

Carbon Nanotube Monte Carlo

Generated by Doxygen 1.8.10

Thu Jul 16 2015 12:39:49

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	_WDIR Struct Reference	3
2.2	_wdirent Struct Reference	3
2.3	CNT Class Reference	3
2.3.1	Constructor & Destructor Documentation	4
2.3.1.1	CNT()	4
2.3.1.2	CNT(const string fileName, const string filePath, double segLenMin)	4
2.3.2	Member Function Documentation	4
2.3.2.1	getCNTNum()	4
2.3.2.2	getCylinderHeight()	4
2.3.2.3	getDiameter()	5
2.3.2.4	getLength()	5
2.3.2.5	getm()	5
2.3.2.6	getMinSpacing()	5
2.3.2.7	getn()	6
2.3.2.8	getTubeSeparation()	7
2.3.2.9	isInitialized()	7
2.4	DIR Struct Reference	7
2.5	dirent Struct Reference	7
2.6	exciton Class Reference	8
2.6.1	Detailed Description	8
2.6.2	Constructor & Destructor Documentation	8
2.6.2.1	exciton()	8
2.6.2.2	exciton(int cidx, int sidx, int energy)	8
2.6.2.3	~exciton()	9
2.6.3	Member Function Documentation	9
2.6.3.1	getCNTidx()	9
2.6.3.2	getEnergy()	9

2.6.3.3	getSegidx()	9
2.6.3.4	setCNTidx(int cidx)	9
2.6.3.5	setEnergy(int energy)	10
2.6.3.6	setSegidx(int sidx)	10
2.7	segment Struct Reference	10
2.7.1	Member Function Documentation	10
2.7.1.1	hasExactExciton(shared_ptr< exciton > e)	10
2.7.1.2	hasExciton(shared_ptr< exciton > e)	11
2.7.1.3	removeExciton(shared_ptr< exciton > e)	11
2.7.1.4	setExciton(shared_ptr< exciton > e)	11
2.8	tableElem Class Reference	11
2.8.1	Constructor & Destructor Documentation	12
2.8.1.1	tableElem()	12
2.8.1.2	tableElem(double rnew, double t, double g, int tube, int seg)	12
2.8.1.3	~tableElem()	12
2.8.2	Member Function Documentation	12
2.8.2.1	calcDist(Vector3d v1, Vector3d v2)	12
2.8.2.2	calcThet(vector< shared_ptr< segment >>::iterator s1, vector< shared_ptr< segment >>::iterator s2)	12
2.8.2.3	getGamma()	13
2.8.2.4	getr()	13
2.8.2.5	getRate()	13
2.8.2.6	getSegidx()	13
2.8.2.7	getTheta()	13
2.8.2.8	getTubeidx()	13
	Index	15

Chapter 1

Class Index

1.1 Class List

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

_WDIR	3
_wdirent	3
CNT	3
DIR	7
dirent	7
exciton	8
segment	10
tableElem	11

Chapter 2

Class Documentation

2.1 _WDIR Struct Reference

Public Attributes

- struct [_wdirent](#) **ent**
- WIN32_FIND_DATAW **data**
- int **cached**
- HANDLE **handle**
- wchar_t * **patt**

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

- CNTMC_Project/CNTMC_Project/dirent.h

2.2 _wdirent Struct Reference

Public Attributes

- long **d_ino**
- unsigned short **d_reclen**
- size_t **d_namlen**
- int **d_type**
- wchar_t **d_name** [PATH_MAX]

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

- CNTMC_Project/CNTMC_Project/dirent.h

2.3 CNT Class Reference

Public Member Functions

- [CNT](#) ()
- [CNT](#) (const string fileName, const string filePath, double segLenMin)
- double [getDiameter](#) ()
- double [getLength](#) ()

- double [getCylinderHeight](#) ()
- double [getTubeSeparation](#) ()
- double [getMinSpacing](#) ()
- int [getm](#) ()
- int [getn](#) ()
- int [getCNTNum](#) ()
- bool [isInitialized](#) ()

Public Attributes

- shared_ptr< vector< shared_ptr< [segment](#) > > > **segs**

2.3.1 Constructor & Destructor Documentation

2.3.1.1 CNT::CNT ()

Sets the [CNT](#) object to some default values. DO NOT USE CNTs CONSTRUCTED THIS WAY. This is only to appease the compiling gods.

Returns

[CNT](#) Object

2.3.1.2 CNT::CNT (const string *fileName*, const string *folderPath*, double *segLen*)

Reads a [CNT](#) file and creates a [CNT](#) object with all the information stored in that file.

Parameters

<i>filePath</i>	The path of the file containing the CNT info
-----------------	--

Returns

[CNT](#) Object

2.3.2 Member Function Documentation

2.3.2.1 int CNT::getCNTNum ()

Gets the tube number

Parameters

<i>void</i>	
-------------	--

Returns

[CNT](#) number

2.3.2.2 double CNT::getCylinderHeight ()

Gets the height of each compositional cylinders

Parameters

<i>void</i>	
-------------	--

Returns

The height of each compositional cylinders

2.3.2.3 double CNT::getDiameter ()

Gets the diameter of the [CNT](#)

Parameters

<i>void</i>	
-------------	--

Returns

Diameter of the [CNT](#)

2.3.2.4 double CNT::getLength ()

Gets the length of the [CNT](#)

Parameters

<i>void</i>	
-------------	--

Returns

Length of the [CNT](#)

2.3.2.5 int CNT::getm ()

Gets m parameter of the [CNT](#)

Parameters

<i>void</i>	
-------------	--

Returns

m

2.3.2.6 double CNT::getMinSpacing ()

Gets the minimum spacing between two nanotubes

Parameters

<i>void</i>	
-------------	--

Returns

minimum spacing between two nanotubes

2.3.2.7 int CNT::getn ()

Gets n parameter of the [CNT](#)

Parameters

<i>void</i>	
-------------	--

Returns

n

2.3.2.8 double CNT::getTubeSeparation ()

Gets the separation between two compositional cylinders

Parameters

<i>void</i>	
-------------	--

Returns

The separation between two compositional cylinders

2.3.2.9 bool CNT::isInitialized ()

Says whether or not the CNT was initialized

Returns

initialization status

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

- CNTMC_Project/CNTMC_Project/CNT.h
- CNTMC_Project/CNTMC_Project/CNT.cpp

2.4 DIR Struct Reference

Public Attributes

- struct dirent **ent**
- struct _WDIR * **wdirp**

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

- CNTMC_Project/CNTMC_Project/dirent.h

2.5 dirent Struct Reference

Public Attributes

- long **d_ino**
- unsigned short **d_reclen**
- size_t **d_namlen**
- int **d_type**

- char **d_name** [PATH_MAX]

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

- CNTMC_Project/CNTMC_Project/dirent.h

2.6 exciton Class Reference

```
#include <exciton.h>
```

Public Member Functions

- [exciton](#) ()
- [exciton](#) (int cidx, int sidx, int energy)
- [~exciton](#) ()
- void [setCNTidx](#) (int cidx)
- void [setSegidx](#) (int sidx)
- void [setEnergy](#) (int energy)
- int [getCNTidx](#) ()
- int [getSegidx](#) ()
- int [getEnergy](#) ()

2.6.1 Detailed Description

[exciton.h](#) Purpose: Header for exciton.cpp

Author

Alex Gabourie

Version

1.00

2.6.2 Constructor & Destructor Documentation

2.6.2.1 [exciton::exciton](#) ()

[exciton.cpp](#) Purpose: Stores relevant exciton information

Author

Alex Gabourie

Version

1.00 Creates exciton object

Returns

[tableElem](#) Object

2.6.2.2 [exciton::exciton](#) (int *cidx*, int *sidx*, int *energy*)

Creates exciton object

Parameters

<i>cidx</i>	Index of the CNT the exciton belongs to
<i>sidx</i>	Index of the segment the exciton belongs to
<i>energy</i>	Whether the 1st or 2nd energy level

Returns

tableElem Object

2.6.2.3 exciton::~~exciton ()

destroys exciton object

Returns

tableElem Object

2.6.3 Member Function Documentation

2.6.3.1 int exciton::getCNTidx ()

Gets cnt index

Returns

cnt index

2.6.3.2 int exciton::getEnergy ()

Gets energy level

Returns

energy level

2.6.3.3 int exciton::getSegidx ()

Get segment index

Returns

segment index

2.6.3.4 void exciton::setCNTidx (int cidx)

Sets cnt index

Parameters

<i>cidx</i>	Index of the CNT the exciton belongs to
-------------	---

2.6.3.5 void exciton::setEnergy (int *energy*)

Sets energy level

Parameters

<i>energy</i>	Whether the 1st or 2nd energy level
---------------	-------------------------------------

2.6.3.6 void exciton::setSegidx (int *sidx*)

Sets segment index

Parameters

<i>sidx</i>	Index of the segment the exciton belongs to
-------------	---

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

- CNTMC_Project/CNTMC_Project/exciton.h
- CNTMC_Project/CNTMC_Project/exciton.cpp

2.7 segment Struct Reference

Public Member Functions

- bool [hasExciton](#) (shared_ptr< [exciton](#) > e)
- bool [setExciton](#) (shared_ptr< [exciton](#) > e)
- bool [removeExciton](#) (shared_ptr< [exciton](#) > e)
- bool [hasExactExciton](#) (shared_ptr< [exciton](#) > e)

Public Attributes

- int **segNum**
- Vector3d **p1**
- Vector3d **p2**
- Vector3d **mid**
- shared_ptr< vector< [tableElem](#) > > **tbl**
- shared_ptr< vector< double > > **rateVec**
- shared_ptr< [exciton](#) > **ex1**
- shared_ptr< [exciton](#) > **ex2**

2.7.1 Member Function Documentation

2.7.1.1 bool segment::hasExactExciton (shared_ptr< [exciton](#) > e)

Checks to see if the exciton that is passes is the exact exciton that already exists in the location.

2.7.1.2 `bool segment::hasExciton (shared_ptr< exciton > e)`

segment.cpp Purpose: Segment struct used in each CNT object

Author

Alex Gabourie

Version

1.00 Determines if the segment has an exciton of the same type as the passed exciton

Parameters

<code>e</code>	The exciton desired to see if a slot is available
----------------	---

Returns

Whether or not the exciton can be added

2.7.1.3 `bool segment::removeExciton (shared_ptr< exciton > e)`

Removes the exciton of the correct type

Parameters

<code>e</code>	The exciton desired to be removed from the segment
----------------	--

Returns

True if exciton removed, false if no exciton to remove

2.7.1.4 `bool segment::setExciton (shared_ptr< exciton > e)`

Sets the exciton in the correct slot to the exciton passed to function.

Parameters

<code>e</code>	The exciton desired to be added to the segment
----------------	--

Returns

True if assignment works and false if assignment not successful

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

- CNTMC_Project/CNTMC_Project/segment.h
- CNTMC_Project/CNTMC_Project/segment.cpp

2.8 tableElem Class Reference

Public Member Functions

- [tableElem](#) ()
- [tableElem](#) (double rnew, double t, double g, int tube, int seg)

- [~tableElem](#) ()
- double [getRate](#) ()
- double [getr](#) ()
- double [getTheta](#) ()
- double [getGamma](#) ()
- int [getTubeidx](#) ()
- int [getSegidx](#) ()

Static Public Member Functions

- static double [calcDist](#) (Vector3d v1, Vector3d v2)
- static double [calcThet](#) (vector< shared_ptr< [segment](#) >>::iterator s1, vector< shared_ptr< [segment](#) >>::iterator s2)

2.8.1 Constructor & Destructor Documentation

2.8.1.1 [tableElem::tableElem](#) ()

Creates table element object

Returns

[tableElem](#) Object

2.8.1.2 [tableElem::tableElem](#) (double *rnew*, double *t*, double *g*, int *tube*, int *seg*)

Creates table element object

Returns

[tableElem](#) Object

2.8.1.3 [tableElem::~~tableElem](#) ()

Destructor for class

2.8.2 Member Function Documentation

2.8.2.1 double [tableElem::calcDist](#) (Vector3d *v1*, Vector3d *v2*) [static]

Calculates distance between two segments

Parameters

<i>v1</i>	first segment center
<i>v2</i>	second segment center

Returns

The distance between v1 and v2

2.8.2.2 double [tableElem::calcThet](#) (vector< shared_ptr< [segment](#) >>::iterator *s1*, vector< shared_ptr< [segment](#) >>::iterator *s2*) [static]

Calculates the angle between two vectors

Parameters

<i>s1</i>	first segment
<i>s2</i>	second segment

2.8.2.3 double tableElem::getGamma ()

Gets gamma value

Returns

gamma value

2.8.2.4 double tableElem::getr ()

Gets r value

Returns

r value

2.8.2.5 double tableElem::getRate ()

Gets the total transition rate based on gamma, r, and theta

Returns

The transition rate in inverse seconds

2.8.2.6 int tableElem::getSegidx ()

Gets the segment number

Returns

segment number

2.8.2.7 double tableElem::getTheta ()

Gets theta value

Returns

theta value

2.8.2.8 int tableElem::getTubeidx ()

Gets tube number

Returns

tube number

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

- CNTMC_Project/CNTMC_Project/tableElem.h
- CNTMC_Project/CNTMC_Project/tableElem.cpp

Index

- [_WDIR, 3](#)
 - [_wdirent, 3](#)
 - [~exciton](#)
 - [exciton, 9](#)
 - [~tableElem](#)
 - [tableElem, 12](#)
- [CNT, 3](#)
 - [CNT, 4](#)
 - [getCNTNum, 4](#)
 - [getCylinderHeight, 4](#)
 - [getDiameter, 5](#)
 - [getLength, 5](#)
 - [getMinSpacing, 5](#)
 - [getTubeSeparation, 7](#)
 - [getm, 5](#)
 - [getn, 5](#)
 - [isInitialized, 7](#)
- [calcDist](#)
 - [tableElem, 12](#)
- [calcThet](#)
 - [tableElem, 12](#)
- [DIR, 7](#)
- [dirent, 7](#)
- [exciton, 8](#)
 - [~exciton, 9](#)
 - [exciton, 8](#)
 - [getCNTidx, 9](#)
 - [getEnergy, 9](#)
 - [getSegidx, 9](#)
 - [setCNTidx, 9](#)
 - [setEnergy, 10](#)
 - [setSegidx, 10](#)
- [getCNTNum](#)
 - [CNT, 4](#)
- [getCNTidx](#)
 - [exciton, 9](#)
- [getCylinderHeight](#)
 - [CNT, 4](#)
- [getDiameter](#)
 - [CNT, 5](#)
- [getEnergy](#)
 - [exciton, 9](#)
- [getGamma](#)
 - [tableElem, 13](#)
- [getLength](#)
 - [CNT, 5](#)
- [getMinSpacing](#)
 - [CNT, 5](#)
- [getRate](#)
 - [tableElem, 13](#)
- [getSegidx](#)
 - [exciton, 9](#)
 - [tableElem, 13](#)
- [getTheta](#)
 - [tableElem, 13](#)
- [getTubeSeparation](#)
 - [CNT, 7](#)
- [getTubeidx](#)
 - [tableElem, 13](#)
- [getm](#)
 - [CNT, 5](#)
- [getn](#)
 - [CNT, 5](#)
- [getr](#)
 - [tableElem, 13](#)
- [hasExactExciton](#)
 - [segment, 10](#)
- [hasExciton](#)
 - [segment, 10](#)
- [isInitialized](#)
 - [CNT, 7](#)
- [removeExciton](#)
 - [segment, 11](#)
- [segment, 10](#)
 - [hasExactExciton, 10](#)
 - [hasExciton, 10](#)
 - [removeExciton, 11](#)
 - [setExciton, 11](#)
- [setCNTidx](#)
 - [exciton, 9](#)
- [setEnergy](#)
 - [exciton, 10](#)
- [setExciton](#)
 - [segment, 11](#)
- [setSegidx](#)
 - [exciton, 10](#)
- [tableElem, 11](#)
 - [~tableElem, 12](#)
 - [calcDist, 12](#)
 - [calcThet, 12](#)
 - [getGamma, 13](#)
 - [getRate, 13](#)

getSegidx, [13](#)
getTheta, [13](#)
getTubeidx, [13](#)
getr, [13](#)
tableElem, [12](#)