Table of Contents

Cover 1.1

Introduction 1.2

- PyCAC features and non-features 1.2.1
- Compilation and execution 1.2.2
- CAC Publications 1.2.3
- Acknowledgements and citations 1.2.4

Background 1.3

- Atomic field theory 1.3.1
- Governing equations of the CAC method 1.3.2

Algorithm 1.4

- Module 1.4.1
- Subroutine 1.4.2
- Parallelization 1.4.3
- Arithmetic precision 1.4.4

Python Interface 1.5

Command 1.6

- bd_group 1.6.1
- boundary 1.6.2
- box 1.6.3
- cal_num 1.6.4
- cal 1.6.5
- constrain 1.6.6
- convert 1.6.7
- debug 1.6.8
- deform 1.6.9
- dump 1.6.10
- dynamics 1.6.11

ele_size 1.6.12
ensemble 1.6.13
force_dir 1.6.14
grain_dir 1.6.15
grain_mat 1.6.16
grain_move 1.6.17
grain_num 1.6.18
grain_uc 1.6.19
group_num 1.6.20
group 1.6.21
lattice 1.6.22
limit 1.6.23
mass 1.6.24
mass_mat 1.6.25
minimize 1.6.26
modify_num 1.6.27
modify 1.6.28
neighbor 1.6.29
potential 1.6.30
refine 1.6.31
restart 1.6.32
run 1.6.33
simulator 1.6.34
temperature 1.6.35
uc_num 1.6.36
zigzag 1.6.37
Post-processing 1.7
Examples 1.8

PyCAC User's Manual

Mar 15 2017 version

Introduction

A brief discussion of what concurrent atomistic-continuum (CAC) method is: it is written in FORTRAN with a Python interface.

PyCAC features and non-features

what PyCAC can and can not do: e.g., it can simulate FCC pure metals with LJ and EAM potentials, can not do HCP and multicomponent metals etc

Compilation and execution

compiler, OS and hardware requirement

users may run it on MATIN when it is ready

CAC Publications

A list of book chapters on CAC:

1. Shengfeng Yang, Youping Chen, Concurrent atomistic-continuum simulation of defects in polyatomic ionic materials, in *Multiscale Materials Modeling for Nanomechanics* (ed: Christopher R. Weinberger, Garritt J. Tucker), Switzerland: Springer International Publishing, 2016

A list of dissertations and theses on CAC:

1. Shuozhi Xu, The concurrent atomistic-continuum method: Advancements and applications in plasticity of face-centered cubic metals, *Ph.D. Dissertation*, Georgia Institute of Technology, 2016
2. Xiang Chen, A concurrent atomistic-continuum study of phonon transport in crystalline materials with microstructures, *Ph.D. Dissertation*, University of Florida, 2016
3. Shengfeng Yang, A concurrent atomistic-continuum method for simulating defects in ionic materials, *Ph.D. Dissertation*, University of Florida, 2014
4. Qian Deng, Coarse-graining atomistic dynamics of fracture by finite element method: Formulation, parallelization and applications, *Ph.D. Dissertation*, University of Florida, 2011
5. Liming Xiong, A concurrent atomistic-continuum methodology and its applications, *Ph.D. Dissertation*, University of Florida, 2011

A list of peer-reviewed journal articles on CAC (by acceptance date):

1. Shuozhi Xu, Liming Xiong, Youping Chen, David L. McDowell. Comparing EAM potentials to model slip transfer of sequential mixed character dislocations across two symmetric tilt grain boundaries in Ni, *JOM* (in press)
2. Shuozhi Xu, Liming Xiong, Youping Chen, David L. McDowell. Shear stress- and line length-dependent screw dislocation cross-slip in FCC Ni, *Acta Mater.* 122 (2017) 412-419
3. Shuozhi Xu, Liming Xiong, Youping Chen, David L. McDowell. An analysis of key characteristics of the Frank-Read source process in FCC metals, *J. Mech. Phys. Solids* 96 (2016) 460-476
4. Shuozhi Xu, Liming Xiong, Youping Chen, David L. McDowell. Edge dislocations bowing out from a row of collinear obstacles in Al, *Scr. Mater.* 123 (2016) 135-139
5. Shuozhi Xu, Liming Xiong, Qian Deng, David L. McDowell. Mesh refinement

schemes for the concurrent atomistic-continuum method, *Int. J. Solids Struct.* 90 (2016) 144-152

6. Shuozhi Xu, Liming Xiong, Youping Chen, David L. McDowell. Sequential slip transfer of mixed character dislocations across $$\Sigma 3$$ coherent twin boundary in FCC metals: A concurrent atomistic-continuum study, *npj Comput. Mater.* 2 (2016) 15016
7. Liming Xiong, Ji Rigelesaiyin, Xiang Chen, Shuozhi Xu, David L. McDowell, Youping Chen. Coarse-grained elastodynamics of fast moving dislocations, *Acta Mater.* 104 (2016) 143-155
8. Shengfeng Yang, Ning Zhang, Youping Chen. Concurrent atomistic-continuum simulation of polycrystalline strontium titanate, *Philos. Mag.* 95 (2015) 2697-2716
9. Shuozhi Xu, Rui Che, Liming Xiong, Youping Chen, David L. McDowell. A quasistatic implementation of the concurrent atomistic-continuum method for FCC crystals, *Int. J. Plast.* 72 (2015) 91–126
10. Shengfeng Yang, Youping Chen. Concurrent atomistic and continuum simulation of bi-crystal strontium titanate with tilt grain boundary, *Proc. R. Soc. A* 471 (2015) 20140758
11. Liming Xiong, Shuozhi Xu, David L. McDowell, Youping Chen. Concurrent atomistic-continuum simulations of dislocation-void interactions in fcc crystals, *Int. J. Plast.* 65 (2015) 33-42
12. Liming Xiong, Xiang Chen, Ning Zhang, David L. McDowell, Youping Chen. Prediction of phonon properties of 1D polyatomic systems using concurrent atomistic-continuum simulation, *Arch. Appl. Mech.* 84 (2014) 1665-1675
13. Liming Xiong, David L. McDowell, Youping Chen. Sub-THz Phonon drag on dislocations by coarse-grained atomistic simulations, *Int. J. Plast.* 55 (2014) 268-278
14. Qian Deng, Youping Chen, A coarse-grained atomistic method for 3D dynamic fracture simulation, *J. Multiscale Comput. Eng.* 11 (2013) 227-237
15. Shengfeng Yang, Liming Xiong, Qian Deng, Youping Chen. Concurrent atomistic and continuum simulation of strontium titanate, *Acta Mater.* 61 (2013) 89–102
16. Liming Xiong, David L. McDowell, Youping Chen. Nucleation and growth of dislocation loops in Cu, Al and Si by a concurrent atomistic-continuum method, *Scr. Mater.* 67 (2012) 633–636
17. Liming Xiong, Qian Deng, Garritt Tucker, David L. McDowell, Youping Chen. Coarse-grained atomistic simulations of dislocations in Al, Ni and Cu crystals, *Int. J. Plast.* 38 (2012) 86–101
18. Liming Xiong, Qian Deng, Garritt Tucker, David L. McDowell, Youping Chen. A concurrent scheme for passing dislocations from atomistic to continuum domains,

Acta Mater. 60 (2012) 899-913

19. Liming Xiong, Garritt Tucker, David L. McDowell, Youping Chen. Coarse-grained atomistic simulation of dislocations, *J. Mech. Phys. Solids* 59 (2011) 160-177
20. Qian Deng, Liming Xiong, Youping Chen. Coarse-graining atomistic dynamics of brittle fracture by finite element method, *Int. J. Plast.* 26 (2010) 1402-1414

Acknowledgements and citations

PyCAC development has been funded by

- National Science Foundation
  - Georgia Institute of Technology, CMMI-1232878
  - University of Florida, CMMI-1233113
  - Iowa State University, CMMI-1536925
- Department of Energy, Office of Basic Energy Sciences
  - University of Florida, DE-SC0006539

If you use PyCAC results in your published work, please cite

- Shuozhi Xu, Rui Che, Liming Xiong, Youping Chen, David L. McDowell. A quasistatic implementation of the concurrent atomistic-continuum method for FCC crystals, *Int. J. Plast.* 72 (2015) 91–126
- Liming Xiong, Garritt Tucker, David L. McDowell, Youping Chen. Coarse-grained atomistic simulation of dislocations, *J. Mech. Phys. Solids* 59 (2011) 160-177

This user's manual is maintained by

- Shuozhi Xu (shuozhixu@gatech.edu)
- Thomas G. Payne (thomas.payne@gatech.edu)

Background

Background of CAC

Atomic field theory

Dr. Chen will help write this section

Governing equations of the CAC method

Algorithm

CAC algorithm

Module

files in the module folder

Subroutine

Files in the src folder

Parallelization

spatial decomposition algorithm

Arithmetic precision

the users can choose between 16 and 32 digit precision

Python interface

how python works with CAC

Command

This chapter describes how a PyCAC input script `cac.in` is formatted and the input script commands used to define a CAC simulation.

CACS reads the entire `cac.in` and then performs a simulation with all the settings. Thus, the sequence of commands does not matter.

A line with the "#" character in the beginning is treated as a comment and discarded.

Many input script errors are detected and an Error or Warning message is printed.

Below is a list of all 37 CACS commands, grouped by category.

Simulation box:

boundary, box, ele_size, grain_dir, grain_mat, grain_move, grain_num, grain_uc, modify_num, modify_#, uc_num, zigzag

Materials:

lattice, mass, potential

Settings:

bd_group, cal_num, cal_#, constrain, dump, ensemble, force_dir, group, group_#, limit, mass_mat, neighbor, simulator, temperature

Actions:

deform, dynamics, minimize, refine, restart, run

Miscellanies:

convert, debug

bd_group

Syntax

```
bd_group x boolean_l boolean_u style_cg style_at depth boolean_def time_start time
_end
         y boolean_l boolean_u style_cg style_at depth boolean_def time_start time
_end
         z boolean_l boolean_u style_cg style_at depth boolean_def time_start time
_end
```

- boolean_l, boolean_u, boolean_def = *t* or *f*

  ```
  t is true
  f is false
  ```

- style_cg = *null* or *element* or *node*
- style_at = *null* or *atom*
- depth = real number
- time_start, time_end = an integer

Examples

```
bd_group x f f null atom 2. t 200 1000 y t f node atom 3. t 0 1000 z t t element n
ull 1. f 500 1000
```

Description

The `bd_group` command provides a shortcut to create groups for the elements/nodes/atoms within a certain distance from each boundary. The IDs of these groups are after the regular groups created by the group command. The groups created using this command are rigid, i.e., the elements/nodes/atoms are not displaced subject to the interatomic forces, expect (possibly) following the overall deformation of the simulation cell.

boolean_l and boolean_u decide whether the lower and upper boundaries along each axis are involved, respectively. If any of these two boolean is true, style_cg and style_at decide whether the group contains elements, nodes, atoms, or null, see here for the differences between *element* and *node*.

All groups defined by the `bd_group` command have a block shape. Along the y axis, for example, the two block-shape groups are bounded by

```
x inf inf y inf depth z inf inf
x inf inf y upper_b-depth inf z inf inf
```

where upper_b is the upper bound of the simulation cell, similar to that in the group command. Note that the depth must be a real number, e.g., `2.`, instead of an integer, e.g., `2`. The depth is in unit of the lattice_space_max along respective axis.

boolean_def decides whether the group is deformed along with the simulation cell, similar to the one in the group command.

time_start and time_end, in the unit of time step, are two integers of when the groups begin and stop being rigid.

Related commands

This command provides a shortcut to create groups, which can be done by the group command.

Related files

`group_init.f90`

Default

None.

boundary

Syntax

```
boundary bc_x bc_y bc_z
```

- bc_x, bc_y, bc_z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`cac_init.f90` , among many

Default

```
boundary s s s
```

box

Syntax

```
box x i j k y i j k z i j k
```

- i,j,k = real number

Examples

```
box x 1. 0. 0. y 0. 1. 0. z 0. 0. 1.
box x 1. 0. 0. y 0. 0.94281 -0.33333 z 0. 0. 1.
```

Description

Decide the grain boundary (GB) plane orientation with respect to the simulation cell when there is more than one grain as defined by grain_num

Suppose that the grain_dir = 2, i.e., the grains are aggregated along the y direction, the `box` command in the first example results in a GB plane normal to the y axis; the `box` command in the second example, however, results in a GB plane inclined with respect to the y axis, as shown in Fig. 1 of Xu et al. 2016 and Fig. 1(a) of Xu et al. 2017.

Related commands

As opposed to the grain_mat command whose orientations are for the lattice, the orientations in the concurrent command are with respect to the simulation cell. To convert the lattice orientation to the simulation cell orientation, one may use the convert to calculate the latter based on the former.

Related files

`model_init.f90` , among many

Default

```
box x 1. 0. 0. y 0. 1. 0. z 0. 0. 1.
```

cal_num

Syntax

```
cal_num c_num
```

- c_num = integer (<= 19))

Examples

```
cal_num 0
cal_num 3
```

Description

Number of group-based calculations.

Related commands

Calculations in this command are based on group.

Related files

`dump_init.f90` and `group_cal.f90`

Default

```
cal_num 0
```

cal

Syntax

```
cal group_name cal_variable
```

- group_name = a string with length <= 30
- cal*variable* = _*energy* or *force* or *stress*

Examples

```
cal group_1 force
cal group_3 stress
```

Description

Decide a group and associated quantities that are calculated and saved to a file named `group_cal_#` where # is the ID of cal.

The style *energy*, *force*, and *stress* correspond to a scalar, a 3 by 1 vector, and a 3 by 3 tensor, respectively.

Related commands

There cannot be fewer `cal` commands than the cal_num defined by the `cal_num` command. The group name must match one of the groups defined in the `group` command.

Related files

`group_cal.f90`

Default

None.

constrain

Syntax

```
constrain boolean
```

- boolean = *t* or *f*

  ```
  t is true
  f is faulse
  ```

Examples

```
constrain t
```

Description

Decide if a constrain is added to the system. When `boolean` is *t*, the nodes and atoms are only allowed to move along the direction specified by the force_dir command, either in dynamic or quasi-static simulations.

Related commands

force_dir

Related files

`constraint.f90`

Default

`boolean` = f

convert

Syntax

```
convert i j k
```

- i,j,k = real number

Examples

```
convert -1. 1. 2.
convert 1. -1. 0.
```

Description

Convert the lattice orientation [i,j,k] to the orientation with respect to the simulation cell.

Related commands

This command is useful when the user has a set of lattice orientations in mind and wants to find the orientation with respect to the simulation cell, e.g., to be used in the box command.

Related files

`convert_direction.f90`

Default

None.

debug

Syntax

```
debug boolean_debug boolean_mpi
```

- boolean_debug, boolean_mpi = *t* or *f*

  ```
  t is true
  f is faulse
  ```

Examples

```
debug t f
debug t t
```

Description

Generate a writable file named `debug` for testing purposes. The file is created only when `boolean_debug` is t; the unit number is 13. The user can then write whatever they want into this file using unit number 13. When `boolean_mpi` is t, all processors have access to this file, otherwise only the root does.

Related commands

None.

Related files

`debug_init.f90`

Default

```
debug f f
```

deform

Syntax

```
deform boolean_def def_num
       ij boolean_cg boolean_at def_rate stress_l stress_u flip_frequency
       time time_start time_always_flip time_end
```

- boolean_def, boolean_cg, boolean_at = *t* or *f*

```
t is true
f is false
```

- def_num = integer (<= 9)
- ij = *xx* or *yy* or *zz* or *xy* or *yz* or *yz* or *zy* or *xz* or *zx*
- def_rate, stress_l, stress_u = real number
- flip_frequency, time_start, time_always_flip, time_end = integer

Examples

```
deform t 1 zx t t 0.05 0.6 0.7 10 time 500 1000 2500
deform t 2 xx t f 0.01 1. 1.2 20 yz f t 0.02 0.8 0.9 30 time 400 600 1900
```

Description

Set up the homogeneous deformation of the whole simulation box. The deformation is underway if `boolean_def` is t.

`def_num` defines how many deformation matrix is added. When `def_num` > 1, all deformation matrices are linearly superimposed.

`ij` decides how the strain is applied. Following the standard indexes in continuum mechanics, `i` and `j` are the face on which and the direction along which the strain is applied. When `i` and `j` are the same, a uniaxial strain is set, otherwise, a shear strain is set.

`def_rate` is the strain rate, in the unit of ps$$^{-1}$$.

`stress_l` and `stress_u` are the lower and upper bounds of the applied stress, respectively, for the stress tensor component specified by `ij` , in unit of GPa. Assume that all stress components are initially zero or very small. Then when the stress component is higher than `stress_u` , the corresponding strain rate changes sign, i.e., the deformation is reversed. Afterward, when the stress component is lower than `stress_l` , the corresponding strain rate changes sign again, i.e., the deformation proceeds as the initial setting. Whether the stress component is out of the bounds is checked not every step, but at every `flip_frequency` step.

The deformation begins after time step `time_start` and ends after `time_end` . When the time step is larger than `time_always_flip` but smaller than `time_end` , the strain rate changes sign at every step back and forth, regardless of the stress bounds defined by `stress_l` and `stress_u` .

Related commands

Groups defined by the group and bd_group commands may be homogeneous deformed along with the simulation cell, depending on settings in those two commands.

Related files

`deform_init.f90` and `deform_box.f90`

Default

`boolean_def` = f.

dump

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`dump_init.f90` and `dump.f90`

Default

dynamics

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`dynamics_init.f90` , `dynamics.f90` , `langevin_dynamics.f90` , `quenched_dynamics.f90` , `hybrid.f90` , among many

Default

ele_size

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`model_init.f90`

Default

ensemble

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`thermostat.f90`

Default

force_dir

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`constraint.f90`

Default

grain_dir

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`box_init.f90` and `model_init.f90`

Default

grain_mat

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`grain.f90`

Default

grain_move

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`box_init.f90`

Default

grain_num

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`box_init.f90` and `grain.f90`

Default

grain_uc

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`box_init.f90`

Default

group_num

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`group_init.f90` and `group.f90`

Default

group

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`group.f90`

Default

lattice

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`box_init.f90` and `lattice.f90`

Default

limit

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`neighbor_init.f90`, `update_neighbor.f90`, `cell_neighbor_list.f90`, `update_cell_neighbor.f90`, and `update_cell.f90`

Default

mass

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
      p is periodic
      f is non-periodic and fixed
      s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`crystal.f90` and `mass_matrix.f90` , among many

Default

mass_mat

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`mass_matrix.f90` and `update_`

Default

minimize

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`mini_init.f90` , `mini_energy.f90` , and `hybrid.f90` , among many

Default

modify_num

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`model_modify.f90`

Default

modify

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`model_modify.f90`

Default

neighbor

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`neighbor_init.f90` and `update_neighbor.f90`

Default

potential

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
      p is periodic
      f is non-periodic and fixed
      s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Related files

`potential.f90` , `eam_tab.f90` , `lj_para.f90` , among many

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

boundary

Syntax

```
boundary x y z
```

- x,y,z = *p* or *s* or *f*

  ```
  p is periodic
  f is non-periodic and fixed
  s is non-periodic and shrink-wrapped
  ```

Examples

```
boundary p f s
```

Description

Set the style of boundaries for the global simulation box in each dimension. The same style is assigned to both the lower and upper face of the box.

The style *p* means the box is periodic, so that atoms/nodes interact across the boundary, and they can exit one end of the box and re-enter the other end.

The styles *f* and *s* mean the box is non-periodic, so that particles do not interact across the boundary and do not move from one side of the box to the other. For style *f*, the position of the face is fixed. For style *s*, the position of the face is set so as to encompass the atoms in that dimension (shrink-wrapping), no matter how far they move.

Related commands

When the style of a boundary is *p*, the corresponding zigzag arg is changed to *f*. In other words, a boundary has to be flat to apply the periodic boundary condition.

Default

Post-processing

visualization using ovito and paraview

Examples

five CAC examples with movies and input files