Instruction Manual

# Carbon Nanotube Mesh Generator

Version: 1.03

Alexander Gabourie

Amirhossein Davoody

Prof. Irena Knezevic

DEPARTMENT OF
Electrical and
Computer Engineering
UNIVERSITY OF WISCONSIN–MADISON

## Contents

## Introduction

The National Academy of Engineering, concerned about the increasing energy demands as well as the environmental cost of meeting them, has determined that development of an economic solar panel is a grand engineering challenge of the 21st century. To address the challenge, much effort has been placed towards finding a suitable light-absorbing material. In recent years, carbon nanotubes (CNTs) have become a material candidate due to more effective separation techniques.[1] Researchers now have the ability to separate optically absorptive, semiconducting nanotubes from others. With a wide range of possible band gaps due to different nanotube chiralities, CNTs can be used to create widely tunable photovoltaic cells. The optical versatility, ultrafast exciton and charge transport, and chemical stability of CNTs has been the impetus of much recent research.[2]

While research on bilayer donor/acceptor heterojunction photovoltaics has shown that exciton dissociation at the semiconducting CNT (s-CNT) and $C_{60}$ junction [fig. 1] is very efficient, the diffusion of the exciton to the junction has been determined to be a bottleneck in increasing the overall process efficiency.[2] Amirhossein Davoody, a member of Professor Knezevic's research group at the the University of Wisconsin-Madison, has undertaken a project to develop the theory of exciton diffusion in CNTs and use his theoretical framework to computationally simulate exciton dynamics in a CNT mesh. Before development of the CNT Mesh Generator began, he had already found success calculating exciton transfer rates between CNTs with arbitrary chirality, angle, and separation.

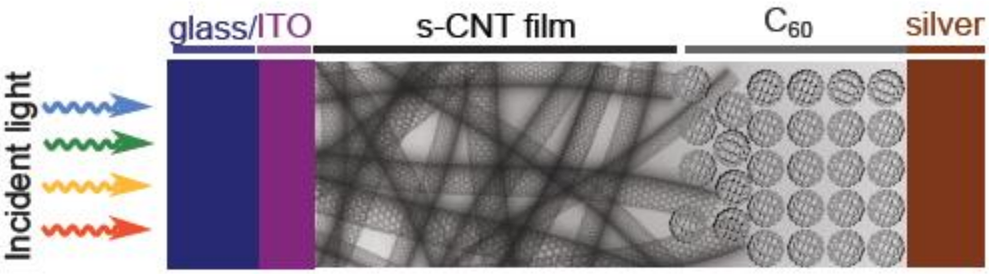


**Fig. 1** Structure of a CNT-based bilayer donor/acceptor heterojunction photovoltaic device.[2]

Since s-CNTs are separated from others through colloidal separation techniques and then deposited on substrates through spin coating, doctor-blade casting, or spray coating, there is no reason to believe that CNTs will have any sort of order when used on a device. If the production devices have uncorded CNTs, then ultimately Amirhossein's simulation should reflect that as accurately as possible. Unfortunately, no software was found that could create the semi-realistic, random network of CNTs desired for the project.

The purpose of this software is to give a user the ability to choose CNT and device region parameters to create a random mesh of CNTs. Once the mesh is complete, each tube's physical parameters are output to .csv file and are available for future processing. With respect to Amirhossein's project's needs, which was the motivation for this software, the final positions of the CNTs, in conjunction with the exciton transfer rate tables, allow the exciton dynamics to be modelled using the Monte Carlo method.

This manual provides the information necessary to understand how to use the software and how the software works.

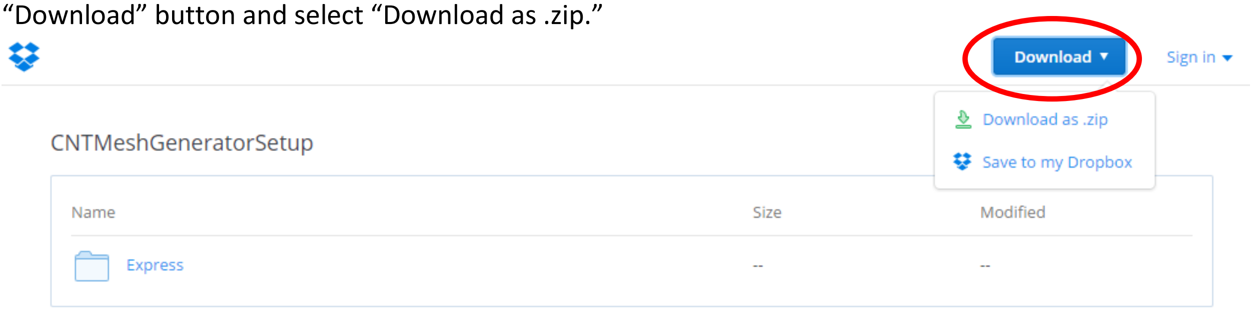
## Usage

### Installing

The current set-up for the software does not have a website for automatic updates. However, it does have a Dropbox link to the most recent deployed version of the software. The link is:

https://www.dropbox.com/sh/x1aqk4ogpjp9gbe/AAAZMK3uj38gCj4krndCipg_a?dl=0

After clicking on the link you will come to a screen that looks like the image below. Click on the "Download" button and select "Download as .zip."



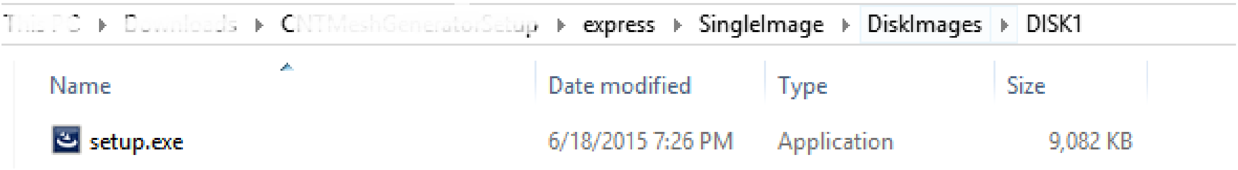
CNTMeshGeneratorSetup

| Name | Size | Modified |
|---|---|---|
| Express | -- | -- |

This will download all the necessary program files. Extract the .zip to a directory of your choosing and open that directory. From that directory navigate to the following path:

*.\express\SingleImage\DiskImages\DISK1*

At this location, a setup executable will be located as seen in the picture below.



This PC ▸ Downloads ▸ CNTMeshGeneratorSetup ▸ express ▸ SingleImage ▸ DiskImages ▸ DISK1

| Name | Date modified | Type | Size |
|---|---|---|---|
| setup.exe | 6/18/2015 7:26 PM | Application | 9,082 KB |

Click on “setup.exe” and follow the on screen instructions to install. The installer may ask to download and install some Microsoft redistributables. Please install them as they are necessary components to the software.

Once the program is installed, it is important to be aware of the location of the CNT_Mesh_Config.xml file. It will initially be located in the program’s directory. This is an example xml file that is necessary to run the simulation. Feel free to move or copy the file. If you are to edit the XML file, only change the text that are part of the elements, do not change the elements themselves.

## Initializing

One can run the program using either the desktop shortcut or the command line. If running the software from the desktop shortcut, a window will appear and ask for the path of the configuration file. The configuration file is an XML file in the format of CNT_Mesh_Config.xml, a file that is included in the program directory. Type the full path to the desired configuration xml file. If running from the command line, there is the option to include the file path to the configuration XML as a command line argument. If no argument is provided, the user will get the same prompt as seen when using the desktop shortcut.

### Configuration File

The configuration file itself can be edited to change the simulation parameters and output locations. This section is focused on explaining each of the editable sections.

Before describing the sections, the user should be aware of acceptable inputs for units in the configuration files. The acceptable terms are listed below:
-> mm, millimeter
-> um, micrometer
-> nm, nanometer
-> pm, picometer
-> A, angstrom

***outputDirectory:*** The folder in which the output files are to be saved. Enter a full file path.

***numberTubes:*** The number of CNTs in the simulation. Must be an integer greater than zero. (Default: 5)

***friction:*** The amount of friction between surfaces. Due to non-realistic gravity, this is non-physical, but can be tuned based on personal preference. The value must be greater than zero. (Default: 1.0)

***gravity:*** The acceleration of gravity. Due to small device sizes, accurate gravity causes nanotubes to move too quickly for simulation step sizes and will cause errors. Tune based on personal preference. The value must be less than zero. (Default: -9.81)

***spacing:*** The minimum spacing between different nanotubes. For <min>, enter any positive number. For units, refer to top of section. (Default: 1.5 nm)

***Lengths:*** The range of lengths that are acceptable for the simulation. <Lmin> is the minimum length, <Lmax> is the maximum. <Lmin> must be less than or equal to <Lmax> and both must be greater than 0. For units, refer to top of section. The simulation chooses a random length that is between <Lmin> and <Lmax>. (Default: Lmin = 100.0 nm, Lmax = 200.0 nm)

***DeviceDimensions:*** The size of the box-like container that will be considered the device region. The <xdim> and <zdim> elements are for the base of the region. The <ydim> element is for the height of the region. All dimensions must be greater than zero. For units, refer to top of section. (Default: xdim = 150.0 nm, ydim = 100.0 nm, zdim = 150.0 nm)

***chirality:*** The chiralities that are acceptable in the simulation. The user is able to enter as many <cnt> elements as desired as long as <n> and <m> are defined as part of the <cnt> children. The software requires positive <n> and <m> values as well as <m> less than or equal to <n>.

The previously discussed parameters are the only parameters relevant to generating the carbon nanotube mesh. The XML file will have other elements past the chirality element which can be ignored. The extra parameters are for a Monte Carlo simulation that uses the output of the mesh generator.

## Runtime

Once the configuration document is accepted by the program, two windows will open. The terminal window will pass along any information about program failures. The other window is the rendering window that shows the user where the simulation is currently at.

Since rendering the objects takes computational resources, the simulation initially renders nothing. This way if the software is called and visuals are not important, the simulation runs anywhere from 25%-50% faster. To turn on the visuals, refer to the top left of the rendering window. It will have instructions to push [shift + d] to toggle the debugDrawer, which will turn rendering on and off.

Another important keypress combination is [shift + q] as this manually ends the simulation. Since the simulation produces random lengths, chiralities, positions, and angles for the CNTs there may be cases when the auto-completion algorithm does not work or works much too slowly. If the user believes the mesh to be good enough, then they have the option to manually end the simulation.

Overall there are many options during runtime. The controls are displayed in the table below. The user is not required to give any input to the simulation during runtime. Controls are for additional needs a user may have.

**Rendering Window Controls**

| Button | Action |
|---|---|
| a | Toggle axis-aligned boundary box |
| w | Toggle wire frame |
| Shift + d | Toggle rendering |
| Shift + q | End Simulation |
| Space | Restart simulation |
| l | Step camera left |
| r | Step camera right |
| f | Step camera forwards |
| b | Step camera backwards |
| z | Zoom in |
| x | Zoom out |
| g | Enable shadows |
| u | Enable textures |
| d | Disable deactivation (current program will not end if pressed!) |
| ctrl + left mouse | Pitch and yaw |
| ctrl + middle mouse | Pan |
| ctrl + right mouse | Zoom in/out |
| left mouse | Grab object |
| right mouse | Shoot box |
| o | Orthonormal view |
| +/- | Shoot box speed +/- 10 |
| h | Hide profiling box |

**Table 1** Controls for the rendering window.

If the user decides to render the simulation they will be able to see colors on the rendered CNTs. The colors of each cylinder help indicate the overall progress of the simulation. The colors mean the following:

| Activation State of Cylinder | Normal | Wire Frame |
|---|---|---|
| Active | Purple | White |
| Wants Deactivation | Blue | Light Blue |
| Sleeping | Light Blue | Green |

**Table 2** Colors of different activation states in different modes.

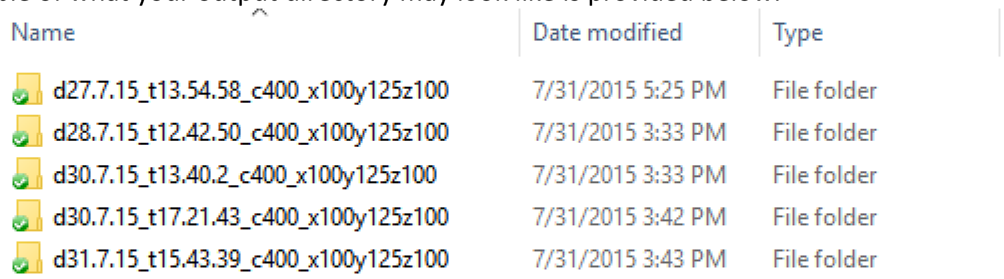
These color states are used to determine if the nanotubes have stopped moving. If a pre-selected percentage of each CNT's cylinders are wanting deactivation or are sleeping, the simulation will end.

## Exit

Once the simulation ends, each CNT that was in the simulation will have its own output file. These output files will be found in a folder with a descriptive, unique name to help identify the run. The folder name format is as follows:

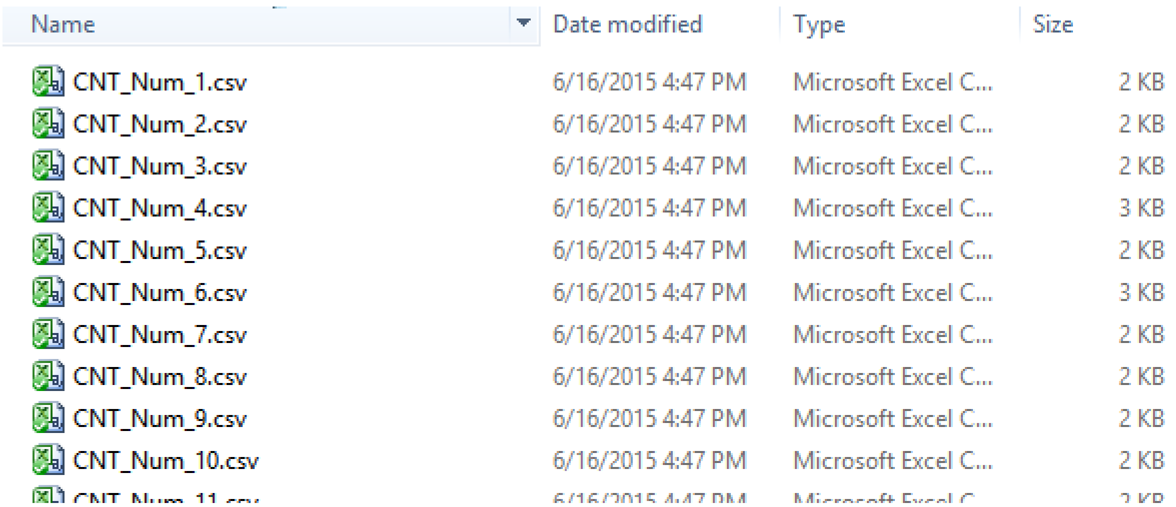
date (d) + time (t) + number of CNTs (c) + device region dimensions (x,y,z)

An example of what your output directory may look like is provided below.



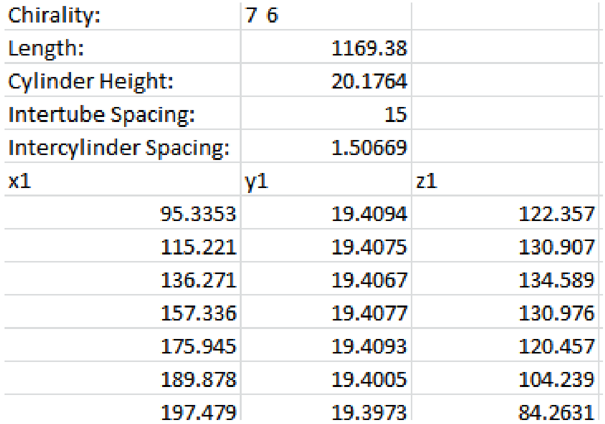
| Name | Date modified | Type |
|---|---|---|
| d27.7.15_t13.54.58_c400_x100y125z100 | 7/31/2015 5:25 PM | File folder |
| d28.7.15_t12.42.50_c400_x100y125z100 | 7/31/2015 3:33 PM | File folder |
| d30.7.15_t13.40.2_c400_x100y125z100 | 7/31/2015 3:33 PM | File folder |
| d30.7.15_t17.21.43_c400_x100y125z100 | 7/31/2015 3:42 PM | File folder |
| d31.7.15_t15.43.39_c400_x100y125z100 | 7/31/2015 3:43 PM | File folder |

The files in each run's directory will be labeled according to which CNT the data is associated with. They will look like the example below.



| Name | Date modified | Type | Size |
|---|---|---|---|
| CNT_Num_1.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_2.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_3.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_4.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 3 KB |
| CNT_Num_5.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_6.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 3 KB |
| CNT_Num_7.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_8.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_9.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_10.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |
| CNT_Num_11.csv | 6/16/2015 4:47 PM | Microsoft Excel C... | 2 KB |

The output files themselves look something like the following:



| | | |
|---|---|---|
| Chirality: | 7 6 | |
| Length: | 1169.38 | |
| Cylinder Height: | 20.1764 | |
| Intertube Spacing: | 15 | |
| Intercylinder Spacing: | 1.50669 | |
| x1 | y1 | z1 |
| 95.3353 | 19.4094 | 122.357 |
| 115.221 | 19.4075 | 130.907 |
| 136.271 | 19.4067 | 134.589 |
| 157.336 | 19.4077 | 130.976 |
| 175.945 | 19.4093 | 120.457 |
| 189.878 | 19.4005 | 104.239 |
| 197.479 | 19.3973 | 84.2631 |

The output file format gives details about each tube. The chirality provides enough information for many physical and electrical calculations. The other parameters represent the physical parameters not captured by the CNT's chirality. Most importantly, the positions of each compositional cylinder as well as the positions of the constraints connecting them are provided. This position data is very important to the Monte Carlo method required for Amirhossein's project (described in the introduction).

In addition to the CNT output files, the configuration XML file is copied into the results folder and is renamed to run identifier used for the results folder. This ensures that the simulation configuration is tied to the results in name as well as file location. The file location is also important as the Monte Carlo simulation reads the contents of the results folder and relies on information in the configuration file to run.

Lastly, a bitmap screenshot of the finished simulation is added to the output folder. With the screenshot, the user can rapidly verify that the CNT mesh simulation ended properly. The screenshot is also included in the case that the user ends the simulation manually.

*Additional Note:*
In this version of the software a MATLAB script is included that can import the .csv files into a MATLAB workspace.

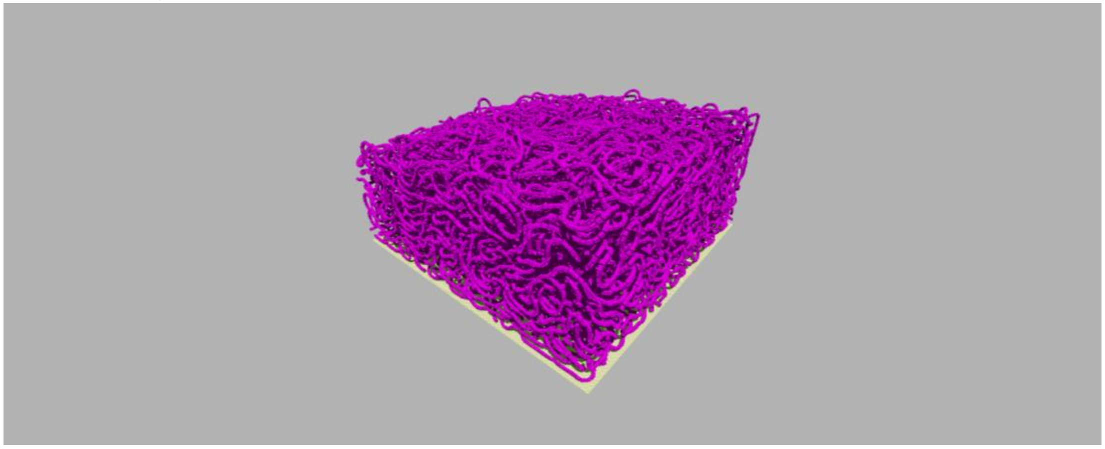


**Fig. 2** Screenshot of finished mesh

## Simulation Mechanics

Assuming that all of the configuration inputs are correct, the program has a simple flow chart that can describe its behavior.

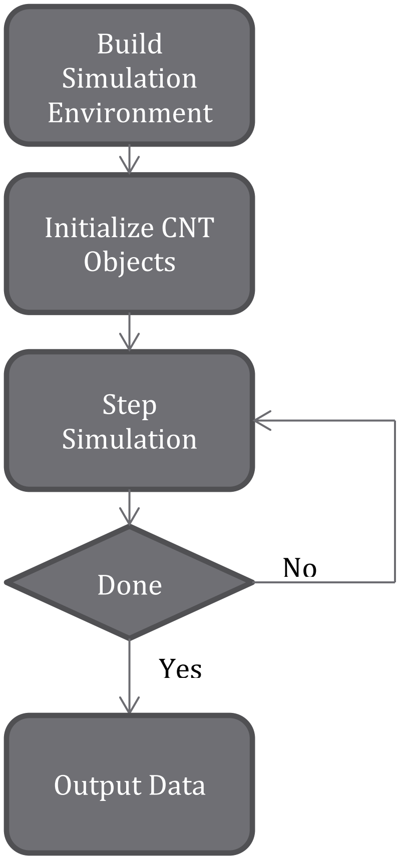


**Fig. 3** Program flow chart.

The simulation environment is the device region specified in the configuration file, plus some additional constructs based on that information to ensure that all nanotubes end up in the device region.

Initializing the CNT objects consists of calculating physical nanotube parameters based on the CNT's chirality and more. The parameters that are calculated from chiral numbers are tube diameter and maximum allowed curvature. Additionally, since each CNT is given a random length between the two length limits, compositional cylinder heights and cylinder spacing must be calculated to ensure that the specified lengths are used in the simulation. Lastly, spacing cylinders are added to the CNT object to enforce the minimum spacing between CNT requirements. Once all of the objects are created for each CNT, they are rotated and shifted to their random initial starting position and angle in the simulation. All of the CNT objects are added to the dynamics world used for the actual physics simulation.

Stepping the simulation simply takes all of the objects, applies the force of gravity to them for some time step, calculates any forces due to object collisions, and updates all object's positions accordingly. Bullet physics performs this step accurately and efficiently as the software is the result of combining decades of collision detection and rigid body dynamics algorithms.

After each simulation step, if the CNTs are determined to be approximately at rest, then the simulation will stop and the CNT data will be output to file. If not, then the simulation steps again.

### Carbon Nanotube Parameters

The current version of the software is only concerned with a few properties of carbon nanotubes: Diameter, helicity, and critical curvature. The diameter (equation **(1)**) is calculated directly from the Hamada parameters n and m. This number is used to determine the radius of the compositional cylinders of the CNT object in the simulation.

$$d_t = a\sqrt{n^2 + m^2 + nm} \qquad \textbf{(1)}$$

A CNT's helicity (equation **(2)**) is the angle between the perimeter vector and the basis vector.[3] This is just another way to understand the chirality of the CNT and is essential for critical curvature calculations.

$$h_g = \cos^{-1}\left(\frac{\mathbf{a}_1 \bullet C_h}{|\mathbf{a}_1||C_h|}\right) \qquad \textbf{(2)}$$

The critical curvature of a carbon nanotube is the maximum curvature a CNT can have before a kink will form in the nanotube.[4] Figure 3 shows an example of a nanotube that has been bent past the critical curvature and has created a kink to relieve stresses. Recent studies have shown how even slight compressive forces on a nanotube cause changes in its electronic structure.[5] Intuitively this conclusion makes sense as the atoms' orbitals will overlap differently if the physical structure of the tube is changed. A kink is an extreme deformation and since Amirhossein's code does not account for the electronic structure of a CNT with a kink, we choose to avoid angles that would give this structure. While we know that any bending in a CNT changes the electronic structure, in this software, we choose to assume that all bending angles below the critical curvature angle result in a nanotube that has properties identical to a straight nanotube. This may be changed in future iterations of the software. Iijima *et al.* performed many experimental measurements as well as computer simulations and fit an equation (equation **(3)**) that describes the angle at which a CNT will form a kink.[4] This equation was used to determine the maximum curvature that was allowed for the simulation.

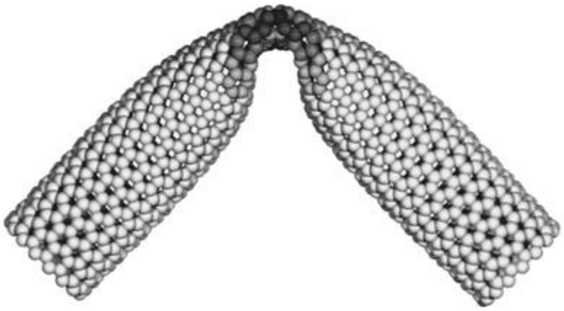


**Fig. 4** A CNT with a kink due to bending.[4]

$$C_c = \frac{1.49}{d_t^{\,2}}\left[1 + \frac{9.89}{d_t^{\,5}} 10^3 \cos\left(6h_g\right)\right] \qquad \textbf{(3)}$$

The results from the equations (1) and (3) are used directly in the construction of the rigid bodies and constraints in the Bullet physics simulation.

## Bullet Physics Integration

This section is concerned with the implementation of CNT parameters using Bullet physics. The topics covered will be diameter, intertube spacing, intra-tube cylinder spacing, and CNT length calculations.

***Diameter:*** The diameter of the CNTs are halved as cylinder objects take radius as a parameter. One simple line of code can create simple cylinder object:

```
btCollisionShape* cylinder = new btCylinderShape(btVector3(radius, h/2, radius));
```

***Intertube Spacing:*** The intertube spacing is completed by adding a spacing cylinder that has a radius equal to a compositional cylinder plus half of the intertube spacing specified in the configuration file. This ensures that when two different nanotubes get near each other, they will at least be separated by the intertube spacing requirement (**Fig. A2)**.

***Intra-tube Cylinder Spacing:*** Since a CNT is a represented by a chain of compositional cylinders, there must be a way to connect each cylinder to another. The simulation connects them by placing a point to point constraint at the half-way point between the two cylinders along their axis. Since the compositional cylinders collide with each other, choosing a particular separation between the two cylinders automatically determines the angle that can form between them. This serves as the basis for how curvature is calculated in the end. With the constraint conditions, the problem we have is the following:

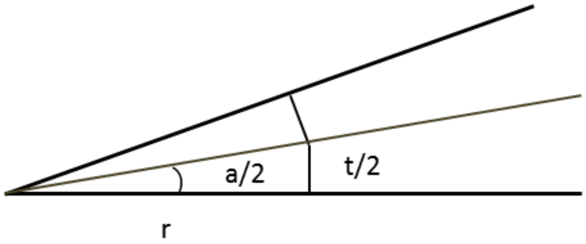


**Fig. 5** Zoomed in section of two cylinders connected by a point to point constraint.

Where ‘a’ is the desired angle, ‘t’ is the intra-tube cylinder spacing, and ‘r’ is the radius of the cylinder. This leads to an expression for finding the intra-tube cylinder spacing.

$$t = 2r \cdot \tan(a/2) \qquad \textbf{(4)}$$

***Length:*** The length calculations are relatively more complicated as each length requires the calculation of a somewhat arbitrary number of compositional cylinders of unknown height. Additionally, each selection of height will change the intra-tube cylinder spacing. The program randomly generates a length and is required to make the following equation true:

$$l = ih + (i-1)\cdot t(C_c, r, h) \tag{5}$$

In equation **(5)**, ‘l’ is the desired length of the CNT, ‘i’ is an integer greater than zero representing the number of compositional cylinders to be used in the CNT, ‘h’ is the height of each compositional cylinder, ‘$C_c$’ (written as ‘c’ from now on) is critical curvature from equation **(3)**, and ‘r’ is the radius of the current nanotube. The ‘t(c,r,h)’ term is the intra-tube cylinder spacing being added as part of the overall CNT length. From this point on, ‘t(c,r,h)’ will be written as ‘t’. The intra-tube cylinder spacing calculation changes dramatically when curvature requirements are imposed. Since curvature requirements are enforced by using discrete objects, curvature will be calculated by the following:

$$c = \frac{a}{2h+t} \tag{6}$$

With equation **(4)** and equation **(6)**, the following manipulations can be made:

$$\begin{gathered}
c\cdot(2h+t) = a \\
2\tan^{-1}\left(\frac{t}{2r}\right) = a \\
2\tan^{-1}\left(\frac{t}{2r}\right) = c\cdot(2h+t) \\
2\tan^{-1}\left(\frac{t}{2r}\right) = 2ch+ct \\
2\tan^{-1}\left(\frac{t}{2r}\right) - ct = 2ch \\
\therefore \frac{\tan^{-1}\left(\frac{t}{2r}\right)}{c} - \frac{t}{2} = h
\end{gathered}$$

Looking at this expression, it is clear that ‘t’ cannot be solved for analytically due to the inverse tangent in the expression. It is true that if the expression can be written as *f(t) = 0*, a numerical fixed point method can be used to find ‘t’. Attempting to find this expression yields

$$f(t) = \frac{\tan^{-1}\left(\frac{t}{2r}\right)}{c} - \frac{t}{2} - h = 0 \tag{7}$$

Newton’s method, which guarantees quadratic convergence to a solution if the expression can be written as *f(t) = 0*, requires the following expression to be used during iterations.

$$g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)} = x_{n+1} \tag{8}$$

The expression for *f(t)* has already been determined and the derivative of that function is

$$f'(t) = \frac{1}{c}\frac{2r}{t^2 + 4r^2} - \frac{1}{2} \tag{9}$$

With both the function and its derivative, *g(t$_n$)* is readily found.

$$g(t_n) = t_n - \frac{\dfrac{\tan^{-1}\left(\dfrac{t_n}{2r}\right)}{c} - \dfrac{t_n}{2} - h}{\dfrac{1}{c}\dfrac{r}{t_n^{\,2} + r^2} - \dfrac{1}{2}} = t_{n+1} \tag{10}$$

Using this fixed point method with an initial guess of *0* will yield the correct answer to ten decimal places. Even though a fixed point method must be used, the intra-tube cylinder spacing can be treated as a function as seen in equation **(5)**.

Now that the intra-tube cylinder spacing can be calculated quickly and easily, an examination of equation **(5)** points to the next step towards creating a CNT of the correct length. Before determining the exact height of each compositional cylinders, the number of cylinders must be known. To solve for the number of cylinders, 'i', equation **(5)** will be rewritten as:

$$i = \frac{l + t(c,r,h)}{h + t(c,r,h)} \tag{11}$$

It is clear that since h, and therefore t, are not known that this equation cannot be solved and a solution must be designed. First, the intra-tube cylinder spacing increases monotonically with increasing height of compositional cylinders. This implies that as 'h' increases in equation (11), 'i' decreases. The algorithm that finds a suitable 'i' value iterates with a starting minimum height of one nanometer and increases the height in small increments until the decimal of the current 'i' value is greater than the previous iteration's decimal. This 'i' value is then rounded up to the nearest integer. This yields the number of compositional cylinders needed to achieve the specified CNT length while also keeping the height of the compositional cylinders as close to the minimum height as possible.

Since the 'i' value is rounded up to the nearest integer, the height of the compositional cylinder is not exactly known. However, in equation **(5)** the length and the number of compositional cylinders are now known. Once again, Newton's fixed point method (equation **(8)**) is used. This time equation **(11)** is rewritten to be in the *f(h)=0* form.

$$f(h) = \frac{l + t(c,r,h)}{h + t(c,r,h)} - i = 0 \tag{12}$$

Unfortunately equation **(12)** is not differentiable analytically so numerical differentiation is used. The expression for Newton's fixed point method is:

$$g(h_n) = h_n - \frac{\left(\dfrac{l + t(c,r,h_n)}{h_n + t(c,r,h_n)} - i\right)}{f\,'(h_n)} = h_{n+1} \tag{13}$$

Solving for the height using equation **(13)** allows for an accurate height of the compositional cylinders which can then be used to get an accurate intra-tube cylinder spacing.

With all of the parameters now defined for a CNT of a particular chirality and length, the software connects all of the components to make the objects necessary for a simulation.

## Limitations/Inaccuracies

This section covers some of the limitations and inaccuracies of the current version of the software.

***Iijima's Equation:*** Equation **(3)**, while used to determine how much the nanotubes can bend, is only valid for CNTs with diameters between 1.0 nm and 1.5 nm. For the chiralities of interest, (7,5), (7,6), (8,6), (8,7), and (9,7), it is clear that some of the diameters are outside of the range of the equation. Currently, all of the diameters are entered in the expression as is. This may be dangerous because the out-of-range nanotubes have diameter < 1.0 nm and thus have larger allowed curvatures. Assuming a larger curvature may be less safe than opposite as bending causes differences in electrical properties. The next versions of the code may include a curvature limit for smaller diameter tubes as a smaller curvature should create less error when using this software's results with Amirhossein's outputs.

***General Physical Parameters:*** The purpose of this project was to get a random mesh of CNTs. The software accomplishes this, but does not claim to be perfect. If the reader has read the manual, it is clear that CNT-CNT repulsions and attractions do not exist. Also, gravity and friction also do not have physical values. Another inaccuracy is due to the complete disregard for the lattice atoms in each CNT. Lastly, CNT outputs do not provide any information about the location of CNT lattice atoms as each CNT is approximated by many cylinders.

***Dynamics Output:*** In the interest of speed, Bullet physics uses only single-precision numbers. Additionally, while the collision detection algorithms are the best known to be publicly available, the solutions are not perfect. For example, a ball at rest on the ground will be continuously pulled by gravity causing a collision with the ground every single simulation step. For a collision to happen, the ball must be going through the ground slightly, which is not physically correct. These are the types of errors to expect with collision detection. If comparing output to a real world situation, errors from the dynamics engine will be much smaller than the errors accumulated by inaccuracies outlined in the general physical parameters section.

***Errors Due to Input:*** The current version of the software does not limit the CNT lengths or device region sizes. The software may not behave as desired if the lengths of the CNTs are much too large for the device region. In this case, the simulation should still end, but the CNTs will be partially in a structure responsible for funneling all of the CNTs into the device region. A user can check if their sizes are appropriate by rendering the simulation (refer to Table 1) and watching for unwanted behavior.

***Constraint Errors:*** When the simulation is running, there may be a situation that places large stresses between two compositional cylinders and separates them more than specified by the intra-tube cylinder spacing parameter. These circumstances, while rare, almost always resolve themselves. This is because the CNTs must be at rest enough for the simulation to be considered complete. If these stresses are present, the simulation acts to resolve the issue with the constraints

A serious constraint error can occur if the user interferes with the simulation. If a box is shot at a high speed, or the CNTs are moved abruptly by grabbing them with the mouse, constraints can be ruined beyond repair. In this situation, hit the space bar to reset the simulation.

## Appendix A

CNT Simulation Representation

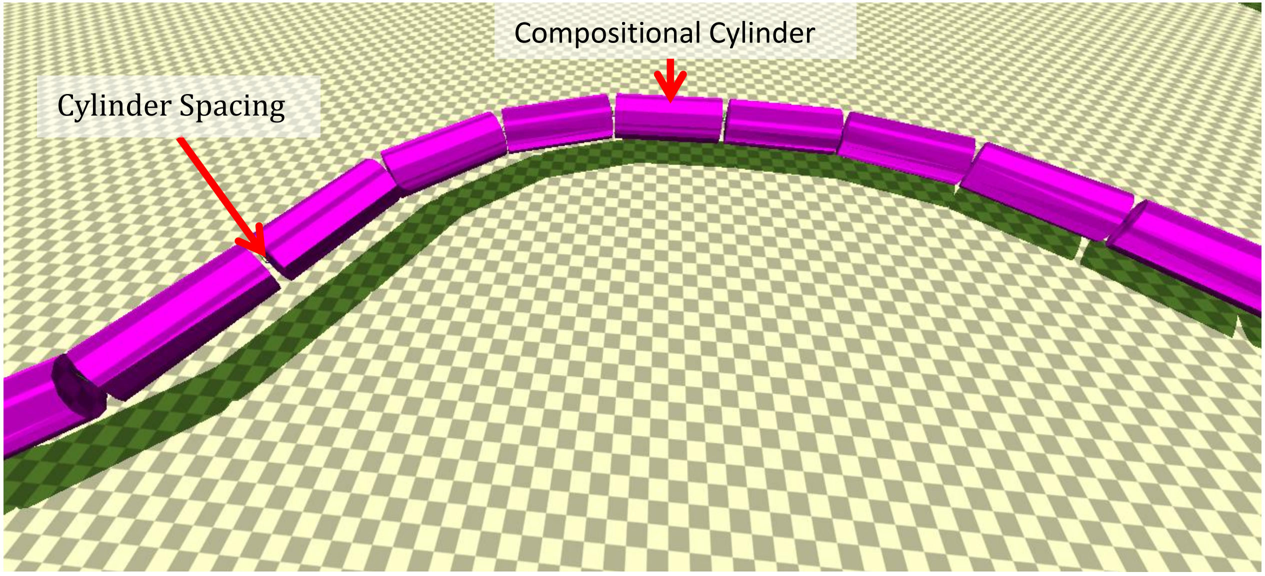


**Fig. A1** Texture view of CNT with labels.

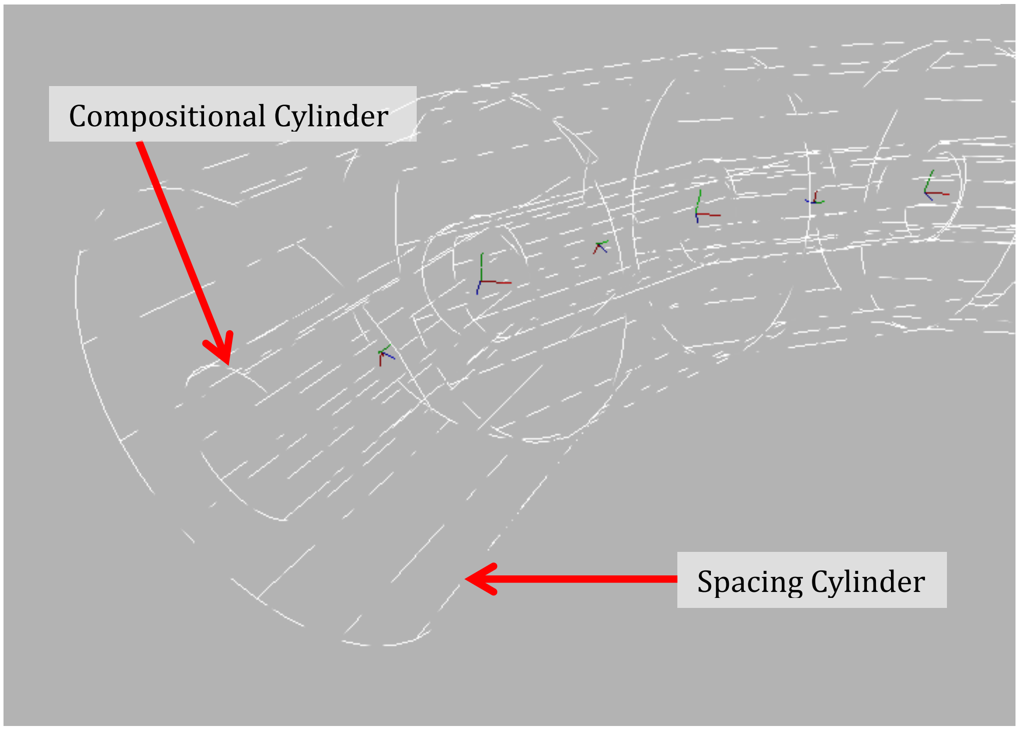


**Fig. A2** Wireframe debug mode with labels.

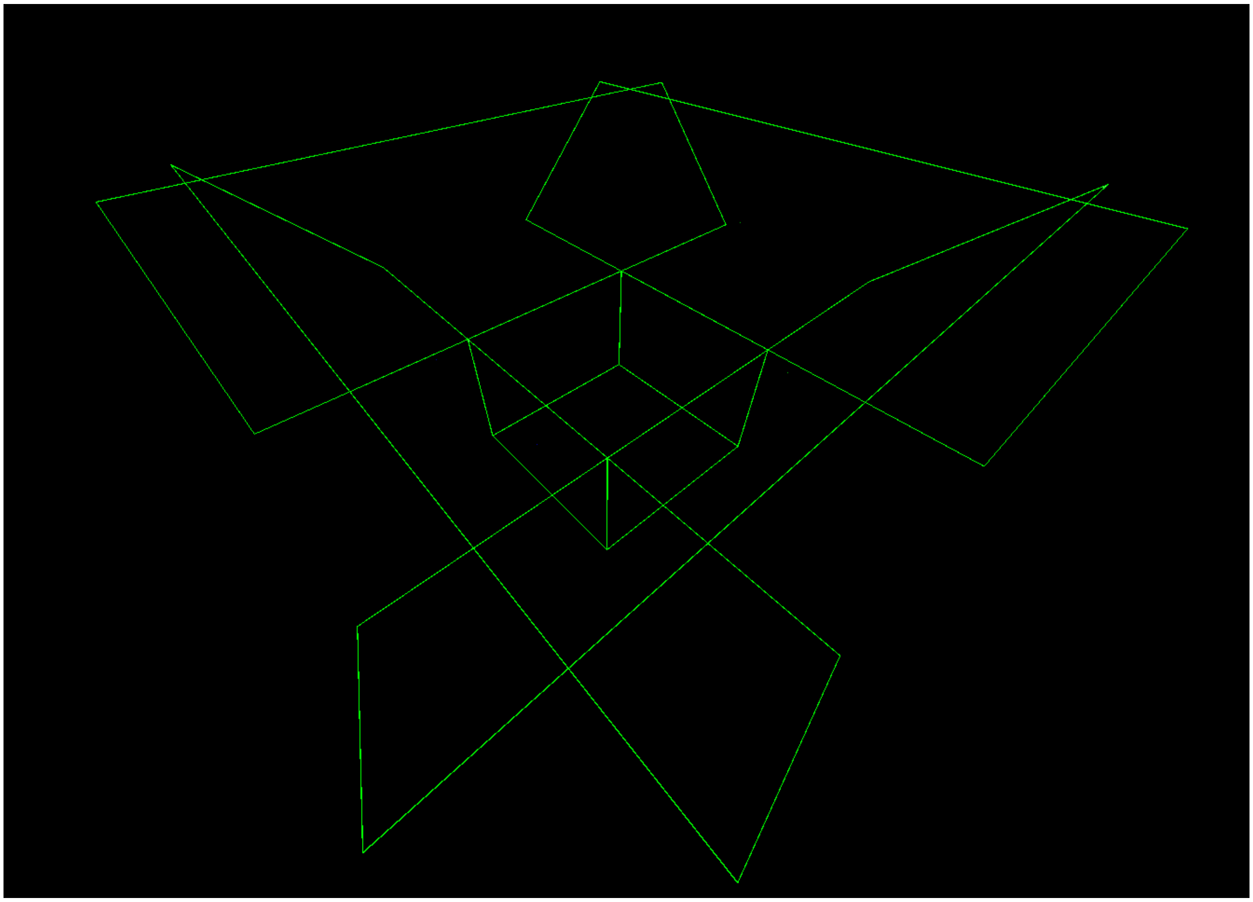


**Fig. A3** Wireframe view of device region with a rectangular funnel on top. The funnel dimensions are based on max CNT lengths to ensure that they all end up in the device region.

## References

[1] M. S. Arnold, J. L. Blackburn, J. J. Crochet, S. K. Doorn, J. G. Duque, A. Mohite, and H. Telg, "Recent developments in the photophysics of single-walled carbon nanotubes for their use as active and passive material elements in thin film photovoltaics.," *Phys. Chem. Chem. Phys.*, vol. 15, no. 36, pp. 14896–918, 2013.

[2] A. Davoody and I. Knezevic, "Theoretical study of exciton dynamics in carbon nanotube composites," Poster presented at ICPS, Zurich, Switzerland, 2013 .

[3] L.-C. Qin, "Determination of the chiral indices (n,m) of carbon nanotubes by electron diffraction.," *Phys. Chem. Chem. Phys.*, vol. 9, pp. 31–48, 2007.

[4] S. Iijima, C. Brabec, a Maiti, and J. Bernholc, "Structural flexibility of carbon nanotubes," *J. Chem. Phys.*, vol. 104, no. 1996, pp. 2089–2092, 1996.

[5] K. Suzuki, M. Ohnishi, and H. Miura, "Change in Electronic Properties of Carbon Nanotubes Caused by Local Distortion under Axial Compressive Strain," presented at SISPAD , Washington, DC, USA, 2015.

**Additional Credits**

A special thanks must go to the Bullet Collision Detection and Physics Library software (http://bulletphysics.org) as it has given me the tools to complete the project in a reasonable amount of time.

Also, thanks to rapidxml (http://rapidxml.sourceforge.net/) for making my xml processing very easy and fast.