My Project

Generated by Doxygen 1.8.15

Chapter 1

Drawing Shapes

This project helps user to draw shapes. Currently two types of shapes can be drawn:

- How to draw rectangle?
- How to draw circle?

Chapter 2

README

Presentation of this page. Very nice code, you will like it.

Chapter 3

How to draw rectangle?

Lorem ipsum dolor sit amet

Chapter 4

How to draw circle?

This page is about how to draw a circle. Following sections describe circle:

- Definition of Circle
- Circle Class

Chapter 5

Hierarchical Index

5.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 6

Class Index

6.1 Class List

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

Chapter 7

Class Documentation

7.1 Array2D< T, W, H > Class Template Reference

Public Member Functions

- T **at** (unsigned int x, unsigned int y)
- void **set** (unsigned int x, unsigned int y, float val)
- void **initialize** (T value)
- void **add** (unsigned int x, unsigned int y, T val)

Public Attributes

- const int **width** = W
- const int **height** = H

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

- include/Tools.h

7.2 Array3D< T, W, H, D > Class Template Reference

Public Member Functions

- T **at** (unsigned int x, unsigned int y, unsigned int z)
- void **set** (unsigned int x, unsigned int y, unsigned int z, T val)
- void **add** (unsigned int x, unsigned int y, unsigned int z, T val)
- void **resize_width** (int new_W)
- void **initialize** (T value)

Public Attributes

- int **width** = W
- const int **height** = H
- const int **depth** = D

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

- include/Tools.h

7.3 Bead Class Reference

Public Member Functions

- void **assign_position** (double, double, double)
- void **assign_volume_and_scattlen** (const std::string &)

Public Attributes

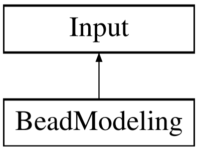
- double **x**
- double **y**
- double **z**
- double **rho**
- double **rho_modified**
- double **v**
- double **nn**
- int **type**
- bool **position_assigned**

The documentation for this class was generated from the following files:

- include/Bead.h
- src/Bead.cpp

7.4 BeadModeling Class Reference

Inheritance diagram for BeadModeling:



Public Member Functions

- **BeadModeling** (const std::string &)
- void **load_input** ()
- void initial_configuration ()
- void **write_xyz** ()
- void **test_flat** ()
- void **update_rho** ()

7.4.1 Member Function Documentation

7.4.1.1 initial_configuration()

```
void BeadModeling::initial_configuration ( )
```

radius of the sphere

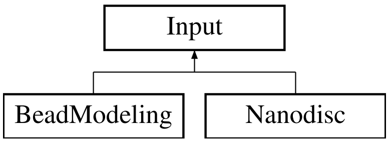
The documentation for this class was generated from the following files:

- include/BeadModeling.h
- src/BeadModeling.cpp

7.5 Input Class Reference

```
#include <Input.h>
```

Inheritance diagram for Input:



Public Member Functions

- std::string parse_line (std::ifstream &, std::string)
- std::string parse_double_delimiter (std::ifstream &, std::string, std::string)
- std::vector< std::vector< double > > load_matrix (const std::string &, int)
- void skip_lines (std::ifstream &, int)

7.5.1 Detailed Description

Class needed to interpret input files

7.5.2 Member Function Documentation

7.5.2.1 load_matrix()

```
vector< vector< double > > Input::load_matrix (
             const std::string & ,
             int  )
```

parses input lines with double delimiters loads a matrix from file. Only the number of columns is needed, not the number of lines.

Inputs

ifstream& file: address of a file stream int ncols: number of columns of the matrix

Returns

vector<vector<double> > vec: the loaded matrix

7.5.2.2 parse_double_delimiter()

```
string Input::parse_double_delimiter (
             std::ifstream & ,
             std::string ,
             std::string  )
```

parses input line with single delimiter parses a numeric value from file line with double delimiter.

Example

from file_line = "!nu_X_h20 50 , comment", the function parses 50 using " " and "," as a delimiters

Inputs

ifstream& file: address of a file stream string delimiter1: first delimiter string delimiter2: second delimiter

Returns

string token: parsed output

7.5.2.3 parse_line()

```
string Input::parse_line (
             std::ifstream & ,
             std::string  )
```

parses a numeric value from file line with single delimiter.

Example

from file_line = "CVLipid 50", the function parses 50 using " " as a delimiter

Inputs

ifstream& file: address of a file stream string delimiter: delimiter

Returns

string str: parsed output

7.5.2.4 skip_lines()

```
void Input::skip_lines (
             std::ifstream & ,
             int  )
```

loads a matrix from file skips a given number of lines while parsing a file

Inputs

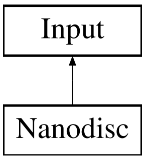
ifstream& file: address of a file stream int num: number of lines to skip

The documentation for this class was generated from the following files:

- include/Input.h
- src/Input.cpp

7.6 Nanodisc Class Reference

Inheritance diagram for Nanodisc:



Public Member Functions

- Nanodisc ()
- ∼Nanodisc ()
- void load_input (const std::string &)
- void nanodisc_form_factor (std::vector< double >)
- double get_radius_major ()
- double get_radius_minor ()
- double get_scale_endcaps ()
- double get_vertical_axis_ellipsoid ()
- double get_rho_solvent ()
- double get_hcore ()
- double get_hlipid ()
- double get_hmethyl ()
- double get_rho_methyl ()
- double get_rho_alkyl ()
- double get_rho_head ()
- double get_cvprotein ()
- double get_xrough ()
- std::complex< double > get_alpha (int, int, int)
- int get_harmonics_order ()

7.6.1 Constructor & Destructor Documentation

7.6.1.1 Nanodisc()

```
Nanodisc::Nanodisc ( )
```

expands the form factor in the basis of spherical harmonics

7.6.1.2 ∼Nanodisc()

```
Nanodisc::∼Nanodisc ( )
```

class constructor

7.6.2 Member Function Documentation

7.6.2.1 get_alpha()

```
complex< double > Nanodisc::get_alpha (
              int i,
              int l,
              int m )
```

returns the roughness coefficient

7.6.2.2 get_cvprotein()

```
double Nanodisc::get_cvprotein ( )
```

returns the scattering length of the lipid heads

7.6.2.3 get_harmonics_order()

```
int Nanodisc::get_harmonics_order ( )
```

returns the value of the expanded form factor at position i,l,m

7.6.2.4 get_hcore()

```
double Nanodisc::get_hcore ( )
```

returns the scattering length of the solvent

7.6.2.5 get_hlipid()

```
double Nanodisc::get_hlipid ( )
```

returns the height of the hydrophobic bilayer

7.6.2.6 get_hmethyl()

```
double Nanodisc::get_hmethyl ( )
```

returns the height of the lipid bilayer

7.6.2.7 get_radius_major()

```
double Nanodisc::get_radius_major ( )
```

computes the form factor of the nanodisc

7.6.2.8 get_radius_minor()

```
double Nanodisc::get_radius_minor ( )
```

returns the major semiaxis of the nanodisc

7.6.2.9 get_rho_alkyl()

```
double Nanodisc::get_rho_alkyl ( )
```

returns the scattering length of methyl groups

7.6.2.10 get_rho_head()

```
double Nanodisc::get_rho_head ( )
```

returns the scattering length of the alkyl chains

7.6.2.11 get_rho_methyl()

```
double Nanodisc::get_rho_methyl ( )
```

returns the height of the methyl groups

7.6.2.12 get_rho_solvent()

```
double Nanodisc::get_rho_solvent ( )
```

returns the vertical axis of the ellipsoid

7.6.2.13 get_scale_endcaps()

```
double Nanodisc::get_scale_endcaps ( )
```

returns the minor semiaxis of the nanodisc

7.6.2.14 get_vertical_axis_ellipsoid()

```
double Nanodisc::get_vertical_axis_ellipsoid ( )
```

returns the scale factor of endcaps

7.6.2.15 get_xrough()

```
double Nanodisc::get_xrough ( )
```

returns the correction factor for the membrane protein volume

7.6.2.16 load_input()

```
void Nanodisc::load_input (
             const std::string &  )
```

class destructor Reads from the WillItFit output file. The number of lines to skip is hardcoded and needs to be re-checked everytime the WillItFit version changes. Excess scattering lengths are in units of the electron scattering length, and are thus adimensional. Volumes are in A^3.

Inputs

const strin& best_fit: path to the best fit file output from WillItFit.

Three checks on volumes are ran to verify the consistency of the quantities loaded from the WillItFit output.

7.6.2.17 nanodisc_form_factor()

```
void Nanodisc::nanodisc_form_factor (
            std::vector< double >  )
```

reads the output of WillItFit to obtain info on the nanodisc

The documentation for this class was generated from the following files:

- include/Nanodisc.h
- src/Nanodisc.cpp

7.7 RandomNumbers Class Reference

Public Member Functions

- RandomNumbers ()
- ∼RandomNumbers ()
- double uniform ()
- double in_range (double, double)
- double in_range2 (double, double)
- double gaussian (double)

7.7.1 Constructor & Destructor Documentation

7.7.1.1 RandomNumbers()

```
RandomNumbers::RandomNumbers ( )
```

class constructor if the define DEBUG is present fix the seed to 0. Generates a reproducible string of rands

allocate gsl arrays for rng calculations

7.7.1.2 ∼RandomNumbers()

```
RandomNumbers::∼RandomNumbers ( )
```

class constructor: allocates specific gsl arrays for the calculation frees gsl_array

7.7.2 Member Function Documentation

7.7.2.1 gaussian()

```
double RandomNumbers::gaussian (
             double tau )
```

compute two uniforms

7.7.2.2 in_range()

```
double RandomNumbers::in_range (
             double min,
             double max )
```

returns a random number uniformly distributed in [0,1]

7.7.2.3 in_range2()

```
double RandomNumbers::in_range2 (
             double min,
             double max )
```

return a random number uniformly distributed in [a,b]

7.7.2.4 uniform()

```
double RandomNumbers::uniform ( )
```

class destructor: deallocates gsl arrays calls gsl function for uniform random numbers in [0,1]

The documentation for this class was generated from the following files:

- include/Random.h
- src/Random.cpp

Index