

## My Project

Generated by Doxygen 1.8.15



<b>1 Drawing Shapes</b>	<b>1</b>
<b>2 README</b>	<b>3</b>
<b>3 How to draw rectangle?</b>	<b>5</b>
<b>4 How to draw circle?</b>	<b>7</b>
<b>5 Hierarchical Index</b>	<b>9</b>
5.1 Class Hierarchy . . . . .	9
<b>6 Class Index</b>	<b>11</b>
6.1 Class List . . . . .	11
<b>7 Class Documentation</b>	<b>13</b>
7.1 Array2D< T, W, H > Class Template Reference . . . . .	13
7.2 Array3D< T, W, H, D > Class Template Reference . . . . .	13
7.3 Bead Class Reference . . . . .	14
7.4 BeadModeling Class Reference . . . . .	14
7.4.1 Member Function Documentation . . . . .	15
7.4.1.1 initial_configuration() . . . . .	15
7.5 Input Class Reference . . . . .	15
7.5.1 Detailed Description . . . . .	15
7.5.2 Member Function Documentation . . . . .	16
7.5.2.1 load_matrix() . . . . .	16
7.5.2.2 parse_double_delimiter() . . . . .	16
7.5.2.3 parse_line() . . . . .	17
7.5.2.4 skip_lines() . . . . .	17
7.6 Nanodisc Class Reference . . . . .	17
7.6.1 Constructor & Destructor Documentation . . . . .	18
7.6.1.1 Nanodisc() . . . . .	18
7.6.1.2 ~Nanodisc() . . . . .	18
7.6.2 Member Function Documentation . . . . .	18
7.6.2.1 get_alpha() . . . . .	18
7.6.2.2 get_cvprotein() . . . . .	19
7.6.2.3 get_harmonics_order() . . . . .	19
7.6.2.4 get_hcore() . . . . .	19
7.6.2.5 get_hlipid() . . . . .	19
7.6.2.6 get_hmethyl() . . . . .	19
7.6.2.7 get_radius_major() . . . . .	19
7.6.2.8 get_radius_minor() . . . . .	19
7.6.2.9 get_rho_alkyl() . . . . .	19
7.6.2.10 get_rho_head() . . . . .	20
7.6.2.11 get_rho_methyl() . . . . .	20

---

7.6.2.12 get_rho_solvent()	20
7.6.2.13 get_scale_endcaps()	20
7.6.2.14 get_vertical_axis_ellipsoid()	20
7.6.2.15 get_xrough()	20
7.6.2.16 load_input()	20
7.6.2.17 nanodisc_form_factor()	21
7.7 RandomNumbers Class Reference	21
7.7.1 Constructor & Destructor Documentation	21
7.7.1.1 RandomNumbers()	21
7.7.1.2 ~RandomNumbers()	21
7.7.2 Member Function Documentation	22
7.7.2.1 gaussian()	22
7.7.2.2 in_range()	22
7.7.2.3 in_range2()	22
7.7.2.4 uniform()	22
<b>Index</b>	<b>23</b>

## 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:

Array2D< T, W, H > . . . . .	13
Array3D< T, W, H, D > . . . . .	13
Array3D< double, 0, NTHETA, NPHI > . . . . .	13
Array3D< std::complex< double >, 0, NH+1, NH+1 > . . . . .	13
Bead . . . . .	14
Input . . . . .	15
BeadModeling . . . . .	14
Nanodisc . . . . .	17
RandomNumbers . . . . .	21



## Chapter 6

# Class Index

### 6.1 Class List

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

<a href="#">Array2D&lt; T, W, H &gt;</a>	13
<a href="#">Array3D&lt; T, W, H, D &gt;</a>	13
<a href="#">Bead</a>	14
<a href="#">BeadModeling</a>	14
<a href="#">Input</a>	15
<a href="#">Nanodisc</a>	17
<a href="#">RandomNumbers</a>	21





## 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

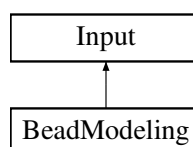
- double **x**
- double **y**
- double **z**
- double **rho**
- double **rho\_modified**
- double **v**
- double **nn**
- int **type**
- bool **position\_assigned**

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

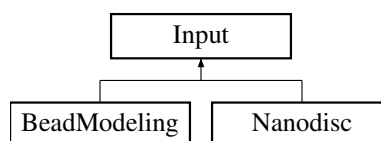
The documentation for this class was generated from the following files:

- 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 = "Inu\_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

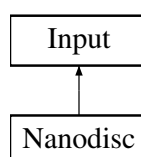
ifstream& file: address of a file stream int num: number of lines to skip

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 WilltFit output file. The number of lines to skip is hardcoded and needs to be re-checked everytime the WilltFit version changes. Excess scattering lengths are in units of the electron scattering length, and are thus adimensional. Volumes are in  $\text{\AA}^3$ .



### 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

- ~Nanodisc
  - Nanodisc, [18](#)
- ~RandomNumbers
  - RandomNumbers, [21](#)
- Array2D< T, W, H >, [13](#)
- Array3D< T, W, H, D >, [13](#)
- Bead, [14](#)
- BeadModeling, [14](#)
  - initial\_configuration, [15](#)
- gaussian
  - RandomNumbers, [22](#)
- get\_alpha
  - Nanodisc, [18](#)
- get\_cvprotein
  - Nanodisc, [18](#)
- get\_harmonics\_order
  - Nanodisc, [19](#)
- get\_hcore
  - Nanodisc, [19](#)
- get\_hlipid
  - Nanodisc, [19](#)
- get\_hmethyl
  - Nanodisc, [19](#)
- get\_radius\_major
  - Nanodisc, [19](#)
- get\_radius\_minor
  - Nanodisc, [19](#)
- get\_rho\_alkyl
  - Nanodisc, [19](#)
- get\_rho\_head
  - Nanodisc, [19](#)
- get\_rho\_methyl
  - Nanodisc, [20](#)
- get\_rho\_solvent
  - Nanodisc, [20](#)
- get\_scale\_endcaps
  - Nanodisc, [20](#)
- get\_vertical\_axis\_ellipsoid
  - Nanodisc, [20](#)
- get\_xrough
  - Nanodisc, [20](#)
- in\_range
  - RandomNumbers, [22](#)
- in\_range2
  - RandomNumbers, [22](#)
- initial\_configuration
  - BeadModeling, [15](#)
  - Input, [15](#)
    - load\_matrix, [16](#)
    - parse\_double\_delimiter, [16](#)
    - parse\_line, [16](#)
    - skip\_lines, [17](#)
  - load\_input
    - Nanodisc, [20](#)
  - load\_matrix
    - Input, [16](#)
  - Nanodisc, [17](#)
    - ~Nanodisc, [18](#)
    - get\_alpha, [18](#)
    - get\_cvprotein, [18](#)
    - get\_harmonics\_order, [19](#)
    - get\_hcore, [19](#)
    - get\_hlipid, [19](#)
    - get\_hmethyl, [19](#)
    - get\_radius\_major, [19](#)
    - get\_radius\_minor, [19](#)
    - get\_rho\_alkyl, [19](#)
    - get\_rho\_head, [19](#)
    - get\_rho\_methyl, [20](#)
    - get\_rho\_solvent, [20](#)
    - get\_scale\_endcaps, [20](#)
    - get\_vertical\_axis\_ellipsoid, [20](#)
    - get\_xrough, [20](#)
    - load\_input, [20](#)
    - Nanodisc, [18](#)
    - nanodisc\_form\_factor, [21](#)
  - nanodisc\_form\_factor
    - Nanodisc, [21](#)
  - parse\_double\_delimiter
    - Input, [16](#)
  - parse\_line
    - Input, [16](#)
  - RandomNumbers, [21](#)
    - ~RandomNumbers, [21](#)
    - gaussian, [22](#)
    - in\_range, [22](#)
    - in\_range2, [22](#)
    - RandomNumbers, [21](#)
    - uniform, [22](#)
  - skip\_lines
    - Input, [17](#)

uniform

RandomNumbers, [22](#)