Rectangular hexagonal mesh generation for parametric modeling

Patrícia Pereira Pampanelli, João Paulo Peçanha, Alessandra Matos Campos,
Marcelo Bernardes Vieira, Marcelo Lobosco
Departamento de Ciência da Computação-ICE
Universidade Federal de Juiz de Fora,
Juiz de Fora-MG
{*patricia.pampanelli,joaopaulo,amcampos,*
marcelo.bernardes,marcelo.lobosco}*@ice.ufjf.br*

Sócrates de Oliveira Dantas
Departamento de Física-ICE
Universidade Federal de Juiz de Fora,
Juiz de Fora-MG
dantas@fisica.ufjf.br

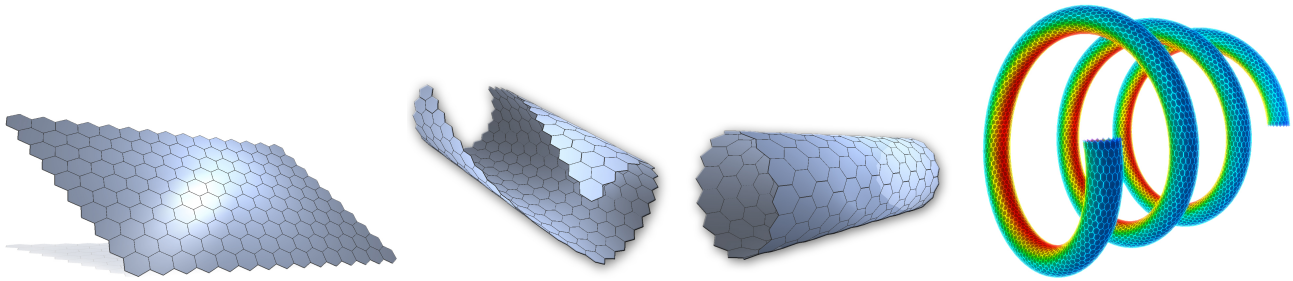


Figure 1. Steps to construct a non-planar object using a bidimensional hexagonal grid.

***Abstract*—This paper addresses the problem of rectangular hexagonal mesh generation for surface modeling. Hexagonal grids are important for several applications such as the simulation of carbon nanostructures. A fast and accurate method to obtain a rectangular region over a hexagonal grid is proposed. Strips of hexagons defined by the Hamada indexes are constructed and stacked using a variation of the Bresenham's algorithm. The final hexagonal mesh is tessellated using half-edges in order to be suitable for several operations and simulation processes. This mesh can be used as a parameter space for obtaining different geometric models. Experimental results show that arbitrarily large meshes are generated fast. The enumeration of rectangular regions with high number of hexagons is accurate despite its dimensions. An evaluation of the method under several circumstances is presented indicating its high performance for the generation of a high number of hexagons.**

***Keywords*-hexagonal meshes, parametric surfaces, Hamada indexes, nanostructure modeling**

I. Introduction

This work presents a method of rectangular hexagonal mesh generation based on an adaptation of Bresenham's algorithm [1]. Using a half-edge data structure, the resulting mesh is suitable for several operations and processing.

Hexagonal surfaces are frequently found in nature, such as the epidermal layer of cells [2] and the honeycombs [3]. In physics and chemistry, hexagonal structures are studied in the form of molecule arrangements. Modeling of chemistry compounds and nanostructures are interesting examples. Carbon nanotubes [4] are a special kind of nanostructure. The outstanding nanotube properties incite the scientists to study their carbon anisotropy. This is a very special case of hexagonal mesh found in nature. However, their high cost of production and manipulation make computer simulation models attractive to study their electromechanical features. This is a case where a fast and accurate hexagonal mesh generation is important.

Much work has been done about hexagonal meshes. Paper [5] presents an algorithm for computing a specified shape using this kind of mesh, computing first the regular triangulation and then turning it into a planar hexagonal mesh. Our method in comparison with this is much specific and proposes a direct hexagonal mesh generation to obtain faster results. Wang & Liu [6] uses an initial quad-mesh to compute a hexagonal mesh and then apply local perturbations to approximate a given surface. A method to spacial subdivision on hexagonal mesh was proposed in [7]. Examples of operations performed into pre-built hexagonal meshes can be found in [8].

Our main objective is to construct a rectangular hexagonal mesh using a small number of parameters. The main object is a strip of hexagons defined by two integers (n,m) which is constructed using a variation of the Bresenham's algorithm. Several strips can be stacked to form a rectangular region. Using half-edges, an explicit mesh is constructed efficiently and can be used to represent non-planar topologies. This

method is shown in Figure 2 and has the following features:

- three positive integers (n,m,l) are used to specify arbitrarily rotated rectangular regions on a hexagonal grid;
- a variation of the Bresenham's algorithm is used to generate hexagonal strips;
- a half-edge structure is used to allow fast mesh generation and non-planar topologies;
- the mesh can be used as a parameter space for obtaining different geometric models.

This work is motivated by the need of fast generation of carbon arrangements for physics and chemistry simulations. Most carbon nanostructures are based on hexagonal cells. Futhermore, bidimensional parametric objects are sufficient for obtaining models suitable for most simulations. Thus, this work concerns the specific problem of bidimensional rectangular hexagonal mesh generation for parametric object representation.

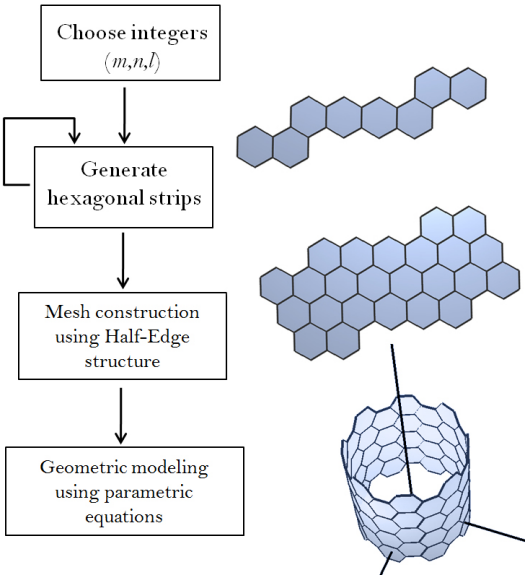


Figure 2. Firstly, three integers (n,m,l) are given. A sequence of strips is generated. The result is stored for further processing.

II. Hexagonal Cell Enumeration

Our method can enumerate hexagonal cells for an arbitrarily rotated rectangular region using three integer positive parameters (n,m,l). This section shows how this enumeration is achieved by a variation of the Bresenham's algorithm for plotting lines. Also, an enumeration method of hexagonal cells is derived. Our method is suitable to generate bidimensional regular hexagonal meshes in any orientation. It is based on the Bresenham's algorithm [1] for drawing straight lines and its main parameters were proposed by Hamada [9].

A. Hamada parameters

Hamada Parameters for defining bidimensional regular hexagonal meshes were initially used to describe carbon nanotubes. It was chosen because it completely describes the hexagonal mesh using two integer numbers (n,m), where $n \geq m$ in order to capture the hexagonal symmetry of the mesh [10].

A non-orthogonal basis called *chiral vectors* $\vec{c}_h$ is used to describe the mesh orientation. In the original paper, these vectors are used to define the direction of a carbon nanotube over a graphene sheet (Fig. 3a).

The chiral vector is defined as:

$$\vec{c}_h = n\vec{a}_1 + m\vec{a}_2 \equiv (n, m) \tag{1}$$

where $\vec{a}_1$ and $\vec{a}_2$ are the unit vectors of hexagonal mesh.

The information given by this vector will be used in the demonstration of the method in the Section II-B.

B. Hexagonal strip generation

A variation of the Bresenham's algorithm is derived for hexagonal strip generation.

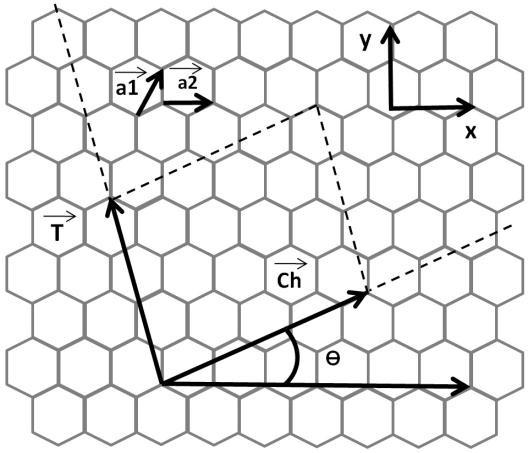


Figure 3. The elements on the hexagonal grid considered by the method.

The initial point *P*, shown in Figure 4, is the centroid of the first hexagon, next to the origin. The decision of which hexagon should be drawn is based on the distances between the line, given by the chiral vector, and the centroid *P*. The hexagon with smallest distance is chosen. First, we take the line equation:

$$y = bx + q \tag{2}$$

where we consider q as zero, assuming that the line passes through the origin and b is the slope of the line.

According to the Figure 4, we take $P_0(x_0,y_0)$ as an initial point and calculate P_1 and P_2 as functions of x_0 and y_0:

$$P_1 = (x_0 + \sqrt{3}a_0, y_0) \tag{3}$$

$$P_2 = (x_0 + \frac{\sqrt{3}}{2}a_0, y_0 + \frac{3}{2}a_0) \tag{4}$$

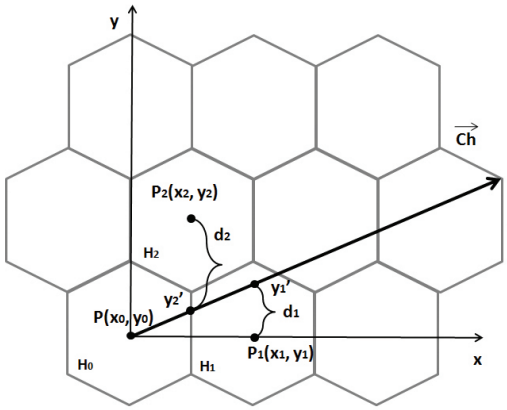


Figure 4. The method considers the distances d_1 and d_2 from the line to choose which hexagon is better to approximate the line.

where constant a_0 represents the side of the hexagon.

The objective of this demonstration is to represent the distance from the centroid to the line, as showed in Figure 4. We call these distances as d_1 and d_2 and represent them as:

$$d_1 = y_1^{'} - y_1 \tag{5}$$

$$d_2 = y_2 - y_2^{'} \tag{6}$$

We define $P_k(x_k,y_k)$ as a generic point of k-iteration:

$$y_1^{'} = b(x_k + \sqrt{3}a_0) \tag{7}$$

$$y_2^{'} = b(x_k + \frac{\sqrt{3}}{2}a_0) \tag{8}$$

Replacing the equations 7 and 8 on 5 and 6, as well as y_1 and y_2:

$$d_1 = b(x_k + \sqrt{3}a_0) - y_k \tag{9}$$

$$d_2 = (y_k + \frac{3}{2}a_0) - b(x_k + \frac{\sqrt{3}}{2}a_0) \tag{10}$$

Analyzing the values of d_1 and d_2, we observe that:

$$\begin{cases} \text{if } \ d_1 \geq d_2, \ \text{choose } H_2 \\ \text{if } \ d_2 > d_1, \ \text{choose } H_1. \end{cases} \tag{11}$$

Instead of using the inequality 11, we can analyze the signal of difference $d_1 - d_2$. This difference defines a parameter p_k and replace $b = \frac{\Delta y}{\Delta x}$, where Δy and Δx are the variations of x and y, respectively:

$$\begin{aligned} p_k &= \Delta x(d_1 - d_2) \\ &= 2x_k\Delta y + \Delta y\sqrt{3}a_0 + \Delta y\tfrac{\sqrt{3}}{2}a_0 - 2y_k\Delta x - \tfrac{3}{2}a_0\Delta x \\ &= 2x_k\Delta y - 2y_k\Delta x + (3\Delta y\tfrac{\sqrt{3}}{2}a_0 - \tfrac{3}{2}a_0\Delta x) \end{aligned} \tag{12}$$

where the last two terms are not in function of x_0 and y_0. We can consider these terms as constants and group them in a constant c:

$$p_k = 2x_k\Delta y - 2y_k\Delta x + c \tag{13}$$

This constant c is used to calculate the initial value of p_k, considering that P_0 is at the origin. The values that change between the iterations are represented by x_k and y_k, as showed on Equation 13. Based on this information, on the iteration $k+1$, the parameter p_{k+1} is given by:

$$p_{k+1} = 2x_{k+1}\Delta y - 2y_{k+1}\Delta x + c \tag{14}$$

Applying the difference between p_{k+1} and p_k:

$$p_{k+1} = p_k + 2\Delta y(x_{k+1} - x_k) - 2\Delta x(y_{k+1} - y_k) \tag{15}$$

The Equation 15 shows the iteration $k+1$ as a function of the previous iteration plus an increment.

Assuming that hexagon H_1 is chosen, as shown in Figure 5, we consider these differences:

$$x_{k+1} - x_k = \sqrt{3}a_0 \tag{16}$$

$$y_{k+1} - y_k = 0 \tag{17}$$

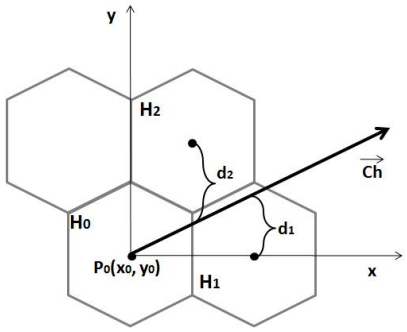


Figure 5. The configuration of the chiral vector $\vec{c}_h$ when the hexagon H_1 is choosen.

Replacing these results on equation 15:

$$p_{k+1} = p_k + 2\Delta y\sqrt{3}a_0 \tag{18}$$

The Equation 18 represents the increment of p_{k+1} when hexagon H_1 is chosen. The demonstration is similar for hexagon H_2.

Liu [8] proposed a method for drawing straight lines on hexagonal grids using only integer operations. The approach described in this section is a particular case which can be used to generate rectangular regions. Similarly, Lijun [11] presents an algorithm to generate line on hexagonal prism grid. First he describes the method on a 2D hexagonal grid and then extends it to a 3D space. He also uses integer operations.

C. Rectangle construction

As stated above, a strip over a hexagonal grid can be defined by the parameters n and m. As shown in Figure 6, the strips can be combined to form a rectangular region as a stack of $l+1$ strips. Thus, a rectangular region on a hexagonal grid can be defined with a triple of positive

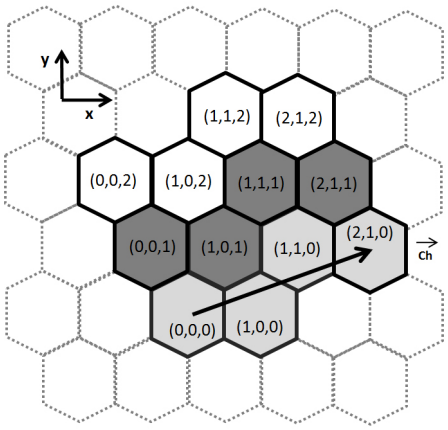


Figure 6. Enumeration of a rectangle defined by the triple of integers $(4, 2, 2)$. This means that each strip is defined by $(4, 2)$ coordinates and the rectangle has 3 strips of this type.

integers (n,m,l). This rectangular mesh can be used to represent other topologies as presented on Section III.

An important characteristic is that all strips are equal, as the method is independent of position in the plane. This fact allows appending new strips at any time.

III. Mesh construction using half-edges

The Section II presented a method for hexagon enumeration. The result is a sequence that needs to be linked and stored in order to be manipulated. In order to perform physics simulations, the widely used half-edge structure is an interesting choice since it allows easier and faster operations on the mesh. Half-edges provide an efficient and flexible local control of vertices, faces and edges. Fast access of the geometric attributes of the mesh is important for an efficient simulation. Futhermore, the geometry of the objects can abruptly change due to mesh ruptures during electromechanical simulations. Thus, implicitly connected hexagonal meshes are not suitable for this kind of application. This section presents a method for hexagonal grid generation based on the half-edge data structure.

The generated rectangle has some important properties for the construction of nanostructures. A special feature is that the rectangle can be rolled for any triple integers (n,m,l) with guaranteed match of the hexagons of the borders [3]. The Figure 7a highlight the corresponding vertices and Figure 7b presents the mesh after their connection. In this example, we use the parametric equation of the cylinder to construct the nanotube. The enumeration of hexagonal grid allows connecting the corresponding vertices in the parametrical space. Other objects with cylinder topology can be generated using the same process.

For some indexes (n,m,l) it is possible to obtain a torus topology connecting the first hexagonal strip of the mesh with the last one (Fig. 9b). This is the case when $m = 0$, $n > m$ and l is odd or when $n = m$, for instance.

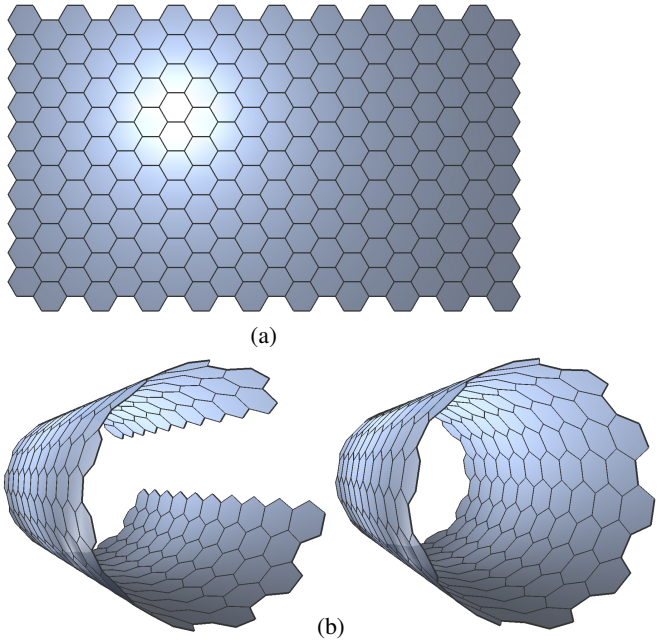


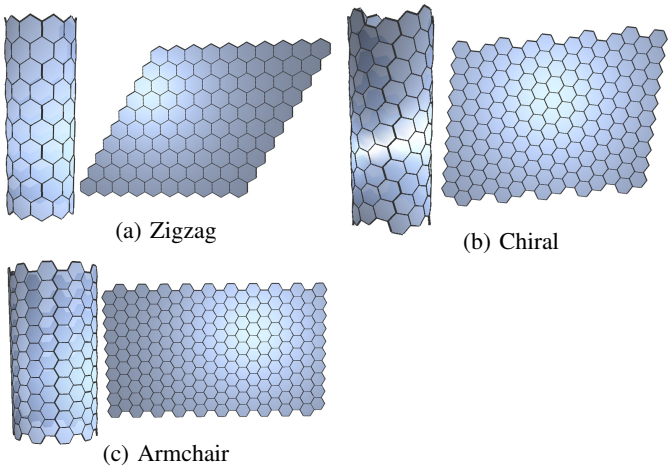
Figure 7. Illustration of a hexagonal connected in the $\vec{c}_h$ direction (a) rectangle (b) cylinder before and after the border connection.

IV. Applications

One of the applications of this work is to model nanostructures such as carbon nanotubes (single and multiwalled), carbon nanotube forests and other structures. The generation of meshes using the integers (n,m,l) is sufficient to represent the three classes of single-walled nanotubes: zigzag, armchair and chiral [10]. This classification is important since electromechanical properties depend on the topology of these structures. The structural difference is more evident on the border of nanotube, as shown on Figure 8. The generation of these structures is fast even if using high values of n, m or l. Typical values of the parameters are $n < 500$, $m < 300$ and l can be arbitrarily large.

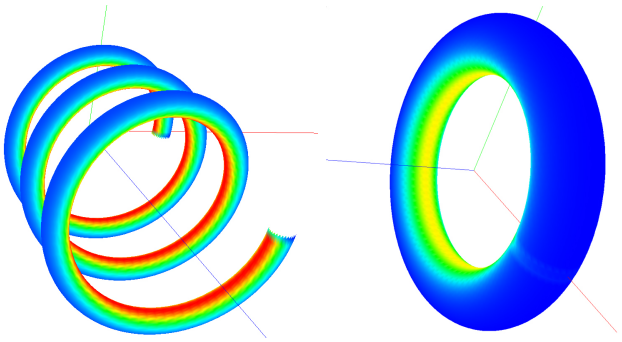
The visualization of physics systems is an important aspect for the analysis of the studied structures and several phenomena. As an example, we use the Lennard-Jones potential [12], which is an all-to-all process on vertices, to extract on site energy of the system. For viewing the result, a thermal color palette (inner regions have lower energy) was used to represent the energy variation, as shown on Figure 9. The half-edge structure makes easy the vertex scanning.

In fact, the diameter of carbon tubes has nanometric dimension but its length may have several micrometers [13]. Also, forests containing several thousands of nanotubes are interesting for simulations. Thus, it is important to provide a method capable of generating thousands of nanotubes, each one with distinct attributes: chirality, diameter, long length, single or multiwalled (Fig. 10a). As showed in next section,



(a) Zigzag (b) Chiral (c) Armchair

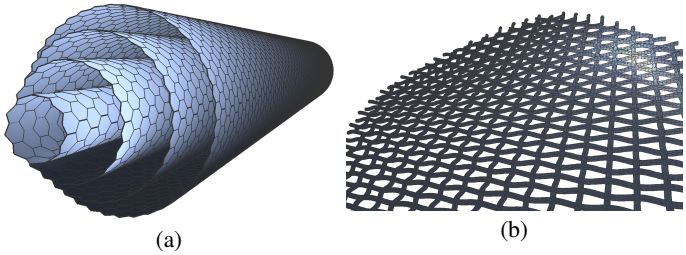
Figure 8. Types of nanotubes and their respective graphene sheet representation using a hexagonal grid.



(a) (b)

Figure 9. Hexagonal meshes representing nanostructures with Lennard-Jones potential computed. (a) spiral (b) torus.

our method is suitable for fast generation of structures with several nanotubes.



(a) (b)

Figure 10. Complex structures typically used in physics simulations: (a) Multiwalled nanotube (b) nanotubes interlaced forming a tissue.

V. Experimental results

In this section, we present experimental results obtained with a Dual Xeon E5410 Quad Core PC running at 2.33 GHz, with 4 GB RAM and 12 MB L2 cache under Windows Vista Ultimate 64 bits operating system. Although it is a multiprocessor machine, our implementation is currently sequential and so does not take advantage of it. The objective of our experiments is to show that the time to generate a nanostructure is a linear function of the number of hexagons that compose it. So, the estimated computation time should take the form

Table I
Configurations evaluated

Type of nanotube	n	m	l
Zigzag	10 to 40	0	5,000 to 25,000
Chiral	11 to 41	10 to 40	5,000 to 25,000
Armchair	10 to 40	n	5,000 to 25,000

Table II
Time to generate the nanotubes

Type of nanotube	n,m,l	Hexagons	Time (ms)
Zigzag	40;0;15,000	600,000	1,910.5
Chiral	31;30;10,000	610,000	1,931.7
Armchair	30;30;10,000	600,000	1,908.1

$$computation\_time = k * [(n + m) * l] \qquad (19)$$

where k is the time to generate one hexagon and $[(n+m)*l]$ is the total of hexagons.

To evaluate the performance of the proposed method, we examined its computational implementation under several distinct configurations defined by the different values of n, m and l, as illustrated by table I. We evaluated 20 different configurations of Zigzag, 15 of Armchair, with n ranging from 10 to 40 in steps of 10. A total of 38 different configurations were evaluated for Chiral, with n ranging from 11 to 41 and m ranging from 10 to 40, both m and n in steps of 10. In all three cases, l ranges from 5,000 to 25,000 in steps of 5,000.

We executed each configuration ten times, and reported the average execution time for some of them in table II and in Figure 11. The standard deviation was less than 10% for all configurations.

As can be observed in table II, the implemented algorithm is very fast. In fact, for the configuration with the greater number of hexagons (n=31, m=10 and l=25,000, a total of 1,025,000 hexagons), it took less than 3.4 s to generate the nanostructure.

A careful analysis of our figures showed us that the time to generate a single hexagon is, in average, equal to 4.2 μs on the machine used for the evaluation. The figures also lead us to conclude that the execution time is linear with respect to the number of hexagons, as Figure 11 demonstrates.

VI. Conclusions

A method for bidimensional rectangular hexagonal mesh generation was proposed. Experimental results has shown that our method is suitable for fast and accurate generation of meshes, even with huge values of (n,m,l). In fact, it has been

Figure 11. Mean execution times for three different configurations of armchair, chiral and zigzag with approximately the same number of hexagons.

shown that the proposed method is linear with the number of hexagons to draw, as shown on Section V. Our method allows us to generate several nanostructures and monitor their behavior on simulation engines. Fast generation of hexagonal structure is important to allow the modeling, in real-time, of complex and intricate objects like tissues containing thousands of single or multiwalled structures (Fig. 10b).

A perspective of this work is to use the Lennard-Jones potential to perform deformations on the mesh and then obtain a structure more stable. Stability here means that the geometry is subject to some constraints that must be obeyed. As the example of torus or the helix, the application of the potential will help to respect some physic features like the distance between atoms. A prior analysis, as shown on Figure 9, highlights some regions of larger energy that do not satisfy this condition.

We also have the goal to generate structures that allows the construction of more complex models using nanojunctions [14].

ACKNOWLEDGMENT

Authors thank to FAPEMIG-*Fundação de Amparo à Pesquisa do Estado de Minas Gerais*, CAPES-*Coordenação de Aperfeiçoamento de Pessoal de Ensino Superior*, CNPQ-*Conselho Nacional de Desenvolvimento Científico e Tecnológico* and UFJF for funding this research.

REFERENCES

[1] J. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[2] D. W. Thompson, *On Growth and Form*. Cambridge, University Press, 1945.

[3] Brimkov, E. Valentin, Barneva, and P. Reneta, "Analytical honeycomb geometry for raster and volume graphics," *Computer Journal*, vol. 48, no. 2, pp. 180–199, March 2005. [Online]. Available: http://dx.doi.org/10.1093/comjnl/bxh075

[4] S. Iijima, "Helical microtubules of graphitic carbon," *Nature*, vol. 354, pp. 56–58, November 1991. [Online]. Available: http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1991Natur.354...56I

[5] W. Wang, Y. Liu, D.-M. Yan, B. Chan, R. Ling, and F. Sun, "Hexagonal meshes with planar faces," Dept. of CS, HKU, Tech. Rep., 2008.

[6] W. Wang and Y. Liu, "A note on planar hexagonal meshes," *I.Z. Emiris, F. Sottile and T. Theobald, editors IMA Volume 151: Nonlinear Computational Geometry , Springer-Verlag*, 2009.

[7] P. Oswald and P. Schröder, "Composite primal/dual $\sqrt{3}$-subdivision schemes," *Comput. Aided Geom. Des.*, vol. 20, no. 3, pp. 135–164, 2003.

[8] Y.-K. Liu, "The generation of straight lines on hexagonal grids," *Comput. Graph. Forum*, vol. 12, no. 1, pp. 27–32, mar 1993.

[9] N. Hamada, S.-i. Sawada, and A. Oshiyama, "New one-dimensional conductors: Graphitic microtubules," *Phys. Rev. Lett.*, vol. 68, no. 10, pp. 1579–1581, Mar 1992.

[10] O. V. Kharissova, J. Rangel-Cardenas, and M. G. Casta, "Structural computing of hamada parameters for carbon nanotubes processed by microwave heating," *Innovative Computing, Information and Control, International Conference on*, vol. 0, p. 357, 2007.

[11] L. He, Y. Liu, and S. Bao, "3d line generation algorithm on hexagonal prism grids," in *CISP '08: Proceedings of the 2008 Congress on Image and Signal Processing, Vol. 3*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 741–745.

[12] S. J. Stuart, A. B. Tutein, and J. A. Harrison, "A reactive potential for hydrocarbons with intermolecular interactions," *Journal of Chemical Physics*, 2002.

[13] H. Terrones, F. López-Urías, E. Muñoz-Sandoval, J. Rodríguez-Manzo, A. Zamudio, A. Elías, and M. Terrones, "Magnetism in fe-based and carbon nanostructures: Theory and applications," *Solid State Sciences*, vol. 8, no. 3-4, pp. 303 – 320, 2006, special issue in honour of A.K.Cheetham.

[14] I. Zsoldos, G. Kakuk, T. Réti, and A. Szasz, "Geometric construction of carbon nanotube junctions," *Modelling and Simulation in Materials Science and Engineering*, vol. 12, no. 6, pp. 1251–1266, 2004. [Online]. Available: http://stacks.iop.org/0965-0393/12/1251