

Brownian motion of interacting particles

Generated by Doxygen 1.8.13

Contents

1 Namespace Index	1
1.1 Namespace List	1
2 Class Index	3
2.1 Class List	3
3 File Index	5
3.1 File List	5
4 Namespace Documentation	7
4.1 code_zippull_server Namespace Reference	7
4.1.1 Variable Documentation	7
4.1.1.1 string_copy	7
4.1.1.2 string_unzip	7
4.1.1.3 zip_file_name	7
4.2 code_zippush_server Namespace Reference	8
4.2.1 Variable Documentation	8
4.2.1.1 string_copy	8
4.2.1.2 string_zip	8
4.2.1.3 zip_file_name	8
4.3 compile_and_gen_header Namespace Reference	8
4.3.1 Function Documentation	8
4.3.1.1 call_make_file()	9
4.3.1.2 write_conf_int_header()	9
4.3.2 Variable Documentation	9

4.3.2.1	args	9
4.3.2.2	conf_string	9
4.3.2.3	int_string	9
4.4	masterinteract Namespace Reference	9
4.4.1	Variable Documentation	10
4.4.1.1	f	10
4.4.1.2	file_name	10
4.4.1.3	find	10
4.4.1.4	mpi_flag	10
4.4.1.5	mpi_valid	10
4.4.1.6	numtasks	10
4.4.1.7	runscript	10
4.4.1.8	scriptname	11
4.4.1.9	sn	11
4.4.1.10	str_run_script	11
5	Class Documentation	13
5.1	TAG_SimParams Struct Reference	13
5.1.1	Member Data Documentation	13
5.1.1.1	accur	13
5.1.1.2	deffaccur	14
5.1.1.3	F	14
5.1.1.4	initwidth	14
5.1.1.5	N	14
5.1.1.6	numbtest	14
5.1.1.7	numtasks	14
5.1.1.8	plotpoints	14
5.1.1.9	reset_stepnumb	14
5.1.1.10	setnumb	15
5.1.1.11	simlong	15
5.1.1.12	stepnumb	15

5.1.1.13	testab	15
5.1.1.14	time_step	15
5.2	TransportCoeffs Struct Reference	15
5.2.1	Detailed Description	16
5.2.2	Member Data Documentation	16
5.2.2.1	deff	16
5.2.2.2	meanspd	16
5.2.2.3	meanx	16
5.2.2.4	meanxqub	16
5.2.2.5	meanxsqu	16
5.2.2.6	msd	16
5.2.2.7	mthree	16
5.2.2.8	mu	16
6	File Documentation	17
6.1	code_zippull_server.py File Reference	17
6.2	code_zippush_server.py File Reference	17
6.3	comp_gen_header.h File Reference	18
6.4	compile_and_gen_header.py File Reference	18
6.5	conf_cos.c File Reference	19
6.5.1	Function Documentation	19
6.5.1.1	copycode_conf()	20
6.5.1.2	prfx_conf()	20
6.5.1.3	specs_conf()	20
6.6	conf_cos.h File Reference	20
6.6.1	Macro Definition Documentation	21
6.6.1.1	AMP	21
6.6.1.2	CHECKN	21
6.6.1.3	K_COS	22
6.6.1.4	MAX_HALF_WIDTH	22
6.6.1.5	SQRT_SHIFT	22

6.6.1.6	XMAX	22
6.6.2	Function Documentation	22
6.6.2.1	copycode_conf()	22
6.6.2.2	prfx_conf()	22
6.6.2.3	specs_conf()	22
6.7	conf_sept.c File Reference	23
6.7.1	Function Documentation	23
6.7.1.1	copycode_conf()	23
6.7.1.2	prfx_conf()	23
6.7.1.3	specs_conf()	24
6.8	conf_sept.h File Reference	24
6.8.1	Macro Definition Documentation	25
6.8.1.1	AMP	25
6.8.1.2	MAX_HALF_WIDTH	25
6.8.1.3	R_CONF_SQ	25
6.8.2	Function Documentation	25
6.8.2.1	copycode_conf()	25
6.8.2.2	prfx_conf()	26
6.8.2.3	specs_conf()	26
6.9	conf_splitter.c File Reference	26
6.9.1	Function Documentation	27
6.9.1.1	copycode_conf()	27
6.9.1.2	prfx_conf()	27
6.9.1.3	specs_conf()	27
6.10	conf_splitter.h File Reference	27
6.10.1	Macro Definition Documentation	28
6.10.1.1	Lp	28
6.10.1.2	M	28
6.10.1.3	MAX_HALF_WIDTH	29
6.10.1.4	R_CONF_SQ	29

6.10.1.5	SQRT_SHIFT	29
6.10.2	Function Documentation	29
6.10.2.1	copycode_conf()	29
6.10.2.2	prfx_conf()	29
6.10.2.3	specs_conf()	29
6.11	headers_include.h File Reference	30
6.12	int_hardspheres.c File Reference	30
6.12.1	Function Documentation	31
6.12.1.1	copycode_int()	31
6.12.1.2	prfx_int()	31
6.12.1.3	specs_int()	31
6.12.2	Variable Documentation	31
6.12.2.1	fintx	31
6.12.2.2	fintxpair	32
6.12.2.3	finty	32
6.12.2.4	fintypair	32
6.13	int_hardspheres.h File Reference	32
6.13.1	Macro Definition Documentation	33
6.13.1.1	INT_CUTOFF	33
6.13.1.2	R_INT	33
6.13.2	Function Documentation	33
6.13.2.1	copycode_int()	33
6.13.2.2	prfx_int()	33
6.13.2.3	specs_int()	33
6.13.3	Variable Documentation	34
6.13.3.1	fintx	34
6.13.3.2	fintxpair	34
6.13.3.3	finty	34
6.13.3.4	fintypair	34
6.14	int_lennardjones.c File Reference	34

6.14.1 Function Documentation	35
6.14.1.1 copycode_int()	35
6.14.1.2 prfx_int()	35
6.14.1.3 specs_int()	35
6.14.2 Variable Documentation	35
6.14.2.1 fintx	35
6.14.2.2 fintxpair	36
6.14.2.3 finty	36
6.14.2.4 fintypair	36
6.15 int_lennardjones.h File Reference	36
6.15.1 Macro Definition Documentation	37
6.15.1.1 EPS_L	37
6.15.1.2 INT_CUTOFF	37
6.15.1.3 LJMIN	37
6.15.1.4 LJMINPOW	38
6.15.1.5 LJPREFAC	38
6.15.1.6 R_INT	38
6.15.2 Function Documentation	38
6.15.2.1 copycode_int()	38
6.15.2.2 prfx_int()	38
6.15.2.3 specs_int()	38
6.15.3 Variable Documentation	38
6.15.3.1 fintx	39
6.15.3.2 fintxpair	39
6.15.3.3 finty	39
6.15.3.4 fintypair	39
6.16 main_brownconf.c File Reference	39
6.16.1 Macro Definition Documentation	40
6.16.1.1 MASTER	40
6.16.2 Function Documentation	41

6.16.2.1 adapt_posshifts()	41
6.16.2.2 calc_transpcoeffs()	41
6.16.2.3 calloc_2Ddouble_array()	41
6.16.2.4 calloc_2Dlint_array()	41
6.16.2.5 copy_main()	41
6.16.2.6 delerrorfiles()	42
6.16.2.7 histogramm2d_mpi_reduce()	42
6.16.2.8 histogramm_mpi_reduce()	42
6.16.2.9 init_particle_int()	42
6.16.2.10 init_particle_pos()	43
6.16.2.11 main()	43
6.16.2.12 makedirectory()	43
6.16.2.13 print_hist_countercheck()	43
6.16.2.14 print_muoverf()	43
6.16.2.15 print_positions()	44
6.16.2.16 print_resallthreads()	44
6.16.2.17 print_results_over_time()	44
6.16.2.18 print_runtime()	44
6.16.2.19 print_runtime_threads()	44
6.16.2.20 reset_pos_time()	45
6.16.2.21 update_equcounter()	45
6.16.3 Variable Documentation	45
6.16.3.1 tcoeff	45
6.17 masterinteract.py File Reference	45
6.18 par_sim.c File Reference	46
6.18.1 Function Documentation	46
6.18.1.1 check_parameter_consistency()	46
6.18.1.2 copycode_par()	47
6.18.1.3 init_simparams()	47
6.18.1.4 specs_basic()	47

6.18.1.5 <code>time_step()</code>	47
6.18.2 Variable Documentation	47
6.18.2.1 <code>SimParams</code>	47
6.19 <code>par_sim.h</code> File Reference	48
6.19.1 Macro Definition Documentation	49
6.19.1.1 <code>B</code>	49
6.19.1.2 <code>BOTTRAD</code>	49
6.19.1.3 <code>R_CONF</code>	49
6.19.1.4 <code>RADF</code>	49
6.19.2 Typedef Documentation	49
6.19.2.1 <code>T_SimParams</code>	49
6.19.3 Function Documentation	49
6.19.3.1 <code>check_parameter_consistency()</code>	50
6.19.3.2 <code>copycode_par()</code>	50
6.19.3.3 <code>init_simparams()</code>	50
6.19.3.4 <code>specs_basic()</code>	50
6.19.3.5 <code>time_step()</code>	50
6.19.4 Variable Documentation	50
6.19.4.1 <code>nbin</code>	50
6.19.4.2 <code>SimParams</code>	50

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

dtool_create_header_makefile	??
dtool_pull_code_from_server	??
dtool_push_code_to_server	??
masterinteract	9

Chapter 2

Class Index

2.1 Class List

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

PrintResults	??
TAG_SimParams	13
TransportCoeffs	15

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

comp_gen_header.h	18
conf_cos.c	19
conf_cos.h	20
conf_sept.c	23
conf_sept.h	24
conf_splitter.c	26
conf_splitter.h	27
dtool_create_header_makefile.py	??
dtool_pull_code_from_server.py	??
dtool_push_code_to_server.py	??
headers_include.h	30
int_hardspheres.c	30
int_hardspheres.h	32
int_lennardjones.c	34
int_lennardjones.h	36
main_brownconf.c	39
masterinteract.py	45
par_sim.c	46
par_sim.h	48

Chapter 4

Namespace Documentation

4.1 dtool_create_header_makefile Namespace Reference

Functions

- def `get_module_files` (prefix, message)
- def `get_compiler` (mpicc)
- def `select_file` (list_select, scan_message)
- def `adapt_line` (line, new_part)
- def `adapt_makefile` (compiler, conf_file, int_file)
- def `write_conf_int_header` (conf_file, int_file, mpi_flag)
- def `call_make_file` ()
- def `check_for_mpicc` ()

Variables

- string `list_title_conf` = "\n\nList of possible confinements:\n"
- def `conf_list_files` = `get_module_files`('conf', `list_title_conf`)
- string `request_conf` = "\nchoose module for confinement to be linked in compilation (ind or name):"
- def `conf_file` = `select_file`(`conf_list_files`, `request_conf`)
- string `list_title_int` = "\n\nList of possible interactions:\n"
- def `int_list_files` = `get_module_files`('int', `list_title_int`)
- string `request_int` = "\nchoose module for interaction to be linked in compilation (ind or name):"
- def `int_file` = `select_file`(`int_list_files`, `request_int`)
- def `mpicc_flag` = `check_for_mpicc`()
- def `compiler` = `get_compiler`(`mpicc_flag`)

4.1.1 Detailed Description

Script that helps to create makefile for the compilation of brownian motion code.

It presents the possible files for confinements and interactions and composes the makefile based on the user's choices.

4.1.2 Function Documentation

4.1.2.1 adapt_line()

```
def dtool_create_header_makefile.adapt_line (
    line,
    new_part )
```

4.1.2.2 adapt_makefile()

```
def dtool_create_header_makefile.adapt_makefile (
    compiler,
    conf_file,
    int_file )
```

4.1.2.3 call_make_file()

```
def dtool_create_header_makefile.call_make_file ( )
```

4.1.2.4 check_for_mpicc()

```
def dtool_create_header_makefile.check_for_mpicc ( )
```

4.1.2.5 get_compiler()

```
def dtool_create_header_makefile.get_compiler (
    mpicc )
```

function that contains list of suggested compilers and takes user input to select compiler

4.1.2.6 get_module_files()

```
def dtool_create_header_makefile.get_module_files (
    prefix,
    message )
```

4.1.2.7 select_file()

```
def dtool_create_header_makefile.select_file (
    list_select,
    scan_message )
```

4.1.2.8 write_conf_int_header()

```
def dtool_create_header_makefile.write_conf_int_header (
    conf_file,
    int_file,
    mpi_flag )
```

4.1.3 Variable Documentation

4.1.3.1 compiler

```
def dtool_create_header_makefile.compiler = get\_compiler\(mpicc\_flag\)
```

4.1.3.2 conf_file

```
def dtool_create_header_makefile.conf_file = select\_file\(conf\_list\_files, request\_conf\)
```

4.1.3.3 conf_list_files

```
def dtool_create_header_makefile.conf_list_files = get\_module\_files\('conf', list\_title\_conf\)
```

4.1.3.4 int_file

```
def dtool_create_header_makefile.int_file = select_file(int_list_files, request_int)
```

4.1.3.5 int_list_files

```
def dtool_create_header_makefile.int_list_files = get_module_files('int', list_title_int)
```

4.1.3.6 list_title_conf

```
string dtool_create_header_makefile.list_title_conf = "\n\nList of possible confinements:\n"
```

4.1.3.7 list_title_int

```
string dtool_create_header_makefile.list_title_int = "\n\nList of possible interactions:\n"
```

4.1.3.8 mpicc_flag

```
def dtool_create_header_makefile.mpicc_flag = check_for_mpicc()
```

4.1.3.9 request_conf

```
string dtool_create_header_makefile.request_conf = "\nchoose module for confinement to be  
linked in compilation (ind or name):"
```

4.1.3.10 request_int

```
string dtool_create_header_makefile.request_int = "\nchoose module for interaction to be linked  
in compilation (ind or name):"
```

4.2 dtool_pull_code_from_server Namespace Reference

Functions

- def `scan_input` (message)
- def `read_out_content_server` ()
- def `print_list_index` (print_list, message)
- def `get_name_zip` (zip_selection, action='download')
- def `get_selected_file` (file_selected, list_files)
- def `check_zip_list` (list_zips)
- def `download_selected_zip` (zip_file_name)
- def `unzip_file` (zip_file_name)

Variables

- string `SERVER` = "motthoma@161.116.80.211"
- def `list_code_zips` = `read_out_content_server()`
- def `zip_selected` = `get_name_zip(list_code_zips)`
- def `list_chosen_zips` = `get_selected_file(zip_selected, list_code_zips)`
- def `chosen_zip` = `check_zip_list(list_chosen_zips)`

4.2.1 Detailed Description

Script that can be used to pull code to server.

Functions from this file are also used in file to push code from server.

4.2.2 Function Documentation

4.2.2.1 check_zip_list()

```
def dtool_pull_code_from_server.check_zip_list (
    list_zips )
```

4.2.2.2 download_selected_zip()

```
def dtool_pull_code_from_server.download_selected_zip (
    zip_file_name )
```

4.2.2.3 get_name_zip()

```
def dtool_pull_code_from_server.get_name_zip (
    zip_selection,
    action = 'download' )
```

4.2.2.4 get_selected_file()

```
def dtool_pull_code_from_server.get_selected_file (
    file_selected,
    list_files )
```

4.2.2.5 print_list_index()

```
def dtool_pull_code_from_server.print_list_index (
    print_list,
    message )
```

4.2.2.6 read_out_content_server()

```
def dtool_pull_code_from_server.read_out_content_server ( )
```

4.2.2.7 scan_input()

```
def dtool_pull_code_from_server.scan_input (
    message )
```

4.2.2.8 unzip_file()

```
def dtool_pull_code_from_server.unzip_file (
    zip_file_name )
```

4.2.3 Variable Documentation

4.2.3.1 chosen_zip

```
def dtool_pull_code_from_server.chosen_zip = check_zip_list(list_chosen_zips)
```

4.2.3.2 list_chosen_zips

```
def dtool_pull_code_from_server.list_chosen_zips = get_selected_file(zip_selected, list_code_zips)
```

4.2.3.3 list_code_zips

```
def dtool_pull_code_from_server.list_code_zips = read_out_content_server()
```

4.2.3.4 SERVER

```
string dtool_pull_code_from_server.SERVER = "motthoma@161.116.80.211"
```

4.2.3.5 zip_selected

```
def dtool_pull_code_from_server.zip_selected = get_name_zip(list_code_zips)
```

4.3 dtool_push_code_to_server Namespace Reference

Functions

- def `replace_warning` (`zip_name`, `list_zips`)
- def `create_zip` (`zip_file_name`)
- def `push_zip` (`zip_file_name`)

Variables

- `SERVER` = `pull_code.SERVER`
- `list_code_zips` = `pull_code.read_out_content_server()`
- `zip_name` = `pull_code.get_name_zip(list_code_zips, 'upload')`
- `zip_name_suf` = `zip_name`

4.3.1 Detailed Description

File that can be used to zip code in same directory
and push it to server used for albeniz

Script relies on functions contained in file for
pulling zip files from server.

4.3.2 Function Documentation

4.3.2.1 create_zip()

```
def dtool_push_code_to_server.create_zip (
    zip_file_name )
```

4.3.2.2 push_zip()

```
def dtool_push_code_to_server.push_zip (
    zip_file_name )
```

4.3.2.3 replace_warning()

```
def dtool_push_code_to_server.replace_warning (
    zip_name,
    list_zips )
```

4.3.3 Variable Documentation

4.3.3.1 list_code_zips

```
dtool_push_code_to_server.list_code_zips = pull_code.read_out_content_server()
```

4.3.3.2 SERVER

```
dtool_push_code_to_server.SERVER = pull_code.SERVER
```

4.3.3.3 zip_name

```
dtool_push_code_to_server.zip_name = pull_code.get_name_zip(list_code_zips, 'upload')
```

4.3.3.4 zip_name_suf

```
string dtool_push_code_to_server.zip_name_suf = zip_name
```

4.4 masterinteract Namespace Reference

Variables

- int `f` = 20
- int `sn` = 20
- int `numtasks` = 10
- `str_run_script` = \
- `mpi_valid` = os.popen("which mpicc").read()
- int `mpi_flag` = 0
- int `find` = f
- string `scriptname` = "LJ_sn_{0}_F_{1:.2f}".format(sn,f)
- `file_name` = open(scriptname, "w")
- string `runscript` = 'qsub ' + scriptname

4.4.1 Variable Documentation

4.4.1.1 f

```
int masterinteract.f = 20
```

4.4.1.2 file_name

```
masterinteract.file_name = open(scriptname, "w")
```

4.4.1.3 find

```
int masterinteract.find = f
```

4.4.1.4 mpi_flag

```
int masterinteract.mpi_flag = 0
```

4.4.1.5 mpi_valid

```
masterinteract.mpi_valid = os.popen("which mpicc").read()
```

4.4.1.6 numtasks

```
int masterinteract.numtasks = 10
```

4.4.1.7 runscript

```
string masterinteract.runscript = 'qsub ' + scriptname
```

4.4.1.8 scriptname

```
string masterinteract.scriptname = "LJ_sn_{0}_F_{1:.2f}".format(sn,f)
```

4.4.1.9 sn

```
int masterinteract.sn = 20
```

4.4.1.10 str_run_script

```
masterinteract.str_run_script = \
```

Chapter 5

Class Documentation

5.1 PrintResults Struct Reference

Public Attributes

- int `state`
- char `fname` [60]
- char `fnamemom` [60]

5.1.1 Detailed Description

structure which contains variables used in the printing functions which print the resulting transport coefficients in .dat files

5.1.2 Member Data Documentation

5.1.2.1 fname

```
char PrintResults::fname[60]
```

5.1.2.2 fnamemom

```
char PrintResults::fnamemom[60]
```

5.1.2.3 state

```
int PrintResults::state
```

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

- [main_brownconf.c](#)

5.2 TAG_SimParams Struct Reference

```
#include <par_sim.h>
```

Public Attributes

- int [N](#)
- int [setnumb](#)
- int [numbtest](#)
- int [stepnumb](#)
- int [simlong](#)
- double [accur](#)
- double [deffaccur](#)
- double [initwidth](#)
- double [F](#)
- int [plotpoints](#)
- int [testab](#)
- int [reset_stepnumb](#)
- double [time_step](#)
- int [numtasks](#)

5.2.1 Member Data Documentation

5.2.1.1 [accur](#)

```
double TAG_SimParams::accur
```

5.2.1.2 [deffaccur](#)

```
double TAG_SimParams::deffaccur
```

5.2.1.3 F

```
double TAG_SimParams::F
```

5.2.1.4 initwidth

```
double TAG_SimParams::initwidth
```

5.2.1.5 N

```
int TAG_SimParams::N
```

5.2.1.6 numbtest

```
int TAG_SimParams::numbtest
```

5.2.1.7 numtasks

```
int TAG_SimParams::numtasks
```

5.2.1.8 plotpoints

```
int TAG_SimParams::plotpoints
```

5.2.1.9 reset_stepnumb

```
int TAG_SimParams::reset_stepnumb
```

5.2.1.10 setnumb

```
int TAG_SimParams::setnumb
```

5.2.1.11 simlong

```
int TAG_SimParams::simlong
```

5.2.1.12 stepnumb

```
int TAG_SimParams::stepnumb
```

5.2.1.13 testab

```
int TAG_SimParams::testab
```

5.2.1.14 time_step

```
double TAG_SimParams::time_step
```

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

- [par_sim.h](#)

5.3 TransportCoeffs Struct Reference

Public Attributes

- long double [meanx](#)
- double [meanspeed](#)
- long double [meanssqu](#)
- long double [meanxqub](#)
- long double [msd](#)
- long double [mthree](#)
- double [deff](#)
- double [mu](#)

5.3.1 Detailed Description

structure which contains transport coefficients that are calculated within individual threads.

5.3.2 Member Data Documentation

5.3.2.1 deff

```
double TransportCoeffs::deff
```

5.3.2.2 meanspeed

```
double TransportCoeffs::meanspeed
```

5.3.2.3 meanx

```
long double TransportCoeffs::meanx
```

5.3.2.4 meanxqub

```
long double TransportCoeffs::meanxqub
```

5.3.2.5 meanxsqu

```
long double TransportCoeffs::meanxsqu
```

5.3.2.6 msd

```
long double TransportCoeffs::msd
```

5.3.2.7 mthree

```
long double TransportCoeffs::mthree
```

5.3.2.8 mu

```
double TransportCoeffs::mu
```

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

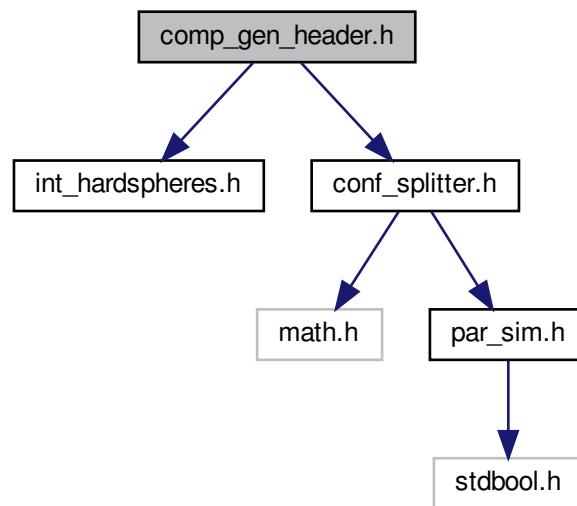
- [main_brownconf.c](#)

Chapter 6

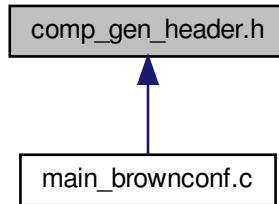
File Documentation

6.1 comp_gen_header.h File Reference

```
#include "int_hardspheres.h"
#include "conf_splitter.h"
Include dependency graph for comp_gen_header.h:
```

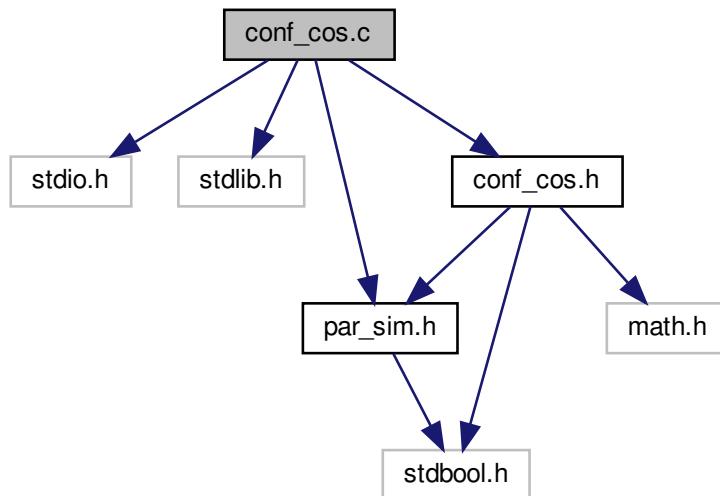


This graph shows which files directly or indirectly include this file:



6.2 conf_cos.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "par_sim.h"
#include "conf_cos.h"
Include dependency graph for conf_cos.c:
```



Functions

- void [specs_conf](#) (double binx, double biny, double bin2d)
- void [copycode_conf](#) ()
- char * [prfx_conf](#) ()

6.2.1 Function Documentation

6.2.1.1 copycode_conf()

```
void copycode_conf ( )
```

6.2.1.2 prfx_conf()

```
char* prfx_conf ( )
```

function to provide prefix in name of working directory

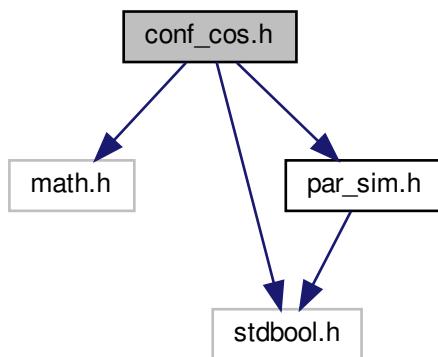
6.2.1.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

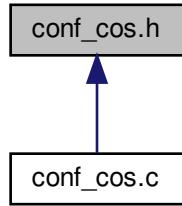
function to write confinement specific information in specs file

6.3 conf_cos.h File Reference

```
#include <math.h>
#include <stdbool.h>
#include "par_sim.h"
Include dependency graph for conf_cos.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define AMP (1.0/(2.0*M_PI))
- #define MAX_HALF_WIDTH (2*AMP+B)
- #define CHECKN 5
- #define K_COS (2.0*M_PI/L)
- #define XMAX (L/4.0-R_CONF)
- #define SQRT_SHIFT R_CONF*sqrt(1+AMP*AMP*K_COS*K_COS)

Functions

- void specs_conf (double binx, double biny, double bin2d)
- void copycode_conf ()
- char * prfx_conf ()

6.3.1 Macro Definition Documentation

6.3.1.1 AMP

```
#define AMP (1.0/(2.0*M_PI))
```

ensure that only one header for spatial confinement is included in code

6.3.1.2 CHECKN

```
#define CHECKN 5
```

6.3.1.3 K_COS

```
#define K_COS (2.0*M_PI/L)
```

6.3.1.4 MAX_HALF_WIDTH

```
#define MAX_HALF_WIDTH (2*AMP+B)
```

6.3.1.5 SQRT_SHIFT

```
#define SQRT_SHIFT R_CONF*sqrt(1+AMP*AMP*K_COS*K_COS)
```

6.3.1.6 XMAX

```
#define XMAX (L/4.0-R_CONF)
```

6.3.2 Function Documentation

6.3.2.1 copycode_conf()

```
void copycode_conf ( )
```

6.3.2.2 prfx_conf()

```
char* prfx_conf ( )
```

function to provide prefix in name of working directory

function that provides name of confinement

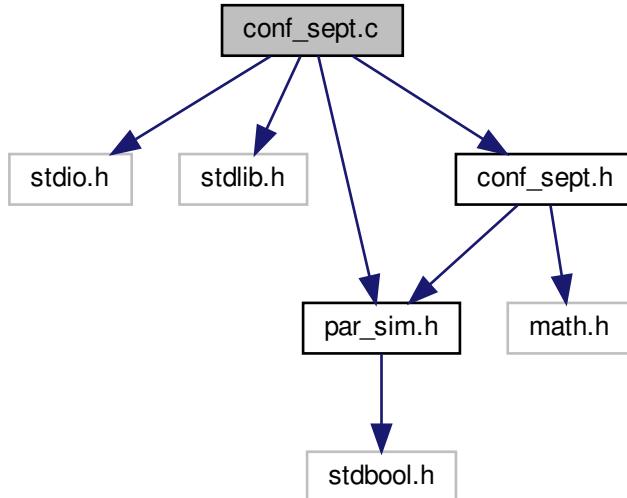
6.3.2.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

function to write confinement specific information in specs file

6.4 conf_sept.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "par_sim.h"
#include "conf_sept.h"
Include dependency graph for conf_sept.c:
```



Functions

- void [specs_conf](#) (double binx, double biny, double bin2d)
- void [copycode_conf](#) ()
- char * [prfx_conf](#) ()

6.4.1 Function Documentation

6.4.1.1 [copycode_conf\(\)](#)

```
void copycode_conf ( )
```

6.4.1.2 [prfx_conf\(\)](#)

```
char* prfx_conf ( )
```

function to provide prefix in name of working directory

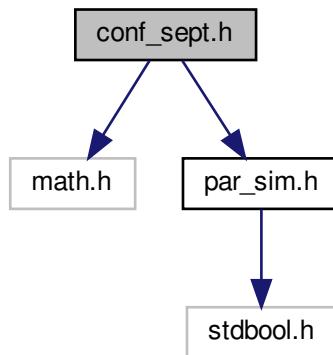
6.4.1.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

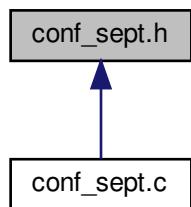
function to write confinement specific information in specs file

6.5 conf_sept.h File Reference

```
#include <math.h>
#include "par_sim.h"
Include dependency graph for conf_sept.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define AMP 1.0
- #define MAX_HALF_WIDTH (AMP+B)
- #define R_CONF_SQ R_CONF*R_CONF

Functions

- void specs_conf (double binx, double biny, double bin2d)
- void copycode_conf ()
- char * prfx_conf ()

6.5.1 Macro Definition Documentation

6.5.1.1 AMP

```
#define AMP 1.0
```

ensure that only one header for spatial confinement is included in code

6.5.1.2 MAX_HALF_WIDTH

```
#define MAX_HALF_WIDTH (AMP+B)
```

6.5.1.3 R_CONF_SQ

```
#define R_CONF_SQ R_CONF*R_CONF
```

6.5.2 Function Documentation

6.5.2.1 copycode_conf()

```
void copycode_conf ( )
```

6.5.2.2 prfx_conf()

```
char* prfx_conf ( )
```

function to provide prefix in name of working directory

function that provides name of confinement

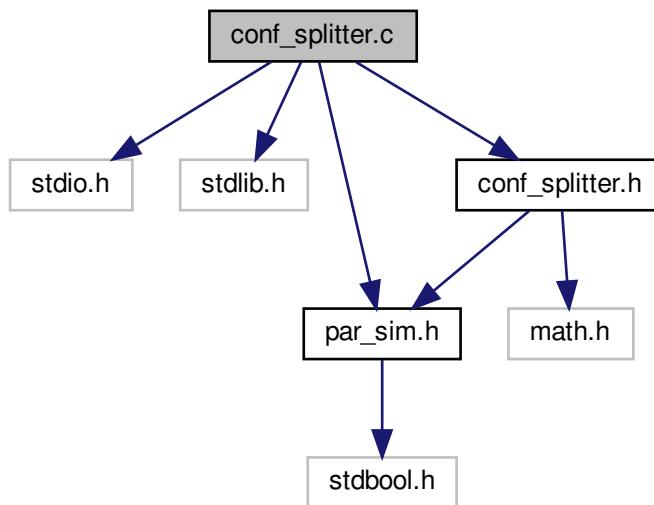
6.5.2.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

function to write confinement specific information in specs file

6.6 conf_splitter.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include "par_sim.h"
#include "conf_splitter.h"
Include dependency graph for conf_splitter.c:
```



Functions

- void [specs_conf](#) (double binx, double biny, double bin2d)
- void [copycode_conf](#) ()
- char * [prfx_conf](#) ()

6.6.1 Function Documentation

6.6.1.1 copycode_conf()

```
void copycode_conf ( )
```

6.6.1.2 prfx_conf()

```
char* prfx_conf ( )
```

function that provides name of confinement

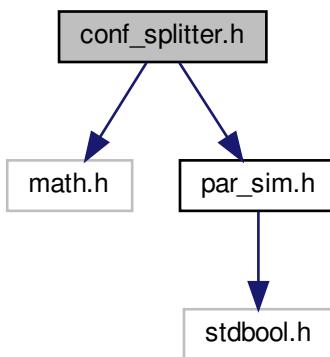
6.6.1.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

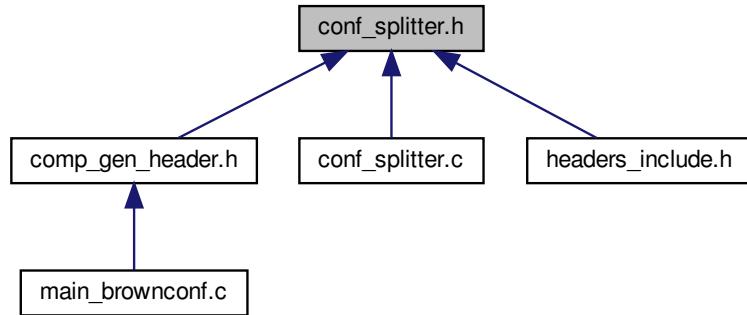
function to write confinement specific information in specs file

6.7 conf_splitter.h File Reference

```
#include <math.h>
#include "par_sim.h"
Include dependency graph for conf_splitter.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- `#define M 0.9`
- `#define MAX_HALF_WIDTH (M*L + B)`
- `#define SQRT_SHIFT R_CONF*sqrt(1 + M*M)`
- `#define R_CONF_SQ R_CONF*R_CONF`
- `#define Lp (L - (R_CONF*M)/(sqrt(1 + M*M)))`

Functions

- `void specs_conf (double binx, double biny, double bin2d)`
- `void copycode_conf ()`
- `char * prfx_conf ()`

6.7.1 Macro Definition Documentation

6.7.1.1 Lp

```
#define Lp (L - (R_CONF*M) / (sqrt(1 + M*M)))
```

6.7.1.2 M

```
#define M 0.9
```

ensure that only one header for spatial confinement is included in code

6.7.1.3 MAX_HALF_WIDTH

```
#define MAX_HALF_WIDTH (M*L + B)
```

6.7.1.4 R_CONF_SQ

```
#define R_CONF_SQ R_CONF*R_CONF
```

6.7.1.5 SQRT_SHIFT

```
#define SQRT_SHIFT R_CONF*sqrt(1 + M*M)
```

6.7.2 Function Documentation

6.7.2.1 copycode_conf()

```
void copycode_conf ( )
```

6.7.2.2 prfx_conf()

```
char* prfx_conf ( )
```

function to provide prefix in name of working directory

function that provides name of confinement

6.7.2.3 specs_conf()

```
void specs_conf (
    double binx,
    double biny,
    double bin2d )
```

function to write confinement specific information in specs file

6.8 dtool_create_header_makefile.py File Reference

Namespaces

- [dtool_create_header_makefile](#)

Functions

- def [dtool_create_header_makefile.get_module_files](#) (prefix, message)
- def [dtool_create_header_makefile.get_compiler](#) (mpicc)
- def [dtool_create_header_makefile.select_file](#) (list_select, scan_message)
- def [dtool_create_header_makefile.adapt_line](#) (line, new_part)
- def [dtool_create_header_makefile.adapt_makefile](#) (compiler, conf_file, int_file)
- def [dtool_create_header_makefile.write_conf_int_header](#) (conf_file, int_file, mpi_flag)
- def [dtool_create_header_makefile.call_make_file](#) ()
- def [dtool_create_header_makefile.check_for_mpicc](#) ()

Variables

- string [dtool_create_header_makefile.list_title_conf](#) = "\n\nList of possible confinements:\n"
- def [dtool_create_header_makefile.conf_list_files](#) = get_module_files('conf', list_title_conf)
- string [dtool_create_header_makefile.request_conf](#) = "\nchoose module for confinement to be linked in compilation (ind or name):"
- def [dtool_create_header_makefile.conf_file](#) = select_file(conf_list_files, request_conf)
- string [dtool_create_header_makefile.list_title_int](#) = "\n\nList of possible interactions:\n"
- def [dtool_create_header_makefile.int_list_files](#) = get_module_files('int', list_title_int)
- string [dtool_create_header_makefile.request_int](#) = "\nchoose module for interaction to be linked in compilation (ind or name):"
- def [dtool_create_header_makefile.int_file](#) = select_file(int_list_files, request_int)
- def [dtool_create_header_makefile.mpicc_flag](#) = check_for_mpicc()
- def [dtool_create_header_makefile.compiler](#) = get_compiler(mpicc_flag)

6.9 dtool_pull_code_from_server.py File Reference

Namespaces

- [dtool_pull_code_from_server](#)

Functions

- def [dtool_pull_code_from_server.scan_input](#) (message)
- def [dtool_pull_code_from_server.read_out_content_server](#) ()
- def [dtool_pull_code_from_server.print_list_index](#) (print_list, message)
- def [dtool_pull_code_from_server.get_name_zip](#) (zip_selection, action='download')
- def [dtool_pull_code_from_server.get_selected_file](#) (file_selected, list_files)
- def [dtool_pull_code_from_server.check_zip_list](#) (list_zips)
- def [dtool_pull_code_from_server.download_selected_zip](#) (zip_file_name)
- def [dtool_pull_code_from_server.unzip_file](#) (zip_file_name)

Variables

- string `dtool_pull_code_from_server.SERVER` = "motthoma@161.116.80.211"
- def `dtool_pull_code_from_server.list_code_zips` = `read_out_content_server()`
- def `dtool_pull_code_from_server.zip_selected` = `get_name_zip(list_code_zips)`
- def `dtool_pull_code_from_server.list_chosen_zips` = `get_selected_file(zip_selected, list_code_zips)`
- def `dtool_pull_code_from_server.chosen_zip` = `check_zip_list(list_chosen_zips)`

6.10 dtool_push_code_to_server.py File Reference

Namespaces

- `dtool_push_code_to_server`

Functions

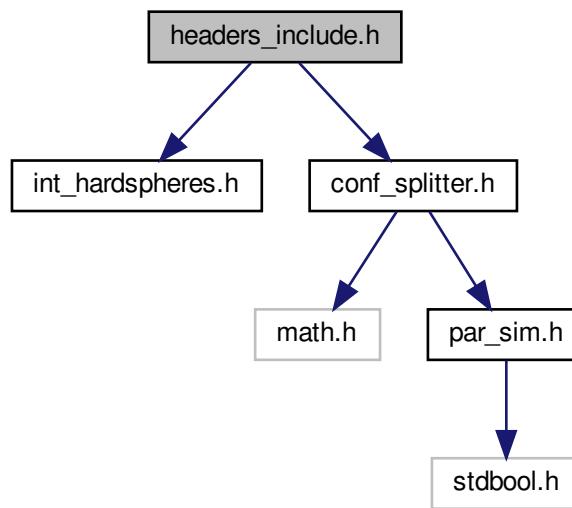
- def `dtool_push_code_to_server.replace_warning` (`zip_name, list_zips`)
- def `dtool_push_code_to_server.create_zip` (`zip_file_name`)
- def `dtool_push_code_to_server.push_zip` (`zip_file_name`)

Variables

- `dtool_push_code_to_server.SERVER` = `pull_code.SERVER`
- `dtool_push_code_to_server.list_code_zips` = `pull_code.read_out_content_server()`
- `dtool_push_code_to_server.zip_name` = `pull_code.get_name_zip(list_code_zips, 'upload')`
- `dtool_push_code_to_server.zip_name_suf` = `zip_name`

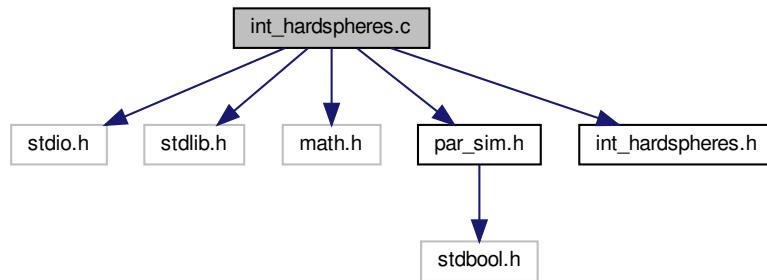
6.11 headers_include.h File Reference

```
#include "int_hardspheres.h"
#include "conf_splitter.h"
Include dependency graph for headers_include.h:
```



6.12 int_hardspheres.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "par_sim.h"
#include "int_hardspheres.h"
Include dependency graph for int_hardspheres.c:
```



Functions

- void [specs_int](#) (double f_cut)
- void [copycode_int](#) ()
- char * [prfx_int](#) ()

Variables

- double [fintx](#) = 0
- double [finty](#) = 0
- double [fintxpairs](#) = 0
- double [fintypairs](#) = 0

6.12.1 Function Documentation

6.12.1.1 copycode_int()

```
void copycode_int ( )
```

6.12.1.2 prfx_int()

```
char* prfx_int ( )
```

6.12.1.3 specs_int()

```
void specs_int (
    double f_cut )
```

6.12.2 Variable Documentation

6.12.2.1 fintx

```
double fintx = 0
```

6.12.2.2 fintxpair

```
double fintxpair = 0
```

6.12.2.3 finty

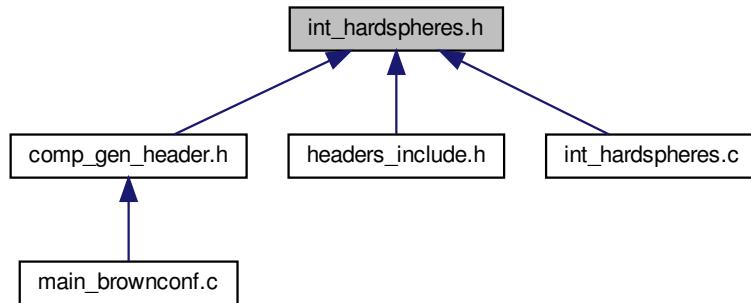
```
double finty = 0
```

6.12.2.4 fintypair

```
double fintypair = 0
```

6.13 int_hardspheres.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- `#define R_INT (1.0*R_CONF)`
- `#define INT_CUTOFF 0`

Functions

- `void specs_int (double f_cut)`
- `void copycode_int ()`
- `char * prfx_int ()`

Variables

- `double fintx`
- `double finty`
- `double fintxpair`
- `double fintypair`

6.13.1 Macro Definition Documentation

6.13.1.1 INT_CUTOFF

```
#define INT_CUTOFF 0
```

6.13.1.2 R_INT

```
#define R_INT (1.0*R_CONF)
```

ensure that only one header for intra- particle interaction is included in code

6.13.2 Function Documentation

6.13.2.1 copycode_int()

```
void copycode_int ( )
```

6.13.2.2 prfx_int()

```
char* prfx_int ( )
```

6.13.2.3 specs_int()

```
void specs_int (
    double f_cut )
```

function for the calculation of the intra-particle force arising from a Lennard-Jones Potential. The potential is described by two parameters, position of minimum LJ_MIN and depth of minimum LJ_EPS. The potential is truncated at INT_CUTOFF.

6.13.3 Variable Documentation

6.13.3.1 fintx

```
double fintx
```

6.13.3.2 fintxpair

```
double fintxpair
```

6.13.3.3 finty

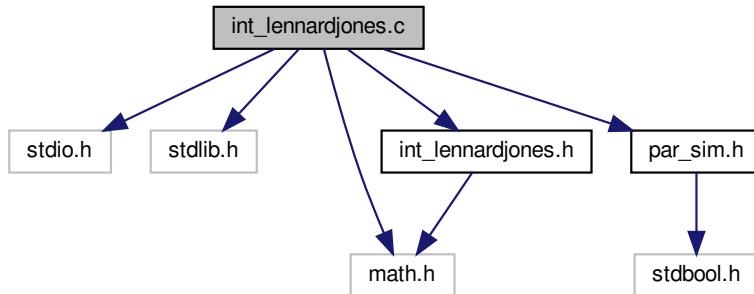
```
double finty
```

6.13.3.4 fintypair

```
double fintypair
```

6.14 int_lennardjones.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "par_sim.h"
#include "int_lennardjones.h"
Include dependency graph for int_lennardjones.c:
```



Functions

- void [specs_int](#) (double f_cut)
- void [copycode_int](#) ()
- char * [prfx_int](#) ()

Variables

- double [fintx](#) = 0
- double [finty](#) = 0
- double [fintxpair](#) = 0
- double [fintypair](#) = 0

6.14.1 Function Documentation

6.14.1.1 copycode_int()

```
void copycode_int ( )
```

6.14.1.2 prfx_int()

```
char* prfx_int ( )
```

6.14.1.3 specs_int()

```
void specs_int (
    double f_cut )
```

function for the calculation of the intra-particle force arising from a Lennard-Jones Potential. The potential is described by two parameters, position of minimum LJ_MIN and depth of minimum LJ_EPS. The potential is truncated at INT_CUTOFF.

6.14.2 Variable Documentation

6.14.2.1 fintx

```
double fintx = 0
```

6.14.2.2 fintxpair

```
double fintxpair = 0
```

6.14.2.3 finty

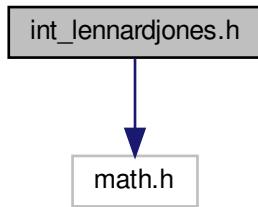
```
double finty = 0
```

6.14.2.4 fintypair

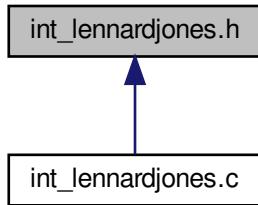
```
double fintypair = 0
```

6.15 int_lennardjones.h File Reference

```
#include <math.h>
Include dependency graph for int_lennardjones.h:
```



This graph shows which files directly or indirectly include this file:



Macros

- #define R_INT (R_CONF)
- #define INT_CUTOFF (0.2*L)
- #define EPS_L (1.0)
- #define LJMIN (0.5*B)
- #define LJMINPOW (pow(LJMIN,6))
- #define LJPREFAC 12.0*EPS_L*LJMINPOW

Functions

- void `specs_int` (double `f_cut`)
- void `copycode_int` ()
- char * `prfx_int` ()

Variables

- double `fintx`
- double `finty`
- double `fintxpair`
- double `fintypair`

6.15.1 Macro Definition Documentation

6.15.1.1 EPS_L

```
#define EPS_L (1.0)
```

6.15.1.2 INT_CUTOFF

```
#define INT_CUTOFF (0.2*L)
```

6.15.1.3 LJMIN

```
#define LJMIN (0.5*B)
```

6.15.1.4 LJMINPOW

```
#define LJMINPOW (pow(LJMIN, 6))
```

6.15.1.5 LJPREFAC

```
#define LJPREFAC 12.0*EPS_L*LJMINPOW
```

6.15.1.6 R_INT

```
#define R_INT (R_CONF)
```

ensure that only one header for intra- particle interaction is included in code

6.15.2 Function Documentation

6.15.2.1 copycode_int()

```
void copycode_int ( )
```

6.15.2.2 prfx_int()

```
char* prfx_int ( )
```

6.15.2.3 specs_int()

```
void specs_int (
    double f_cut )
```

function for the calculation of the intra-particle force arising from a Lennard-Jones Potential. The potential is described by two parameters, position of minimum LJ_MIN and depth of minimum LJ_EPS. The potential is truncated at INT_CUTOFF.

6.15.3 Variable Documentation

6.15.3.1 fintx

```
double fintx
```

6.15.3.2 fintxpair

```
double fintxpair
```

6.15.3.3 finty

```
double finty
```

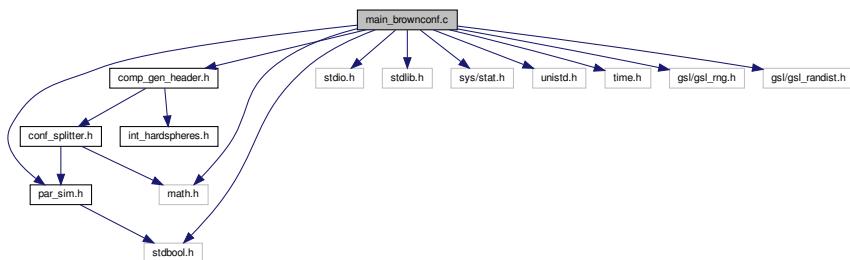
6.15.3.4 fintypair

```
double fintypair
```

6.16 main_brownconf.c File Reference

```
#include "par_sim.h"
#include "comp_gen_header.h"
#include <stdio.h>
#include <stdlib.h>
#include <sys/stat.h>
#include <unistd.h>
#include <math.h>
#include <time.h>
#include <stdbool.h>
#include <gsl/gsl_rng.h>
#include <gsl/gsl_randist.h>
```

Include dependency graph for main_brownconf.c:



Classes

- struct [TransportCoeffs](#)
- struct [PrintResults](#)

Macros

- #define [MASTER](#) 0

Functions

- void `calc_transpcoeffs` (int setn_per_task, double t, long int **posshift, long int **negshift, double **posx, double **x_init)
- char * `makedirectory` (char *confprfx, char *intprfx)
- void `copy_main` ()
- void `delerrorfiles` ()
- void `print_runtime` (clock_t start)
- void `print_positions` (int m, double **posx, double **posy)
- void `print_runtime_threads` (clock_t start, int numtasks, int taskid)
- void `print_results_over_time` (double t, int abb, int abbdeff)
- void `print_resallthreads` (long double msdall, double meanspeedall, double muall, double deffall, long double meanxall, long double meanxsquall, long double mthreeall)
- void `print_muoverf` (double muall, double deffall, char *namefile)
- int `histogramm_mpi_reduce` (int m, double backshift, double length, double bin, double **positions, char *fname, int taskid)
- int `histogramm2d_mpi_reduce` (int m, double bin2d, double **positionsx, double **positionsy, char *fname, int taskid)
- void `init_particle_pos` (int setn_per_task, double **positionx, double **positiony, double ***xstart, gsl_rng *r)
- void `init_particle_int` (int setn_per_task, double **positionx, double **positiony, double **fintxarray, double **fintyarray)
- void `print_hist_countercheck` (int xcheck, int ycheck, int twodcheck, char *fname_specs)
- double `reset_pos_time` (int setn_per_task, long int **posshift, long int **negshift)
- void `adapt_posshifts` (int shiftind, int i, int j, long int **posshift, long int **negshift)
- int `update_eucounter` (double tran_quant, double tran_quanto, double accuracy, int eucounter)
- double ** `calloc_2Ddouble_array` (int m, int n)
- long int ** `calloc_2Dlint_array` (int m, int n)
- int `main` (int argc, char **argv)

Variables

- struct `TransportCoeffs` tcoeff
- struct `PrintResults` printres

6.16.1 Macro Definition Documentation

6.16.1.1 MASTER

```
#define MASTER 0
```

6.16.2 Function Documentation

6.16.2.1 adapt_posshifts()

```
void adapt_posshifts (
    int shiftind,
    int i,
    int j,
    long int ** possshift,
    long int ** negshift )
```

function to update shifts which are monitored to calculate absolute position in x-direction from sitcoeff.mulation with cyclic boundary conditions

6.16.2.2 calc_transpcoeffs()

```
void calc_transpcoeffs (
    int setn_per_task,
    double t,
    long int ** possshift,
    long int ** negshift,
    double ** posx,
    double ** x_init )
```

Function that caluclates transport coefficients such as mobility, means-squared displacement or thir moment of position. Function takes relative x-positions and values of shift to first call function for calculation of individual absolute particle positions and subsequently ensemble averages of the first three moments of the position.

6.16.2.3 calloc_2Ddouble_array()

```
double** calloc_2Ddouble_array (
    int m,
    int n )
```

Function that allocates memory for a 2 dimensional array of doubles

6.16.2.4 calloc_2Dlint_array()

```
long int** calloc_2Dlint_array (
    int m,
    int n )
```

Function that allocates memory for a 2 dimensional array of long ints

6.16.2.5 copy_main()

```
void copy_main ( )
```

copies main_brownianconf.c to directory created by 'makedirectory'

6.16.2.6 delerrorfiles()

```
void delerrorfiles ( )
```

deletes error and log files used on albeniz

6.16.2.7 histogramm2d_mpi_reduce()

```
int histogramm2d_mpi_reduce (
    int m,
    double bin2d,
    double ** positionsx,
    double ** positionsy,
    char * fname,
    int taskid )
```

function that stores 2 dimensional histogram in file named fname. during a spatial scan over all nx*ny rectangular fields, all particle positions are checked and the number of particles in the respective field is increased if a particle is detected. The number counter of all particles in the field is stored in the file together with the upper and lower boundaries in both directions of the field.

6.16.2.8 histogramm_mpi_reduce()

```
int histogramm_mpi_reduce (
    int m,
    double backshift,
    double length,
    double bin,
    double ** positions,
    char * fname,
    int taskid )
```

function that stores 1 dimensional histogram in file named fname. during a spatial sweep through all length/bin slices from position 'backshift' to position 'length', all particle positions are checked and the number of particles in the respective slice is increased if a particle is detected. The number counter of all particles in the slice is stored in the file together with the upper and lower boundary of the slice.

6.16.2.9 init_particle_int()

```
void init_particle_int (
    int setn_per_task,
    double ** positionx,
    double ** positiony,
    double ** fintxarray,
    double ** fintyarray )
```

6.16.2.10 init_particle_pos()

```
void init_particle_pos (
    int setn_per_task,
    double ** positionx,
    double ** positiony,
    double ** xstart,
    gsl_rng * r )
```

6.16.2.11 main()

```
int main (
    int argc,
    char ** argv )
```

main function of Brownian motion simulation

calculate number of samples of interacting particles

calculate number of samples of interacting particles per task

6.16.2.12 makedirectory()

```
char* makedirectory (
    char * confprfx,
    char * intprfx )
```

creates directory, where code and data is transferred to moves to this directory as working directory

0700 is modus for determining access rights to dir

6.16.2.13 print_hist_countercheck()

```
void print_hist_countercheck (
    int xcheck,
    int ycheck,
    int twodcheck,
    char * fname_specs )
```

Function that prints the result of the counter checks available from the histogram functions. There, the total number of particles is counted in order to check if all particles are represented in the histogram and situated in the confinement.

6.16.2.14 print_muoverf()

```
void print_muoverf (
    double muall,
    double deffall,
    char * namefile )
```

prints results to file outside of working directory

6.16.2.15 print_positions()

```
void print_positions (
    int m,
    double ** posx,
    double ** posy )
```

prints particle positions to file

6.16.2.16 print_resallthreads()

```
void print_resallthreads (
    long double msdall,
    double meanspeedall,
    double muall,
    double deffall,
    long double meanxall,
    long double meansquall,
    long double mthreeall )
```

prints results of simulation to file at the end of the simulation

6.16.2.17 print_results_over_time()

```
void print_results_over_time (
    double t,
    int abb,
    int abbdeff )
```

function to print online results of sitcoeff.mulation over time

6.16.2.18 print_runtime()

```
void print_runtime (
    clock_t start )
```

prints run time of code into file

6.16.2.19 print_runtime_threads()

```
void print_runtime_threads (
    clock_t start,
    int numtasks,
    int taskid )
```

prints individual runtime of each thread

measure program run time of task

6.16.2.20 reset_pos_time()

```
double reset_pos_time (
    int setn_per_task,
    long int ** posshift,
    long int ** negshift )
```

Function to reset the sitcoeff.mulated system time t and the positions to originate values in order to skip transient effects from start of sitcoeff.mulation.

6.16.2.21 update_eqcounter()

```
int update_eqcounter (
    double tran_quant,
    double tran_quanto,
    double accuracy,
    int eqcounter )
```

function for the update of the counter that is used to monitore the equilibration of the system. A transport quantity tran_quant is compared with a previous value. If the difference is below the demanded accuracy, the value of the counter is increased. If the difference is larger than twice the accuracy the counter is decreased

6.16.3 Variable Documentation

6.16.3.1 printres

```
struct PrintResults printres
```

6.16.3.2 tcoeff

```
struct TransportCoeffs tcoeff
```

6.17 masterinteract.py File Reference

Namespaces

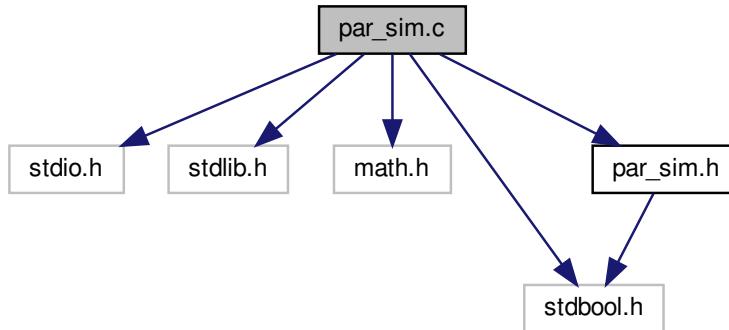
- [masterinteract](#)

Variables

- int `masterinteract.f` = 20
- int `masterinteract.sn` = 20
- int `masterinteract.numtasks` = 10
- `masterinteract.str_run_script` = \
- `masterinteract.mpi_valid` = os.popen("which mpicc").read()
- int `masterinteract.mpi_flag` = 0
- int `masterinteract.find` = f
- string `masterinteract.scriptname` = "LJ_sn_{0}_F_{1:.2f}".format(sn,f)
- `masterinteract.file_name` = open(scriptname, "w")
- string `masterinteract.runscript` = 'qsub ' + scriptname

6.18 par_sim.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>
#include "par_sim.h"
Include dependency graph for par_sim.c:
```



Functions

- void `init_simparams ()`
- bool `check_parameter_consistency ()`
- double `time_step (double lscale_conf, double lscale_part)`
- void `specs_basic (char *fnamespec)`
- void `copycode_par ()`

Variables

- `T_SimParams SimParams`

6.18.1 Function Documentation

6.18.1.1 check_parameter_consistency()

```
bool check_parameter_consistency ( )
```

6.18.1.2 copycode_par()

```
void copycode_par ( )
```

function that copies module in working directory

6.18.1.3 init_simparams()

```
void init_simparams ( )
```

Function that initializes hard coded parameters of simulation

6.18.1.4 specs_basic()

```
void specs_basic (
    char * fnamespec )
```

function used to read simulation parameters in specs file

6.18.1.5 time_step()

```
double time_step (
    double lscale_conf,
    double lscale_part )
```

Function that initializes value of simulation time steps. Time steps are calculated by means of confinement and particle quantities. Therefore, the function has to be called in main, where header files for confinement and particle-particle interactions are included.

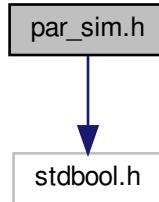
6.18.2 Variable Documentation

6.18.2.1 SimParams

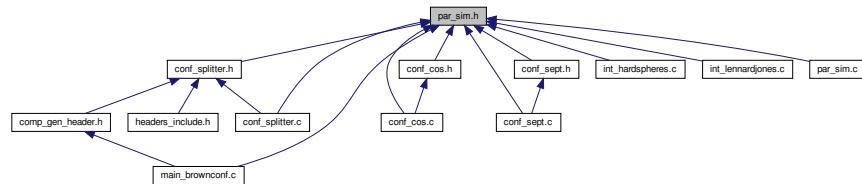
[T_SimParams](#) SimParams

6.19 par_sim.h File Reference

```
#include <stdbool.h>
Include dependency graph for par_sim.h:
```



This graph shows which files directly or indirectly include this file:



Classes

- struct [TAG_SimParams](#)

Macros

- #define [B](#) 0.1
- #define [RADF](#) 0.5
- #define [R_CONF](#) ([RADF](#)*[B](#))
- #define [BOTTRAD](#) (([RADF](#) == 0 ? 1.0 : (1.0/[RADF](#))))

Typedefs

- typedef struct [TAG_SimParams](#) [T_SimParams](#)

Functions

- double [time_step](#) (double [lscale_conf](#), double [lscale_part](#))
- void [init_simparams](#) ()
- bool [check_parameter_consistency](#) ()
- void [specs_basic](#) (char *fnamespec)
- void [copycode_par](#) ()

Variables

- unsigned int `nbin`
- `T_SimParams SimParams`

6.19.1 Macro Definition Documentation

6.19.1.1 `B`

```
#define B 0.1
```

6.19.1.2 `BOTTRAD`

```
#define BOTTRAD ((RADF == 0 ? 1.0 : (1.0/RADF)))
```

6.19.1.3 `R_CONF`

```
#define R_CONF (RADF*B)
```

6.19.1.4 `RADF`

```
#define RADF 0.5
```

6.19.2 Typedef Documentation

6.19.2.1 `T_SimParams`

```
typedef struct TAG_SimParams T_SimParams
```

6.19.3 Function Documentation

6.19.3.1 check_parameter_consistency()

```
bool check_parameter_consistency ( )
```

6.19.3.2 copycode_par()

```
void copycode_par ( )
```

function that copies module in working directory

6.19.3.3 init_simparams()

```
void init_simparams ( )
```

Function that initializes hard coded parameters of simulation

6.19.3.4 specs_basic()

```
void specs_basic (
    char * fnamespec )
```

function used to read simulation parameters in specs file

6.19.3.5 time_step()

```
double time_step (
    double lscale_conf,
    double lscale_part )
```

Function that initializes value of simulation time steps. Time steps are calculated by means of confinement and particle quantities. Therefore, the function has to be called in main, where header files for confinement and particle-particle interactions are included.

6.19.4 Variable Documentation**6.19.4.1 nbin**

```
unsigned int nbin
```

6.19.4.2 SimParams

[T_SimParams](#) SimParams
