Alliance

1.0

Generated by Doxygen 1.9.1

1 Class Index	1
1.1 Class List	1
2 File Index	3
2.1 File List	3
3 Class Documentation	5
3.1 array_size Struct Reference	5
3.1.1 Detailed Description	5
4 File Documentation	7
4.1 array.c File Reference	7
4.1.1 Detailed Description	8
4.1.2 Function Documentation	8
4.1.2.1 get_flat_c()	8
4.1.2.2 get_flat_r()	8
4.1.2.3 get_flatIndexComplex3D()	9
4.1.2.4 getIndChi()	9
4.1.2.5 getIndChiBufEL_c()	9
4.1.2.6 getIndChiBufEL_r()	11
4.1.2.7 getIndChiBufEM_c()	11
4.1.2.8 getIndChiBufEM_r()	12
4.1.2.9 multiply_ar_c()	12
4.1.2.10 multiply_ar_r()	12
4.2 init.c File Reference	12
4.2.1 Detailed Description	13
4.2.2 Macro Definition Documentation	13
4.2.2.1 RANK_IO	13
4.2.3 Function Documentation	13
4.2.3.1 fill_rand()	13
4.2.3.2 fill_randM0()	14
4.2.3.3 fill_randSingleKM()	14
4.2.3.4 init_conditions()	14
4.2.3.5 init_energySpec()	15
4.2.3.6 init_initEnums()	15
4.2.3.7 init_printParameters()	15
4.2.3.8 init_start()	15
Index	17

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:	
array_size	Ę

2 Class Index

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

array.c		
	Array manipulation module	7
init.c		
	Initialization module for alliance	15

File Index

Class Documentation

3.1 array_size Struct Reference

3.1.1 Detailed Description

gives array sizes

The documentation for this struct was generated from the following file:

• array.c

6 Class Documentation

File Documentation

4.1 array.c File Reference

array manipulation module

```
#include "array.h"
#include "utils_fftw.h"
#include "space_config.h"
```

Macros

- #define CHI_EM 3
- #define CHI_EL 1
- #define **FFT_OFFSET** 2

Functions

```
    size_t get_flat_c (size_t is, size_t il, size_t im, size_t ix, size_t iy, size_t iz)
        returns flat index of the element of complex 6D array
    size_t getIndChiBufEM_c (size_t ix, size_t iy, size_t iz, size_t is, size_t ifield)
        returns flat index of an element of electromagnetic gyrokinetic potential in FOURIER SPACE
    size_t getIndChiBufEM_r (size_t ix, size_t iy, size_t iz, size_t is, size_t ifield)
        returns flat index of an element of electromagnetic gyrokinetic potential in POSITION SPACE
```

size_t getIndChiBufEL_c (size_t ix, size_t iy, size_t iz, size_t is)

0120_t gottindombdi22_0 (0120_t ix, 0120_t iy, 0120_t iz, 0120_t io)

returns returns flat index of an element of electrostatic gyrokinetic potential in FOURIER SPACE

size_t getIndChiBufEL_r (size_t ix, size_t iy, size_t iz, size_t is)

returns returns flat index of an element of electrostatic gyrokinetic potential in REAL SPACE

size_t get_flat_r (size_t is, size_t il, size_t im, size_t ix, size_t iy, size_t iz)

returns flat index of the element of real 6D array

size_t get_flatIndexComplex3D (size_t ix, size_t iy, size_t iz)

returns flat array of complex 3D array

- size_t getIndChi (size_t ix, size_t iy, size_t iz, size_t is)
- void multiply ar c (COMPLEX *ar1, COMPLEX *ar2, COMPLEX *ret)
- void multiply_ar_r (const double *ar1, const double *ar2, double *ret)

Variables

- struct array_size array_local_size
- struct array_size array_global_size
- · struct offset size array offset
- struct offset_size array_offset3D

4.1.1 Detailed Description

array manipulation module

contains functions which are supposed to make array manipulation simpler

4.1.2 Function Documentation

4.1.2.1 get_flat_c()

returns flat index of the element of complex 6D array

Parameters

is	species type
il	Laguerre moment
im	Hermite moment
ix	kx index
iy	ky index
iz	kz index

returns flattened index of a complex array from its 6D index. Flattened index then can be passed to distribution function 6D array to get a required element at position (is,il,im,ix,iy,iz).

4.1.2.2 get_flat_r()

```
size_t iy,
size_t iz )
```

returns flat index of the element of real 6D array

Parameters

is	species type
il	Laguerre moment
im	Hermite moment
ix	x index
iy	y index
iz	z index

returns flattened index of a real array from its 6D index. Flattened index then can be passed to distribution function 6D array to get a required element at position (is,il,im,ix,iy,iz).

4.1.2.3 get_flatIndexComplex3D()

returns flat array of complex 3D array

Parameters

ix	kx index
iy	ky index
iz	kz index

returns flattened index of a complex array from its 3D position index. Flattened index then can be passed to one of the fields ($\phi(\mathbf{k}), A_{||}(\mathbf{k}), B_{||}(\mathbf{k})$) 6D array to get a required element at position (ix,iy,iz).

4.1.2.4 getIndChi()

getIndChi(size_t ix,size_t iy, size_t iz, size_t is)

4.1.2.5 getIndChiBufEL_c()

```
size_t iy,
size_t iz,
size_t is)
```

returns returns flat index of an element of electrostatic gyrokinetic potential in FOURIER SPACE

Parameters

ix	kx index
iy	ky index
iz	kz index
is	particle species index

returns flattened index of a gyrokinetic potential $\chi^{\phi}(\mathbf{k})$ from its 4D index in FOURIER SPACE. flattened index is then can be used to access required value of the gyrokinetic potential at position (ix,iy,iz,is).

4.1.2.6 getIndChiBufEL_r()

returns returns flat index of an element of electrostatic gyrokinetic potential in REAL SPACE

Parameters

ix	x index
iy	y index
iz	z index
is	particle species index

returns flattened index of a gyrokinetic potential $\chi^{\phi}(\mathbf{r})$ from its 4D index in REAL SPACE. flattened index is then can be used to access required value of the gyrokinetic potential at position (ix,iy,iz,is).

4.1.2.7 getIndChiBufEM_c()

returns flat index of an element of electromagnetic gyrokinetic potential in FOURIER SPACE

Parameters

ix	kx index
iy	ky index
iz	kz index
is	particle species index
ifield	field type

returns flattened index of a gyrokinetic potential $\chi^{\phi,A,B}$ from its 4D index in FOURIER SPACE. flattened index is then can be used to access required value of the gyrokinetic potential at position (ix,iy,iz,is). Type of gyrokinetic potential is specified by ifield parameter. Use 0 is to access $\chi^{\phi}(\mathbf{k})$, 1 to access $\chi^{A}(\mathbf{k})$ and 2 to access $\chi^{B}(\mathbf{k})$.

4.1.2.8 getIndChiBufEM_r()

returns flat index of an element of electromagnetic gyrokinetic potential in POSITION SPACE

Parameters

ix	x index
iy	y index
iz	z index
is	particle species index
ifield	field type

returns flattened index of a gyrokinetic potential $\chi^{\phi,A,B}(\mathbf{k})$ from its 4D index in POSITION SPACE. flattened index is then can be used to access required value of the gyrokinetic potential at position (ix,iy,iz,is). Type of gyrokinetic potential is specified by ifield parameter. Use 0 is to access $\chi^{\phi}(\mathbf{r})$, 1 to access $\chi^{A}(\mathbf{r})$ and 2 to access $\chi^{B}(\mathbf{r})$.

4.1.2.9 multiply_ar_c()

multiply_ar_c(COMPLEX *ar1, COMPLEX *ar2, COMPLEX *ret)

4.1.2.10 multiply_ar_r()

multiply_ar_r(const double *ar1, const double *ar2, double *ret)

4.2 init.c File Reference

initialization module for alliance.

```
#include "init.h"
#include "distrib.h"
```

4.2 init.c File Reference 13

Macros

• #define RANK_IO 0

Functions

void init_start (char *filename)

initialization of ALLIANCE

void init printParameters ()

parameter output

• void init_initEnums ()

enumerator initialization

void fill_rand (COMPLEX *ar1)

fills the inital conditions randomly

void fill_randM0 (COMPLEX *ar1)

fill zeroth Hermite moment with random values

void fill_randSingleKM (COMPLEX *ar1)

fill single chosen wavevector and Hermite moment

void init_conditions (COMPLEX *data)

distribution function initialization

• double init_energySpec (double k, double m, double amp, double disp)

returns energy spectrum

Variables

- · enum adiabatic kinetic
- enum electromagnetic systemType
- · enum initial initialConditions

4.2.1 Detailed Description

initialization module for alliance.

all the inititalization routines are here.

4.2.2 Macro Definition Documentation

4.2.2.1 RANK_IO

```
#define RANK_IO 0
```

defines rank of the processor used to output information to console

4.2.3 Function Documentation

4.2.3.1 fill_rand()

fills the inital conditions randomly

Parameters

data

complex 6d array to fill initializes distribution with spectrum defined in init_energySpec This function is supposed to be used in-module only and should not be used elsewhere outside init.c file.

4.2.3.2 fill_randM0()

fill zeroth Hermite moment with random values

Parameters

```
data | complex 6D array to fill
```

fills 0-th Hermite moment of a distribution function ar1 with random values This function is supposed to be used in-module only and should not be used elsewhere outside init.c file.

4.2.3.3 fill_randSingleKM()

fill single chosen wavevector and Hermite moment

Parameters

```
data complex 6D array
```

initializes single wavevector and Hermite moment of a distribution function with random variable. This function is only for in-module use and should not be used elsewhere outside init.c file.

4.2.3.4 init_conditions()

distribution function initialization

Parameters

data complex 6D array

initializes distribution function with chosen method (see fill_rand, fill_randM0, fill_randSingleKM)

4.2 init.c File Reference 15

4.2.3.5 init_energySpec()

returns energy spectrum

Parameters

k	a wavenumber at which spectrum is computed	
m	Hermite moment at which amplitude is computed	
amp	amplitude of the spectrum	
disp	dispersion of the spectrum	

computes spectrum of form $A \cdot k^2 exp(-2k^2/\sigma^2)$, where $\sigma = disp$, and A = amp This function is supposed to be used in-module only and should not be used elsewhere outside init.c file.

4.2.3.6 init_initEnums()

```
void init_initEnums ( )
```

enumerator initialization

initializes enumerators, which are then used to define if system is adiabatic or kinetic, electromagnetic or electrostatic, and type of initial conditions

4.2.3.7 init_printParameters()

```
void init_printParameters ( )
```

parameter output

prints parameters of the simulation

4.2.3.8 init_start()

initialization of ALLIANCE

Parameters

filename	specifies parameter filename
----------	------------------------------

initializes all the modules required for ALLIANCE to work.

Index

```
array.c, 7
                                                                init.c, 14
                                                           init_initEnums
     get_flat_c, 8
     get_flat_r, 8
                                                                init.c, 15
     get_flatIndexComplex3D, 9
                                                           init_printParameters
     getIndChi, 9
                                                                init.c, 15
     getIndChiBufEL_c, 9
                                                           init_start
     getIndChiBufEL_r, 11
                                                                init.c, 15
     getIndChiBufEM_c, 11
                                                           multiply_ar_c
     getIndChiBufEM_r, 12
                                                                array.c, 12
     multiply ar c, 12
                                                           multiply_ar_r
     multiply_ar_r, 12
                                                                array.c, 12
array_size, 5
                                                           RANK IO
fill rand
                                                                init.c, 13
     init.c, 13
fill_randM0
     init.c, 14
fill_randSingleKM
     init.c, 14
get_flat_c
     array.c, 8
get_flat_r
     array.c, 8
get_flatIndexComplex3D
     array.c, 9
getIndChi
     array.c, 9
getIndChiBufEL_c
     array.c, 9
getIndChiBufEL_r
     array.c, 11
getIndChiBufEM_c
     array.c, 11
getIndChiBufEM r
     array.c, 12
init.c, 12
     fill rand, 13
     fill_randM0, 14
     fill_randSingleKM, 14
     init_conditions, 14
     init_energySpec, 14
     init_initEnums, 15
     init_printParameters, 15
     init_start, 15
     RANK_IO, 13
init_conditions
     init.c, 14
init_energySpec
```