

3D Molecular Geometry and Band Structure Modelling of Graphene and Carbon Nanotubes

C.S. Rupasinghe, M.R.M.Mufthas, S.Perera

Lanka Software Foundation

Colombo, Sri Lanka

mufthas@opensource.lk,chanakas@opensource.lk, hemapani@opensource.lk

Abstract—Graphene, carbon nanoribbons and carbon nanotubes are allotropes of carbon that are widely used in nanotechnology research due to their remarkable properties. Electrical and mechanical properties of those allotropes vary with their molecular geometry. Theoretically, an infinite number of such geometries do exist, and it is impossible to physically prepare samples of all of them in the real world. This paper is based on developing mathematical models to describe molecular geometries of graphene and carbon nanotubes, examining their geometry dependent electronic band structures and visually representing those relations in a 3D modelling and simulation software. Paper describes equations to derive the coordinates of the atoms, algorithms to derive the orientation of the bonds of the molecular structures, implementation of those algorithms in java3D and derivation of electronic band structures. Successful results were obtained after the implementation of the algorithms which are illustrated with few screenshots at the end.

Keywords – Graphene, Carbon nanotubes, 3D modelling, java3D, Band structure

I. INTRODUCTION

Nanotechnology deals with objects that are at nanometer scale and are not visible to the naked eye or even to powerful optical microscopes. Therefore, virtual reality, which modern computer technology has brought within our reach, plays a major role in modeling and simulating the nano-world. Carbon nanotubes exhibit large variations in their molecular geometries, which in turn lead to large variations in their electrical and mechanical properties [1]. An infinite number of such geometries do exist, and it is not feasible to prepare samples of all of them in the real world. Therefore having a 3-dimensional visualization of the atomic structure of the relevant geometries in a virtual environment will greatly enhance our ability to understand the correlations between material properties and the geometry [2].

Graphene is a layer of carbon atoms of graphite, and carbon nanoribbon is a rectangular strip of graphene that is cut as defined by two particular parameters: ‘chirality and length’. When the nanoribbon is rolled to form a tube, it is known as a carbon nanotube. The modeling tool described in this paper is capable of generating the 3D visualization of the nanostructures according to user parameters [3]. It can be also used to plot the electronic band structures of them. The user can also change the preview of molecular geometries and graph plots interactively by using the controllers on the interface.

II. LITERATURE SURVEY

Interests have been grown on nano scale allotropes of carbon since the inventions of fullerenes by Kroto et al [16] in 1985 and carbon nanotubes by Iijima in 1991 [17]. Those nanostructures are formed as a result of the high energy atoms with dangling bonds at the edges of graphene layer which is an atom thick layer of graphite. Those high energy atoms are likely to make new bonds with the atoms at other edges by forming closed cages or tubes of carbon atoms [1].

The way of rolling the graphene layer makes different types of nanotubes. Hence, there exist many possible orientations of hexagons made of carbon atoms for nanotubes and therefore, many different molecular geometries are available for them [18]. Since, those different molecular geometries exhibit different electrical properties, scientists focused on theoretical modelling of their electronic band structure and examining their relations to the molecular geometry. It is found that band structure of nanotubes are closely related to the band structure of graphene which is a three dimensional plot [7]. Band structure of a particular nanotube with a given orientation can be formed of the cross-sections of graphene band structure in a direction derived from that orientation [15]. Thus, nanotubes have a two dimensional plot as the band structure.

Although the geometries of carbon nanotubes are very important and some tools [19] are available to plot them, no attempt is found to rigorously describe the geometries of graphene and carbon nanotubes based on proper mathematical models. This paper is focused on filling that gap by generally describing those geometries with equations and algorithms in addition to implementing already established models for the band structure. Further, the tool Ninithi [20], which is based on this paper clearly illustrates the relationships between molecular geometry and electronic band structure of carbon nanotubes.

III. GRAPHENE

Graphene, which is simply a layer of graphite that is only one atom in thickness, is one of the most important and widely used carbon materials in nanotechnology research. Structure of the planer layer is a honeycomb lattice, and hybridization of the carbon is sp^2 . Graphene is used in many researches due to its remarkable electrical properties, particularly in nanoelectronics [4]. To draw the molecular

structure of graphene, we should first calculate coordinates of the atoms, and next we can derive the bonds among them.

A. Deriving atom coordinates

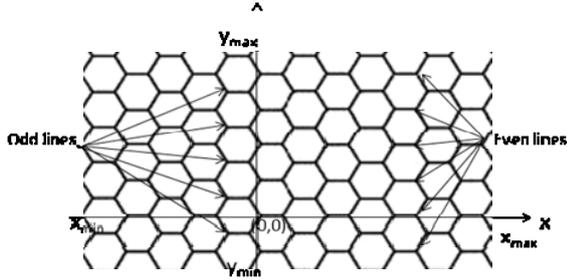


Figure 1. Atom positions of a graphene layer

The locations of the atoms in graphene layer are shown in Fig.1. It is observable that there are two types of carbon atoms in the layer as illustrated in Fig.2.

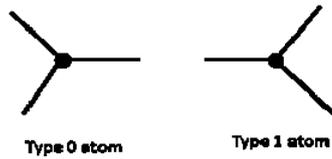


Figure 2. Type 0 and Type 1 atoms

The convention adopted in the paper is to derive the coordinates of the atoms with respect to the origin, which is a Type 0 atom. The entire Graphene layer can be considered as a collection of atom lines. There are two types of such atom lines. The line through the origin is referred to as an 'evenline' whereas other type of lines are referred to as 'odd lines'. Odd lines can be obtained by shifting an even line to the right and they are positioned alternately.

The set of x-coordinates of an even line X_{even} is given by the equation (1) below.

$$X_{even} = \left\{ x_n^+ \left| \begin{array}{l} x_n^+ = \left(x_{n-1}^+ + \frac{2}{x_{n-1}^+ - x_{n-2}^+} \right) a_{cc} \\ \text{where } x_1^+ = 0, x_2^+ = 1, n = 1, 2, 3, \dots \text{ and } x_n^+ \leq x_{max} \end{array} \right. \right\} \cup \left\{ x_n^- \left| \begin{array}{l} x_n^- = \left(x_{n-1}^- - \frac{2}{x_{n-2}^- - x_{n-1}^-} \right) a_{cc} \\ \text{where } x_1^- = -2, x_2^- = -3, n = 1, 2, 3, \dots \text{ and } x_n^- \geq x_{min} \end{array} \right. \right\} \quad (1)$$

Where a_{cc} is the bond length between two carbon atoms in Graphene and x_{min}, x_{max} are as marked in Fig.1.

Since x-coordinates of odd lines can be obtained by shifting an even line by $a_{cc} + a_{cc} \cos(\pi/3)$ to the right, the set of x-coordinates of odd lines, X_{odd} is given by (2) below:

$$X_{odd} = \left\{ x_{odd} \left| x_{odd} = \left(x_{even} + a_{cc} + a_{cc} \cos\left(\frac{\pi}{3}\right) \right) \right. \right\} \quad (2)$$

The set of y-coordinates that are belonged to even lines, Y_{even} is given by (3).

$$Y_{even} = \left\{ y_n^+ \left| \begin{array}{l} y_n^+ = 2na_{cc} \sin\left(\frac{\pi}{3}\right) \\ \text{where } n = (0, 1, 2, \dots) \text{ and } y_n^+ \leq y_{max} \end{array} \right. \right\} \cup \left\{ y_n^- \left| \begin{array}{l} y_n^- = -2na_{cc} \sin\left(\frac{\pi}{3}\right) \\ \text{where } n = (1, 2, 3, \dots) \text{ and } y_n^- \geq y_{min} \end{array} \right. \right\} \quad (3)$$

The set of y-coordinates for odd lines, Y_{odd} is also can be obtained by shifting the set Y_{even} by $a_{cc} \sin(\pi/3)$ upwards. The set of y-coordinates for odd lines is given by (4) below:

$$Y_{odd} = \left\{ y_{odd} \left| y_{odd} = \left(y_{even} + a_{cc} \sin\left(\frac{\pi}{3}\right) \right) \right. \right\} \quad (4)$$

Hence the set of coordinates of atoms as a set of (x,y) ordered pairs is given by,

$$G = (X_{odd} \times Y_{odd}) \cup (X_{even} \times Y_{even}) \quad (5)$$

In the program, G is taken into a 2D array, 'AtomCoordinates' of size $|G| \times 3$ where,

$$\text{AtomCoordinates}[i] = \{x_i, y_i, 0\} \quad \forall (x_i, y_i) \in G$$

Here, z-coordinate is taken as zero. Another constant can also be used as the z-coordinate. The method 'GetAtomCoordinates()' of class 'Graphene' returns the array 'AtomCoordinates' to fulfill the requirements of the input format of the 3D structure rendering thread.

B. Algorithm to derive the bonds

Every atom has three bonds except for the atoms located on the edges of the layer, and every bond has to be drawn joining two atoms. Three neighbor atoms are located in the same atom line, the line above, and the line below. The Following steps would derive the bonds.

1. Start numbering atoms from the leftmost atom of the lowest atom line as atom-1 and put the atom lines in to an array
2. Declare an array of 'atoms' that is an array of parameters {'atom number', 'xyz-coordinates', 'same line neighbor', 'below line neighbor', 'above line neighbor'} and a linked list of 'bonds' that has as parameters atom numbers and their atom coordinates
3. For atom-1, declare the neighbor atom in the same line as next atom, atom-2 if $x_2 - x_1 = a_{cc}$, where x_1, x_2 denotes the x-coordinates of atom-1 and atom-2 respectively. if $x_2 - x_1 \neq a_{cc}$, go to step 6
4. Then, also declare the neighbor atom in the same atom line of the atom-2 as atom-1.

5. Add the bond between atom-1 and atom-2 to the 'bonds' linked list and update the array 'atoms'.
6. Repeat the steps 3, 4, 5 for all atoms of the graphene layer
7. For atom-1, find the neighbor atom-*k* in the line below if $|x_1 - x_k| = a_{xx} \cos(\pi/3)$ where x_1, x_k denotes the x-coordinates of atom-1 and atom-*k* respectively. If there is no atom line below, go to the step 10.
8. Then, also declare the neighbor atom in the atom line above of the atom-*k* as atom-1.
9. Add the bond between atom-1 and atom-*k* to the 'bonds' linked list and update the array 'atoms'.
10. Repeat the steps 7, 8, 9 for all atoms of the graphene layer

Size of the 'bonds' linked list is the number of bonds in the Graphene layer. To get the bond coordinates into an array, 'BondCoordinates', of size number of bonds \times 6 which can be declared as,

$$\text{BondCoordinates}[i] = \{x_{i1}, y_{i1}, z_{i1}, x_{i2}, y_{i2}, z_{i2}\}$$

where, (x_{i1}, y_{i1}, z_{i1}) and (x_{i2}, y_{i2}, z_{i2}) are the coordinates of the atoms connected by the i^{th} bond. The method 'GetBondCoordinates()' of class 'Graphene' returns the array 'BondCoordinates'.

IV. CARBON NANORIBBONS

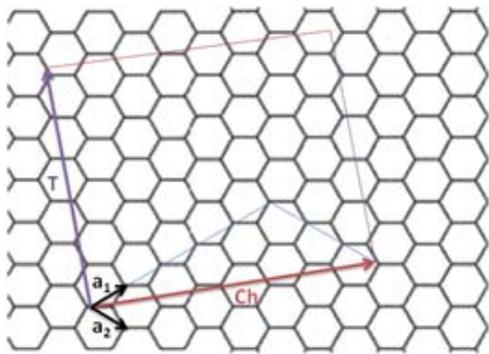


Figure 3. Carbon nanoribbon is marked by the rectangle derived from the chirality vector and the length. a_1 and a_2 as vectors.

A carbon nanoribbon can be described as a cut of a Graphene in a particular structure according to the parameters, 'Chirality vector' and the length. The Chirality is given by a pair of integers (n, m) and the Chirality vector, Ch is given by

$$Ch = na_1 + ma_2 \tag{6}$$

where a_1 and a_2 vectors are given as,

$$a_1 = \left(\frac{\sqrt{3}a_{cc}}{2}, \frac{\sqrt{3}a_{cc}}{2}\right) \text{ and } a_2 = \left(\frac{\sqrt{3}a_{cc}}{2}, -\frac{\sqrt{3}a_{cc}}{2}\right) \tag{5}$$

T is the vector that is perpendicular to Ch and the length of the nanoribbon is given by $|T|$. If $m=n$, the nanoribbon is called as an 'armchair nanoribbon' and if m or $n = 0$, nanoribbon is known as a 'zigzag nanoribbon'.

If any atom that has the coordinates $(x, y) \in G$ is located inside the rectangle with vertices $((0,0), Ch, Ch+T, T)$, then,

$$(x, y) = \alpha Ch + \beta T \tag{7}$$

where $\alpha, \beta \in [0,1]$

A. Deriving atom coordinates

Hence the set of coordinates of the carbon nanoribbon, CNR is given by

$$CNR = \{(x, y) | (x, y) \in G \text{ and } \alpha, \beta \in [0,1]\} \tag{8}$$

In the program CNR is taken into a 2D array, 'AtomCoordinates' of size $|CNR| \times 3$ where,

$$\text{AtomCoordinates}[i] = \{x_i, y_i, 0\} \forall (x_i, y_i) \in CNR$$

Here, z-coordinate is taken as zero. Another constant also can be used as the z-coordinate. The method 'GetAtomCoordinates()' of class 'CarbonNanoRibbon' returns the array 'AtomCoordinates'.

B. Deriving bond coordinates

Since $CNR \subseteq G$, bonds of the nanoribbon can be easily obtained by removing all the bonds connected to the atoms located outside the nanoribbon. If coordinates of both the atoms follow the relation $\alpha, \beta \in [0,1]$, the bond belongs to the nanoribbon too.

The array 'BondCoordinates' is obtained by reducing the 'BondCoordinates' array of Graphene. The method 'GetBondCoordinates()' of class 'CarbonNanoRibbon' returns the reduced array.

V. CARBON NANOTUBES

Carbon nanotubes are formed when a carbon nanoribbon is rolled around an axis parallel to T . Therefore parameters for the carbon nanotubes are same as for the carbon nanoribbons, n, m and length [6].

A. Deriving atom coordinates

$|Ch|$ is the circumference of the nanotube and therefore its radius, r is given by $|Ch|/2\pi$. The set of coordinates of carbon atoms of the nanotube CNT is given by,

$$CNT = \left\{ \begin{array}{l} x = r \cos \left(2\pi \frac{(a, b) \cdot Ch}{|Ch|^2} \right) \\ y = r \sin \left(2\pi \frac{(a, b) \cdot Ch}{|Ch|^2} \right) \\ z = \frac{(a, b) \cdot T}{|T|} \\ \forall (a, b) \in CNR \end{array} \right\} \quad (9)$$

In the algorithm, CNT is taken into a 2D array, ‘AtomCoordinates’ of size $|CNT| \times 3$ and it is returned by the method ‘GetAtomCoordinates ()’ of the class ‘CarbonNanoTubes’.

B. Deriving bond coordinates

Since the nanotube is formed by rolling the nanoribbon [7], the bonds of the nanoribbon still exist though atom coordinates are changed. In addition, a new set of bonds is formed due to rolling.

To find the bonds coordinates, which originally existed in the nanoribbon, we replace coordinates of the atoms in the ‘bonds’ linked list of the nanoribbon with new coordinates of the nanotube referred by the corresponding atom numbers. Then, we create the ‘bonds’ linked list of the nanotube.

Bonds originated due to rolling effect are formed between atoms in the two leftmost edges, and the two rightmost edges of the nanoribbon. For those atoms, if the atom-*i* is in the left and atom-*j* is in the right and the difference between their z- coordinates is not greater than a_{cc} and also if,

$$|(x_i, y_i, z_i) - (x_j, y_j, z_j)| - a_{cc} < 0.05 \quad (10)$$

atom-*i* and atom-*j* are neighbor atoms and the bond should be added to the ‘bonds’ list. The array ‘BondCoordinates’ is returned by the ‘GetBondCoordinates()’ method of the ‘CarbonNanoTube’ class.

VI. IMPLEMENTATION OF MOLECULAR GEOMETRIES USING JAVA3D

Java3D is an application programming interfaces (API) that can be used to visualize 3D objects in Java. 3D programming with Java3D is very easy to understand and less complex than Open Graphics Library (OpenGL) and Java Open Graphics Library (JOGL). Java3D is a scene graph based API for the Java platform, and it runs on top of both OpenGL and Microsoft DirectX [9].

Under Java3D, users can describe a 3D virtual world or scene in terms of a variety of objects arranged in a tree like graph called the scene-graph. These objects can be visual objects, user interaction objects, sound objects etc. A simple application can be rapidly created, even without knowing the all the concepts behind Java3D, using the Java3D class called ‘SimpleUniverse’. An object of ‘SimpleUniverse’ can be constructed with default values for the important parameters and nodes. We use a ‘SimpleUniverse’ object to visualize all the 3D models. Orbit behavior can be added to

the application to visualize the 3D models in different views and angles.

We draw different aspects of visualization with Java3D in the following manner.

A. Drawing atoms

An atom can be drawn using ‘Sphere’ class in Java3D by specifying the radius, appearance etc [10]. By default, it is centered at the origin. Therefore the ‘Transform3D’ class is used to translate the center of sphere to new location. ‘TransformGroup’ class can be used to integrate various compatible objects in Java3D. Translated sphere can be obtained using the object of ‘TransformGroup’. This transformgroup object has two separate objects. The first is translation of the center of the sphere from origin to a new location and the other is sphere with given radius, appearance etc.

Therefore an atom can be drawn by specifying the coordinates of the center, radius of the sphere, appearance of the sphere etc. Similarly, by adding atom by atom to another new ‘TransformGroup’ object, a cluster of atoms can be drawn.

B. Drawing bonds

A bond is simply a cylinder. A Cylinder can be drawn using ‘Cylinder’ class in Java3D. A Cylinder is defined by its radius and height. It is a capped cylinder centered at the origin with its central axis aligned with the Y-axis. The cylinder constructor accepts radius, height, appearance etc [11]. Here, the bond should be drawn between two given points. In Java3D, a cylinder cannot be drawn between two given points directly. The Following steps can be identified to draw a bond between two given points [12].

1. Translating the center of the cylinder to the middle point of given two points.
2. Finding the angle of rotation.
3. Finding the axis of rotation.

Let’s assume, the end points of the bond are $P_1 \equiv (x_1, y_1, z_1)$ and $P_2 \equiv (x_2, y_2, z_2)$. Then the height of the cylinder can be calculated by finding the distance between the two points,

Height of the cylinder =

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2} \quad (11)$$

Radius of the cylinder is a pre-defined value and can be changed by the user with the interface.

Then the center of the cylinder should be translated to the middle point of P_1 and P_2 .

New center of the cylinder

$$\equiv \left(\frac{x_1 + x_2}{2} \right), \left(\frac{y_1 + y_2}{2} \right), \left(\frac{z_1 + z_2}{2} \right) \quad (12)$$

Finally, the angle of rotation and the axis of rotation should be calculated.

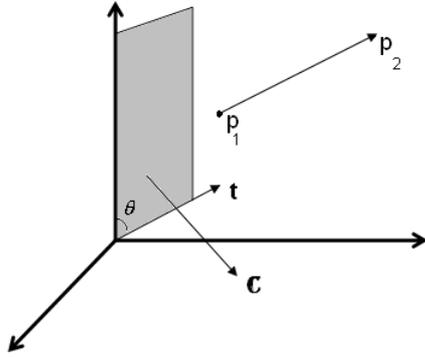


Figure 4. Rataional angle (θ) and axis (C)

The unit vector in the direction of the line connecting P_1 and P_2 , t is given by (13).

$$t = \frac{P_1 - P_2}{|P_1 - P_2|} \tag{13}$$

And angle of rotation, $\theta = \cos^{-1}(t \cdot j)$ (14)

where j is the unit vector along the y-axis.

The axis of rotation should be perpendicular to the plane of rotation.

Therefore, the axis of rotation, $c = t \times j$ (15)

‘Vector3f’ class can be used to represent the vectors in Java3D. First of all, the unit vector in the direction of the line connecting two points should be represented by a vector3f object. Cross vector of Y- axis and unit vector in the direction of the line connecting two points should be calculated and assigned to a vector. Then we set the angle of rotation about the axis of rotation to an object of ‘AxisAngle4f’, and this axisangle4f object to an object of ‘Transform3D’. Let us name this transform3D object as v1.

The we translate the center of the cylinder to a new location using an object of ‘Transform3D’, and then multiply this object by transform3D object v1 and add the result to an object of ‘TransformGroup’. Subsequently, we construct the cylinder with a given radius, calculated height, custom appearance etc. Finally, by adding this cylinder to the above transformgroup object, a bond can be constructed and by adding a bond by bond to another new TransformGroup object, the bunch of bonds can be drawn.

C. Visualizing a model

Models such as graphene, nanoribbon, nanotube and fullerene can be drawn easily with our customized atoms and bonds. To draw atoms, coordinates of the center should be obtained. To draw bonds, coordinates of two points should be obtained. Transformgroup of both atoms and bonds can

be obtained using the coordinates of atoms and bonds. Then, in the program, an object of ‘BranchGroup’ is created and both atoms transformgroup and bond transformgroup are added to branchgroup object. Finally the branchgroup object is added to the ‘SimpleUniverse’ object.

VII. ELECTRONIC BAND STRUCTURES

Electronic band structure of a material represents the possible energies for electrons to possess. It helps to understand the electrical and optical properties of a material. Valence and conduction bands are important parameters to derive the band gap, which is the difference between bottom of conduction band and top of the valance band. Bandgap is zero for metals where semiconductors have different non-zero values [13]. Band structure plots also help to identify direct bandgap and indirect bandgap semiconductors, which are two major types of semiconductors.

In electronic band structure, the energy of each band is plotted over the Brillouin zone, which is the primitive cell of the reciprocal space lattice of the material. Energy band plots over Brillouin zone represent all the solutions for energies which electrons are allowed.

Fig. 5(a) shows the real lattice space of graphene layer, which is formed by the repetition of A,B atoms pair with a_1, a_2 primitive lattice vectors. Shaded box is the primitive unit cell. Fig. 5(b) illustrates the reciprocal space lattice of graphene and the Brillouin zone as a shaded hexagon. b_1, b_2 are reciprocal lattice vectors. k_x, k_y are the directions along Brillouin zone, and they are known as wave vectors.

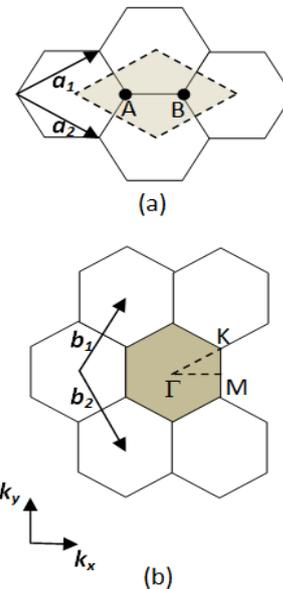


Figure 5. (a) Real space lattice with shaded unit cell and (b) reciprocal space lattice with shaded Brillouin zone

Here, $b_1 = (\frac{2\pi}{\sqrt{3}a}, \frac{2\pi}{a})$ and $b_2 = (\frac{2\pi}{\sqrt{3}a}, -\frac{2\pi}{a})$ (16)

Graphene has a two-dimensional band structure and it can be calculated using the tight-binding energy model and given as a function of k_x, k_y wave vectors in 17 where t is the transfer integral and a is the C-C bond length in graphene [14].

$$E_g(k_x, k_y) = \pm t \left\{ 1 + 4 \cos\left(\frac{\sqrt{3}k_x a}{2}\right) \cos\left(\frac{k_y a}{2}\right) + \cos^2\left(\frac{k_y a}{2}\right) \right\}^{\frac{1}{2}} \quad (17)$$

Hence graphene band structure is a 3D graph plotted on Brillouin zone where k_x, k_y and E_g are the three axes.

Since carbon nanotubes are formed of rolled graphene nano ribbons as depicted in Fig.3, band structure of carbon nanotubes can be derived from graphene band structure and given in 18 [15].

$$E_\mu(k) = E_g\left(k \frac{\mathbf{K}_2}{|\mathbf{K}_2|} + \mu \mathbf{K}_1\right) \quad \text{for } \left(-\frac{\pi}{T} < k < \frac{\pi}{T}, \text{ and } \mu = 1, 2, \dots, N\right) \quad (18)$$

where, $T = \frac{\sqrt{3}|Ch|}{d_R}$ and $N = \frac{2(n^2+m^2+nm)}{d_R}$

m, n and Ch have the same meanings as in 6 and d_R is the greatest common divisor of $(2n + m)$ and $(2m + n)$. \mathbf{K}_1 and \mathbf{K}_2 denote wave vector along circumferential direction and the reciprocal vector along nanotube axis direction respectively as given in 19.

$$\mathbf{K}_1 = \frac{(2n + m)\mathbf{b}_1 + (2m + n)\mathbf{b}_2}{Nd_R} \quad (19)$$

$$\mathbf{K}_2 = \frac{m\mathbf{b}_1 - n\mathbf{b}_2}{N}$$

Therefore, band structure of carbon nano nanotube is a N pairs (for plus and minus) of 2D graphs which are cross sections of 3D graphene band structure along lines denoted by $k \frac{\mathbf{K}_2}{|\mathbf{K}_2|} + \mu \mathbf{K}_1$. There are N pairs of graphs because μ has N values.

VIII. RESULTS

Figure 6 shows the 3D view of a Graphene layer generated by the tool. The dimensions of the layer (width and height) should be given by the user. User is also provided with controllers to change the sizes and colors of the bonds and atoms.

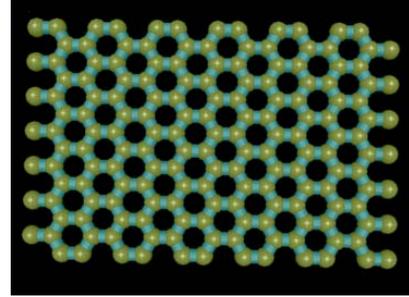


Figure 6. 3D View of a Graphene layer

A 3D view of the nanoribbon is generated when the (n,m) integer pair and the length is provided. Fig.7 shows a screenshots of two nanoribbons generated by the tool.

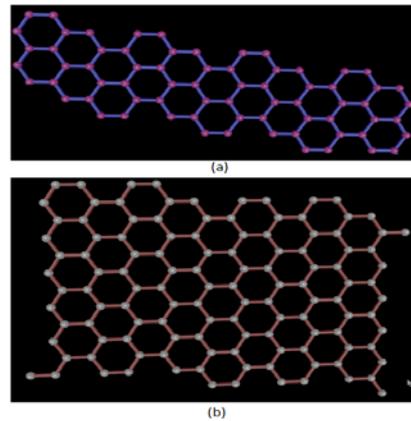


Figure 7. (a) A (4,8) nanoribbon (b) A (4,6) nanoribbon with different lengths

Visualization of the nanotube is the same as the nanoribbon. User can provide the (n,m) pair and the length and model the nanotube in the 3D space. In addition to changing the size and color, user can remove the bonds or atoms. Fig.8 illustrates a nanotube generated by the tool where bonds are removed.

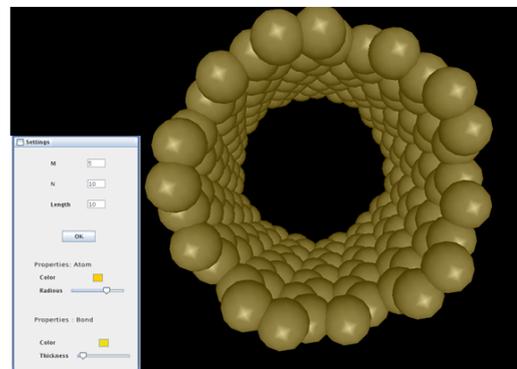


Figure 8. A (10,5) nanotube only with atoms

Chirality of the nanotube can be clearly seen when the viewing angle is along the axis of the nanotube. View of an armchair nanotube ($m=n$), a zigzag nanotube (m or $n=0$) and a chiral nanotube ($m \neq n \neq 0$) are shown in Fig.9. User can drag the nanotube using the mouse and adjust the view as shown in Fig.9(d).

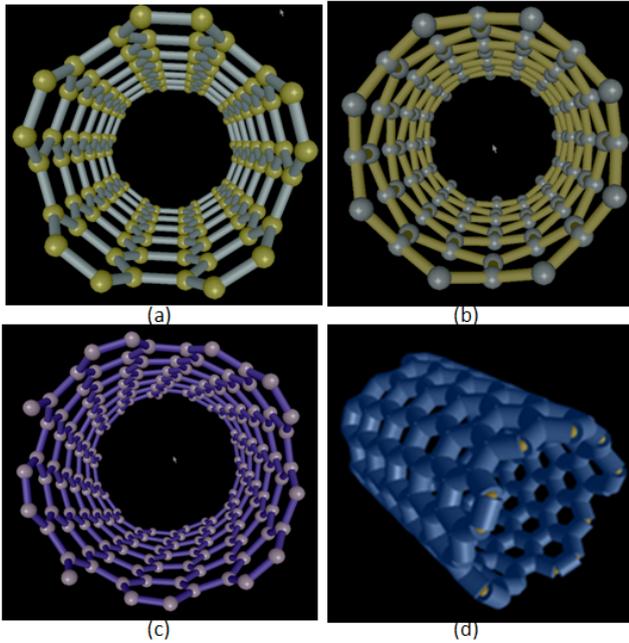


Figure 9. 3D models of (a) a (5,5) armchair nanotube (b) a (8,0) zigzag nanotube (c) a (5,9) chiral nanotube and (d) a nanotube in a different viewing angle.

Graphene band structure is plotted according to 17 and resulted surface is shown in Fig. 10.

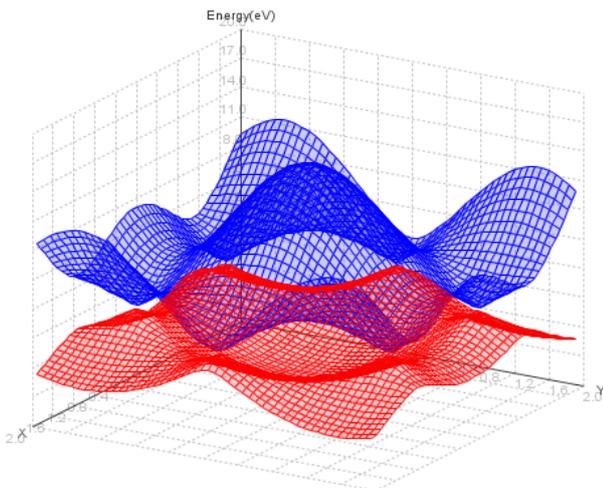


Figure 10. Band structure surface plot of graphene

Only the Brillouin zone of this graph should be considered and it is marked in Fig.11.

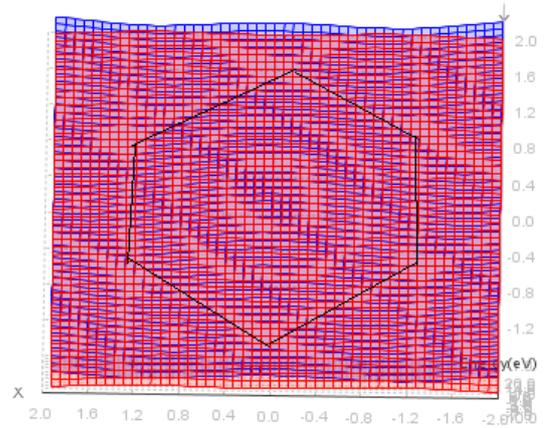


Figure 11. Brillouin zone of graphene marked on surface

Band structure plots of carbon nanotubes are geometry dependent. According to results, carbon nanotubes can be called as direct bandgap materials as there is no difference in k between the lowest of upper band and highest of lower band [13]. Band structure of a (5,5) nanotube is shown in fig.12. This is a metallic nanotube as the bandgap of it is zero.

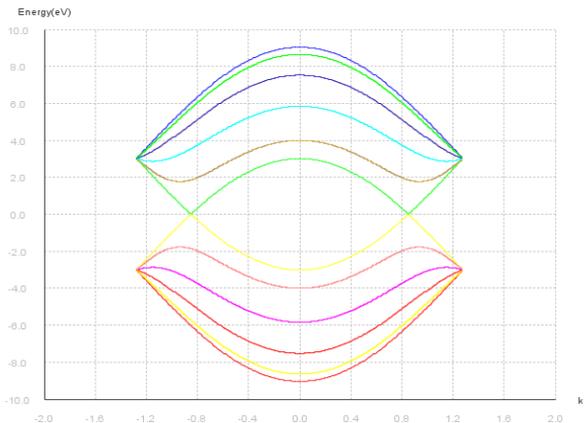


Figure 12. Band structure plot of a (5,5) carbon nanotube

Band structure of a (5,0) carbon nanotube is shown in Fig.13. A bandgap is observable and therefore, this particular nanotube is a semiconductor.

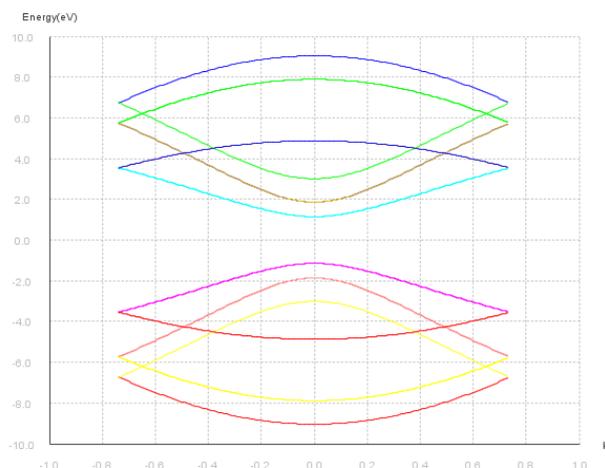


Figure 13. Band structure plot of a (5,0) carbon nanotube

IX. CONCLUSIONS

We have successfully completed the software development, which was done with the objective of modeling the carbon allotropes used in nanotechnology. The generalized equations to represent the atom coordinates of the molecular structures of carbon allotropes as sets of ordered pairs, algorithms to derive their bond coordinates, and the band structure derivations were proven accurate with the obtained results. We have released developed modeling tool as free and open-source software along with additional tools to analyze the electrical properties of modeled allotropes. The tool can be found from [21].

Future work on the tool is focused on developing it as a material library and implementing a modular architecture to make third party contribution much easier. In addition to a material library, there will be tools to design nano systems using the available materials and simulate them.

ACKNOWLEDGMENT

The authors thank Dr.SanjivaWeerawarana, who initiated computational nanotechnology research work in the Lanka Software Foundation, all the staff of the same organization and also the staff of Sri Lanka Institute of Nanotechnology who assisted the development by knowledge and resources.

REFERENCES

- [1] M. S. Dresselhaus, G. Dresselhaus, P. Avouris, "Carbon Nanotubes," New York, pp. 12–27, January 2001.
- [2] A. Appenzeller, E. Joselevich, W. Honlein, "Carbon Nanotubes for Data Processing," in R. Waser ed, "Nanoelectronics and Information Technology," Wiley-VCH (2003)
- [3] Chang, Y. and Z. Wang, 2008. "Research on 3D Visualization of Crystal Molecular Structure Based on Augmented Reality". Proc. International Conference on Computer Science and Software Engineering, Dec. 2008, pp. 1146-1149. doi= <http://dx.doi.org/10.1109/CSSE.2008.503>
- [4] A.H.C.Neto, F. Guinea, N.M.R. Peres, K.S. Novoselov and A.K. Geim " The electronic properties of graphene," Reviews of Modern Physics, vol. 81, no. Jan, pp. 110-113

- [5] J.K. fodor, S.Hong and J.Guo, " Bandstructure of Carbon Nanotubes and Nanoribbons" [Online presentation] 2007, [2009Dec 22], Available at <http://nanohub.org/resources/2762>
- [6] M.Endo, S.Iijima, M. S. Dresselhaus, "Carbon Nanotubes", 1st ed., Pergamon, pp.27–29, 1996
- [7] R. Saito, et. al, "Physical Properties of Carbon Nanotubes", Imperial College Press, 1998
- [8] B. L.Zhang, C. Z.Wang, K. M.Ho, C. H.Xu and C. T.Chan, " The geometry of small fullerene cages - C20 to C70", Journal of Chemical Physics, vol. 97, pp. 5007-5011
- [9] Chi Chung Ko and Chang Dong Cheng, " Interactive web-based virtual reality with Java3D", London: Information Science Reference, pp. 3-8, 2009.
- [10] "Sphere" [2010 Feb 25], <http://download.java.net/media/java3d/javadoc/1.3.2/com/sun/j3d/utills/geometry/Sphere.html>
- [11] "Cylinder" [2010 Feb 25], <http://download.java.net/media/java3d/javadoc/1.3.2/com/sun/j3d/utills/geometry/Cylinder.html>
- [12] P. Pantera, "[JAVA3D] Angle calculation with x y z planes," in java3d-interest, 2003 [2010 Feb 25] <http://www.mail-archive.com/java3d-interest@java.sun.com/msg24712.html>.
- [13] Donald A. Neamen, "Semiconductor Physics and Devices Basic Principles", 3rd ed, McGraw-Hill, 2003
- [14] Changxin Chen and Yafei Zhang, "Nanowelded Carbon Nanotubes: From Field-Effect Transistor to Solar Microcells", in NanoScience and Technology, Springer, 2009
- [15] Saito R. and Kataura H., "Optical Properties and Raman Spectroscopy of Carbon Nanotubes", in Carbon Nanotubes: Synthesis, Structure, Properties and Application, Springer, 2001
- [16] H. W. Kroto, J. R. Heath, S. C. O'Brien, R. F. Curl, R. E. Smalley, Nature(London) 318, 162–163 (1985)
- [17] S. Iijima, Nature (London) 354, 56 (1991)
- [18] R. Saito, G. Dresselhaus, M. S. Dresselhaus, Physical Properties of Carbon Nanotubes (Imperial College Press, London 1998)
- [19] JCrystalSoft, NanotubeModeler, <http://jcrystal.com/#b3>, accessed on 29/10/2011
- [20] C.S. Rupasinghe; M.R.M Mufthas (2010), "Ninithi," <http://nanohub.org/resources/8987>, accessed on 29/10/2011
- [21] Lanka Software Foundation, "Ninithi", <http://ninithi.lk>, accessed on 29/10/2011