2	-1/2	-1/2
1/2	-1/2	-1/2
-1/2	1/2	-1/2
-1/2	-1/2	1/2

fcc		
x	y	z
0	1/2	1/2
1/2	0	1/2
1/2	1/2	0
0	-1/2	1/2
-1/2	0	1/2
-1/2	1/2	0
0	-1/2	-1/2
-1/2	0	-1/2
-1/2	-1/2	0
0	1/2	-1/2
1/2	0	-1/2
1/2	-1/2	0

hcp		
x	y	z
1	0	0
-1	0	0
1/2	$\sqrt{3}/2$	0
-1/2	$\sqrt{3}/2$	0
-1/2	$-\sqrt{3}/2$	0
1/2	$-\sqrt{3}/2$	0
0	$1/\sqrt{3}$	$\sqrt{2/3}$
-1/2	$-1/\sqrt{12}$	$\sqrt{2/3}$
1/2	$-1/\sqrt{12}$	$\sqrt{2/3}$
0	$1/\sqrt{3}$	$-\sqrt{2/3}$
-1/2	$-1/\sqrt{12}$	$-\sqrt{2/3}$
1/2	$-1/\sqrt{12}$	$-\sqrt{2/3}$

References

- [1] A. Stukowski, *Modell. Simul. Mater. Sci. Eng.* 18 (2010) 015012.
- [2] M.I. Mendelev, S. Han, D.J. Srolovitz, G.J. Ackland, D.Y. Sun, M. Asta, *Philos. Mag.* 83 (2003) 3977-3994.
- [3] G. Bonny, R.C. Pasianot, N. Castin, L. Malerba, *Philos. Mag.* 89 (2009) 3531-3546.
- [4] D.Y. Sun, M.I. Mendelev, C.A. Becker, K. Kudin, T. Haxhimali, M. Asta, J.J. Hoyt, A. Karma, D.J. Srolovitz, *Physical Review B* 73 (2006) 024116.