Carbon Nanotube Monte Carlo

Generated by Doxygen 1.8.10

Thu Jul 16 2015 12:39:49

Contents

Chapter 1

Class Index

1.1 Class List

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

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

filePath	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

void	

Returns

CNT number

2.3.2.2 double CNT::getCylinderHeight ( )

Gets the height of each compositional cylinders

Parameters

void	

Returns

The height of each compositional cylinders

2.3.2.3 double CNT::getDiameter ( )

Gets the diameter of the CNT

Parameters

void	

Returns

Diameter of the CNT

2.3.2.4 double CNT::getLength ( )

Gets the length of the CNT

Parameters

void	

Returns

Length of the CNT

2.3.2.5 int CNT::getm ( )

Gets m parameter of the CNT

Parameters

void	

Returns

m

2.3.2.6 double CNT::getMinSpacing ( )

Gets the minimum spacing between two nanotubes

Parameters

void	

Returns

minimum spacing between two nanotubes

2.3.2.7 int CNT::getn ( )

Gets n parameter of the CNT

Parameters

void	

Returns

n

2.3.2.8 double CNT::getTubeSeparation ( )

Gets the separation between two compositional cylinders

Parameters

void	

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

cidx	Index of the CNT the exciton belongs to
sidx	Index of the segment the exciton belongs to
energy	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

cidx	Index of the CNT the exciton belongs to

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

Sets energy level

Parameters

energy	Whether the 1st or 2nd energy level

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

Sets segment index

Parameters

sidx	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

e	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

e	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

e	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

v1	first segment center
v2	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

s1	first segment
s2	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