



Development of computational algorithms for the generation of graphene based structures

João Rafael da Silva Coelho

Thesis to obtain the Master of Science Degree in

Materials Engineering

Supervisor: Prof. José Carlos Garcia Pereira

Examination Committee

Chairperson: Prof. Maria de Fátima Reis Vaz

Supervisor: Prof. José Carlos Garcia Pereira

Members of the Committee: Prof. Luís Filipe da Silva dos Santos

December 2016

Acknowledgements

To all those who helped and supported me throughout this work, especially:

Professor José Carlos, for being an amazing supervisor who has always supported me throughout the realization of this work and displayed a genuine interest in my success, and also for providing me this opportunity to work with him in a project meant to improve a free and open-source program, a project which involved programming and working under Linux, two of my favourite things. Thank you for providing me with the opportunity to work on my favourite project of all since i've entered IST, and making it an enjoyable experience.

To my mother, father and brother, for aiding me in the hardest times and always being present.

To Rona, for her incredible patience, time and support.

Abstract

GAMGI is a free program available for Linux operating systems, used for the construction, visualization and analysis of atomic structures.

The objective of this work was the research, development and implementation in GAMGI of algorithms that allow the creation of molecular structures based on graphene: nanotubes, nanocones and fullerenes.

Graphene is a carbon based structure which presents interesting mechanical, thermal and optical properties. Three important classes of atomic structures may be computationally generated from graphene: carbon nanotubes, carbon nanocones and fullerenes. These molecules, like graphene, present properties that make them interesting for applications in different scientific areas, such as medicine and engineering.

Algorithms were developed and implemented throughout the course of this work, which allow for the generation of any single-walled nanotube of the zig-zag, armchair and chiral types, with an arbitrary height.

Other algorithms were equally developed and implemented which allow for 16 kinds of open or closed nanocones to be built, with an arbitrary height, with disclination angles between 60° and 300° , differing in the positions or only locally, as a result of the applied construction method.

Finally, caps were built which, when correctly applied to specific classes of nanotubes – (5,0), (5,5) and (6,6) – allow for the generation of infinite series of fullerenes, which include the most experimentally observed molecules of this type, C_{60} and C_{70} , in both cases with the 12 pentagons (demanded for this kind of structure) all isolated. For fullerenes, other important topological algorithms were described and in some cases analyzed but not implemented, and may constitute a basis for future work.

Key-words: graphene, algorithms, nanotubes, nanocones, fullerenes

Resumo

O GAMGI é um programa livre disponível para sistemas operativos Linux, utilizado para a construção, visualização e análise de estruturas atómicas.

Este trabalho teve como objectivo a pesquisa, desenvolvimento e implementação no GAMGI de algoritmos que permitam a criação de estruturas moleculares baseadas em grafeno: nanotubos, nanocones e fulerenos.

O grafeno é uma estrutura de carbono que apresenta interessantes propriedades mecânicas, térmicas, eléctricas e ópticas. A partir do grafeno, podem ser geradas computacionalmente três classes importantes de estruturas atómicas: nanotubos de carbono, nanocones de carbono e fulerenos. Estas moléculas, tal como o grafeno, apresentam propriedades que as tornam interessantes para aplicações em diferentes áreas científicas, entre as quais a engenharia e a medicina.

Neste trabalho foram desenvolvidos e implementados algoritmos para construir qualquer nanotubo de parede simples, do tipo “zig-zag”, “armchair” ou “chiral”, com uma altura arbitrária.

Foram igualmente desenvolvidos e implementados algoritmos que permitem construir 16 tipos de nanocones, abertos ou fechados, com uma altura arbitrária, com ângulos de supressão (“disclination”) que variam entre os 60° e os 300°, diferindo na posição dos pentágonos resultantes ou apenas localmente, como resultado do método de construção aplicado.

Finalmente, foram construídas coberturas (“caps”) que, devidamente aplicadas em classes específicas de nanotubos – (5,0), (5,5) e (6,6) – permitem gerar séries infinitas de fulerenos, que incluem nomeadamente as moléculas deste tipo mais observadas experimentalmente, o C_{60} e o C_{70} , em ambos os casos com os 12 pentágonos (exigidos para este tipo de estrutura) todos isolados. Para os fulerenos, foram igualmente analisados e nalguns casos descritos outros algoritmos topológicos importantes, que não foram implementados, constituindo uma base de partida para o futuro.

Palavras-chave: grafeno, algoritmos, nanotubos, nanocones, fulerenos

Table of Contents

Acknowledgements	I
Abstract	II
Resumo	III
Table of Contents	IV
List of Figures	VI
Index of Tables	XI
List of Code Text Segments	XII
Abbreviations	XIV
1.Introduction.....	1
2.State of the Art.....	1
2.1.Graphene.....	1
2.2. Nanotubes.....	2
2.2.1.Definitions and terminology.....	2
2.2.2. Modelling.....	3
2.2.3. Discovery.....	4
2.2.4. Synthesis.....	5
2.2.5. Properties.....	5
2.2.6.Applications.....	6
2.3.Nanocones.....	6
2.3.1. Definition.....	6
2.3.2. Discovery.....	6
2.3.3. Modelling of SWCNCs.....	7
2.3.4. MWCNCs modelling.....	10
.....	11
2.3.5. Energetic considerations.....	11
2.3.6. Synthesis.....	14
2.3.7. Properties and applications.....	14
2.4.Fullerenes.....	15
2.4.1. Discovery.....	15
2.4.2. Euler’s Theorem.....	15
2.4.3. Isomers.....	16
2.4.4. Isolated Pentagon Rule.....	17
2.4.5. Pentagon and hexagon indices.....	17
2.4.6. Fullerene Graph Generation Algorithms.....	19
2.4.6.1. Graph Theory.....	19
2.4.6.2.Fullerene Duals.....	21
2.4.6.3. Ring-Spiral Algorithm.....	23
2.4.6.4. Face-adding algorithms.....	24
2.4.6.5. Leapfrog Transformation.....	26
2.4.6.6. Fullerene Generation Software.....	27
2.4.7. Synthesis.....	27
2.4.8. Applications.....	27
3.Algorithms and Results.....	27
3.1. Nanotubes.....	28
3.1.1. Graphene Base Vectors and Atomic Positions.....	28
3.1.2. Nanotube Base Vectors, Translation Vector and Height.....	29
3.1.3. Lattice Rotation.....	30
3.1.4.Scanning Region.....	31

3.1.5. Coordinate Transformation.....	32
3.1.6. Periodic Boundary Conditions.....	33
3.1.7. Bond Creation.....	34
3.1.8. Results.....	36
3.2. Nanocones.....	37
3.2.1. General Considerations.....	37
3.2.2. Single Disclination Cones.....	38
3.2.2.1. Atom Selection.....	38
3.2.2.2. Scanning region.....	39
3.2.2.3. Coordinate Transformation.....	40
3.2.3. Multiple Disclinations.....	41
3.2.3.1. Origin Translation and Radii.....	42
3.2.3.2. Origin Conversion and Angle.....	42
3.2.3.3. Atom Removal from Disclination Gaps.....	43
3.2.3.4. Rotation Check.....	45
3.2.3.5. Coordinate Transformation.....	48
3.2.3.6. Analysis of Nanocone 120_E.....	49
3.2.3.7. Bond Creation.....	50
3.2.3.8. Analysis of Nanocone 60_O.....	51
3.2.4. Results.....	52
3.3. Fullerenes.....	55
3.3.1.1. L_0 Transformation.....	55
3.3.1.2. $C_{28} \rightarrow C_{24}$ Transformation.....	59
3.3.2. Nanotube Fullerenes.....	61
3.3.2.1. (5-0)-type Nanotube Fullerenes.....	62
3.3.3. Results.....	66
4. Graphical interface.....	68
5. Conclusions.....	72
6. References.....	75
Appendix A - Creation schemes for the non-origin nanocone isomers	81
Appendix B - Table of nanocone bond indices	87

List of Figures

Figure 1 - The graphene honeycomb lattice.....	2
Figure 2 - Electron microscopy image of a single-wall carbon nanotube suspended between two CNT bundles [10].....	3
Figure 3 - Selection rectangle for a (5,5) carbon nanotube. T is the translation vector, b1 and b2 the nanotube base vectors, and h is the height.....	4
Figure 4 - Schematic representation of methods used for carbon nanotube synthesis (a) Arc discharge (b) Chemical vapor deposition (c) Laser ablation (d) hydrocarbon flames [23].....	5
Figure 5 - TEM of the five types of cones in Krishnan's sample, a) apex angle = 19.2°; b) apex angle = 38.9°; c) apex angle = 60°; d. apex angle = 86.6°; e) apex angle = 123.6°; f) High resolution of image of a cone tip. [33].....	7
Figure 6 - The five possible nanocone disclination angles.....	8
Figure 7 - Formation of a nanocone from a graphene sheet with a 60° disclination. This results in a cone with a single pentagon, located on the cone apex, with an apex angle of 112.9°. Thick lines in the image on the right show the location of the pentagon. There is only one possible nanocone structure with a single pentagon [39].....	8
Figure 8 - Optimized structures of five CNCs with one, two, three, four, and five pentagons, respectively. The pentagons colored in light gray are displayed. (a) The top view. (b) The side view. [35].....	9
Figure 9 - Formation of nanocones from a graphene sheet with a total of 240° disclination. There are four pentagons in the resulting structure, with an apex angle of 38.9°. Thick lines in the images on the right show the location of the pentagons. There are many possible structures corresponding to different locations of the pentagons, with two shown here: (a) structure with the pentagons arranged in a rectangular fashion with twofold symmetry,(b) structure with pentagons arranged in a diamond pattern with twofold symmetry. [39].....	9
Figure 10 - Schematic illustrations of CNCs with different opening angles (a). TD = 60°, apex angle = 19.2°; (b). TD = 120°, apex angle = 38.9°; (c). TD = 180°, apex angle = 60°; (d). TD = 240°, apex angle = 86.6°; (e). TD = 300°, apex angle = 123.6°. [33].....	10
Figure 11-Schematic of cone formation by the cone-helix model [36].....	11
Figure 12 - Remaining Nanocone Isomers used for the simulations carried out in SPARTAN [39].....	12
Figure 13 - FESEM images of a cone-covered graphite aggregate. (a) Low-magnification image showing complete coverage of the aggregate surface with conical structures. A ~39° cone is marked by an arrow. (b) Higher magnification	

image of the sample showing a variety of large cones with different apex angles and sharp and blunt tips. Arrows show changes in the apex angle. (c) Close up view of two surfaces which are almost perpendicular and show different cone morphologies —large cones on one surface and globular (artichoke-like) structures on the other. The latter are clusters of large-angle cones. Arrows show some of the cones that are ripped on the side. [36].....13

Figure 14 - Typical cone morphologies. (a) SEM image of a cone with a 60o apex angle, the most common apex angle. The slightly uneven surface of the cone suggests layer growth. (b) FESEM and (c) SEM images of large cones with numerous smaller cones growing on their surface. Smaller cones covering surfaces of large cones have a broad distribution of shapes, but large apex angles prevail (c). (d) FESEM image of four cones having sharp and broad tips (multiple tips are marked by arrows). The cones are oriented to reveal their circular cross sections around the tips and layered growth (ripples). [36].....14

Figure 15 - Number of distinct (non-isomorphic) fullerene isomers CN (with and without fulfilling the IPR) with increasing number of carbon atoms N up to N = 400 (double logarithmic scale). [54].....16

Figure 16 - A selection of different 3D shapes for regular fullerenes (distribution of the pentagons DP are set in parentheses). ‘Spherically’ shaped (icosahedral), for example, (a) C20 - Ih , (b) C60 - Ih , and (c) C960 - Ih (DP = 12 × 1); barrel shaped, for example, (d) C140 - D3h (DP = 6 × 2); trigonal pyramidally shaped (tetrahedral structures), for example, (e) C1140 - Td (DP = 4 × 3); (f) trihedrally shaped C440 - D3 (DP= 3 × 4); (g) nanocone ormenhir C524 - C1 (DP = 5 + 7 × 1); cylindrically shaped (nanotubes), for example, (h) C369 - D5h , (i) C1152 - D6d , (j) C840 - D5d (DP = 2 × 6). The fullerenes shown in this figure and throughout the paper have been generated automatically using the Fullerene program. [54].....17

Figure 17 - A 6 vertice graph [59].....19

Figure 18 - Graph of C20 [60].....21

Figure 19 - Two dual pairs are (a) the icosahedron and the dodecahedron, (b) the octahedron and the cube. The tetrahedron (c) is the only self-dual Platonic solid. [52]22

Figure 20 - Planar embeddings of fullerene graph and dual (blue color and dotted lines for the dual representation), and 3D embeddings of the duals: (a) C20 - Ih , for which the dual is the icosahedron; (b) C60 - Ih , for which the dual is the pentakis-dodecahedron. [55].....23

Figure 21 - The unique spiral of the dodecahedron, shown superimposed on the corresponding planar graph. Each new face in the spiral after the second is adjacent to both (a) its immediate predecessor and (b) the first face in the preceding spiral that still has an open edge [52].....23

Figure 22 - Two of the basic fullerene isomers [63].....24

Figure 23 - Expansions L0, L3, B0,0, B3,2 and F [63].....25

Figure 24 - The L and B expansions in dual representation [64].....	25
Figure 25 - Leapfrogging a dodecahedron: a dodecahedron is converted to a 32-vertex deltahedron by omnicaapping, then to a truncated icosahedron by taking the dual [52].....	26
Figure 26 - Graphene lattice and selection rectangle of figure 3 after applying 30o clockwise rotation.....	30
Figure 27 - The semi-circle area and lozenge of graphene cells for a nanotube. r is the rectangle diagonal.....	31
Figure 28 - Selection rectangle with length T. A is an atom with coordinates (x,y).....	32
Figure 29 - Coordinate transformation for atom A of figure 29.....	33
Figure 30 - Bonding scheme. Atoms 1 and 2 (in red) are A atoms. Atom 1 in cell (1,2) creates local bonds with atoms B (in blue) in cells (1,2) and (0,1), and a bond with atom B of cell (5,2), close to the right border. Atom B in cell (3,3) creates all possible local bonds, but no bonds with atoms close to the border.....	35
Figure 31 - Zig-zag (10,0) nanotube.....	36
Figure 32 - Armchair (10,10) nanotube.....	36
Figure 33 - Chiral (10,5) nanotube.....	37
Figure 34 - The selection area for a nanocone is the circular crown with maximum and minimum radii r_{out} and r_{in} , minus the 60° disclination gap.....	39
Figure 35 - The circle area and lozenge of graphene cells for a nanocone. r is the rectangle diagonal.....	39
Figure 36 - Cone height, h, base radius, r_{cone} , and distance from apex to outer point of base, r_{out}	41
Figure 37 - Bisector, a_1 and a_1 for a generic disclination. A is an atom at a distance l_c from the disclination center, making an angle α_c with the horizontal direction.....	43
Figure 38 - t_0 and t_1 for atom A. Note that t_1 is actually negative.....	44
Figure 39 - New disclination borders for the cases where atoms at a_0 are kept (red dashed lines) and atoms at a_1 are kept (blue dashed lines).....	45
Figure 40 - Disclination closed by rotating both edges. Atoms between $-\max$ and a_0 and atoms between \max and a_1 are rotated.....	46
Figure 41 - Disclination closed by rotating one edge. Atoms at a_0 remain fixed. Atoms between $a_0 + 90^\circ$ and a_1 are rotated.....	46
Figure 42 - Disclination closed by rotating one edge. Atoms at a_1 remain fixed. Atoms between $a_1 - 90^\circ$ and a_1 are rotated.....	47

Figure 43 - Schematic for the creation of isomer 120_E. The origin of the cone is marked by a black dot. Two 60o disclinations are applied, and both of them are closed by rotating both sides.....	49
Figure 44 - Bonding scheme for the 60_o nanocone isomer.....	51
Figure 45 - Nanocone 60_O.....	52
Figure 46 - Nanocone 120_O.....	53
Figure 47 - Nanocone 180_O.....	53
Figure 48 - Nanocone 240_O.....	53
Figure 49 - Nanocone 300_O.....	54
Figure 50 - Nanocones 120_E, 120_H and 180_D.....	54
Figure 51 - Nanocones 240_E, 240_H and 300_E.....	54
Figure 52 - Open nanocone 180_O.....	55
Figure 53 - Atoms that participate in transformation L0.....	56
Figure 54 - Atoms from figure 52 after applying transformation L0. The added atoms are numbered in blue.....	58
Figure 55 - Graphs of C24 (left) and C28 (right), obtained by adding the central atom and three atoms in alternating pentagons.....	59
Figure 56 - From left to right, top down view of caps for (5,0)-, (5,5)- and (6,6)-type nanotubes.....	62
Figure 57 - Pentagon circumscribed in circumference. θ is the inner angle of the triangles, which is known to be $2\pi / 5$, b is the bond length.....	62
Figure 58 - The height at which the cap should be placed, z_{top} , should be such that the bonds between the cap and the tube have the expected bond length.....	64
Figure 59 - Bonding scheme for the (5,5) cap. The black dots represent the cap atoms.....	65
Figure 60 - C20 and (5,0)-type nanotube C50.....	66
Figure 61 - C60 and (5,5)-type nanotube C90.....	67
Figure 62 - C72 and (6,6)-type nanotube C132.....	67
Figure 63 - C24 obtained from applying transformation L0.....	67
Figure 64 - Result of transformation C28 to C24.....	68

Figure 65 - The GAMGI graphical interface. The white arrow points to the "Molecule" option.....	69
Figure 66 - Enlarged image of the "Molecule" option.....	69
Figure 67 - Toolbox for the previous "Molecule" -> "Create" option.....	69
Figure 68 - Toolbox for the new "Molecule" -> "Create" option.....	70
Figure 69 - Input options for a graphene sheet.....	70
Figure 70 - Input options for nanotubes.....	71
Figure 71 - Input options for nanocones.....	71
Figure 72 - Input options for fullerenes.....	72
Figure 73 - Input options for caps.....	72
Figure 74 - Atom options.....	73

Index of Tables

Table 1 - Results of the SPARTAN simulation [39]. In this work, the nanocones in question are those from figures 9a, 9b, 12b, 12c and 12d, instead of 2a, 2b, 4b, 4c and 4d, respectively.....12

List of Code Segments

Code Segment 1 - Definition of the graphene base vectors and creation of graphene atoms.....	28
Code Segment 2 - Definition of nanotube base vectors and translation vector.....	29
Code Segment 3 - Definition of height based on number of rings.....	29
Code Segment 4 - Rotation of the graphene base vectors.....	31
Code Segment 5 - Definition of half-circle radius and maximum number of cells.....	32
Code Segment 6 - Nanotube radius and angle definitions and coordinate transformation.....	33
Code Segment 7 - Creation of graphene atoms with discarding of atoms with $x = T$.	34
Code 8 - Nanotube bond creation.....	35
Code 9 - Nanocone radii definition and atom discarding.....	38
Code Segment 10 - Cone radius and height definitions and coordinate transformation for single disclination.....	41
Code Segment 11 - Nanocone origin translation and atom discarding.....	42
Code Segment 12 - Origin conversion and angle calculation.....	42
Code Segment 13 - Definition of a_0 , a_1 , t_0 and t_1 and atom discarding from angular gap in the case of multiple disclinations.....	45
Code Segment 14 - Multiple disclination rotation process. First, check what edges are to be rotated and adjust the values of bisector and min accordingly. Second, convert angle to local values relative to bisector, with bisector = 0_0 . Third, return atom coordinates if they lie outside the rotation zone. Fourth, get new angle for atoms. Finally, convert angle back to absolute values.....	48
Code Segment 15 - Multiple disclination x and y coordinate conversion for atoms that are to be rotated.....	48
Code Segment 16 - Multiple disclination x, y and z coordinate conversion for atoms which are not to be rotated.....	49
Code Segment 17 - Nanocone bond creation.....	51
Code Segment 18 - L0 transformation atom selection.....	57
Code Segment 19 - L0 transformation bond creation.....	59
Code Segment 20 - C24 → C28 transformation.....	61

Code Segment 21 - Pentagon atom creation for (5,0)-type nanotube fullerenes.....	63
Code Segment 22 - Pentagon height calculation and cap creation for (5,0)-type nanotube fullerenes.....	65
Code Segment 23 - (5,0)-type nanotube fullerenes bond creation.....	66

Abbreviations

CNT – carbon nanotube

SWCNT – single walled carbon nanotube

MWCNT – multi walled carbon nanotube

CNC – carbon nanocone

SWCNC – single walled carbon nanocone

MWCNC – multi walled carbon nanocone

IPR – isolated-pentagon rule

1. Introduction

This thesis was elaborated with the objective of creating additional functionalities to the free and open-source software program GAMGI (General Atomistic Modelling Interface) [1]. The goal of this program, as stated on the official website, is "... to provide a free package to construct, view and analyse atomic structures, as powerful and simple to use as possible".

At the beginning of this work, GAMGI lacked the functionality of directly creating any kinds of molecules from simple input parameters given by the user. Over the course of this work, code has been written and added to the program, which allows for the creation of several different graphene based structures, namely carbon nanotubes, carbon nanocones and fullerenes. Algorithms that create the molecules based on user input were researched and then implemented in the C programming language, after which they were subjected to continuous development. The graphical interface also suffered various modifications to adjust to the new functionalities, so as to allow for direct user input.

2. State of the Art

2.1. Graphene

Graphite is a crystalline allotrope of carbon, and the most stable under standard temperature and pressure conditions. It has a layered, planar structure, and belongs to space group 194 (as the well known hexagonal compact structure), with Wyckoff positions b $(0, 0, \frac{1}{4})$, $(0, 0, \frac{3}{4})$ and c $(\frac{1}{3}, \frac{2}{3}, \frac{1}{4})$, $(\frac{2}{3}, \frac{1}{3}, \frac{3}{4})$ [2].

A layer of graphite is called graphene, and its crystalline structure can this be described as a 2D hexagonal lattice (with $a = 2.464 \text{ \AA}$, the graphite experimental value), with c atoms at positions $(0, 0)$ plus $(\frac{1}{3}, \frac{2}{3})$ or $(0,0)$ plus $(\frac{2}{3}, \frac{1}{3})$, forming a honeycomb pattern (Figure 1). Each atom is bonded to three other atoms on the same plane, the distances between them being about 1.42 \AA [3].

Graphene has various outstanding properties, such as being much stronger than the strongest steels, conducting heat and electricity efficiently and being nearly transparent [4], [5], [6]. In 2010, the Nobel Prize in Physics was awarded jointly to Andre Geim and Konstantin Novoselov "for groundbreaking experiments regarding two-dimensional material graphene".

Graphene holds great promise in terms of its applications, the most attractive being within the area of material/device applications, such as solar cells, LEDs, touch panels and smart windows or phones [7].

Graphene is a crystalline allotrope of other important carbon structures with 2-dimensional properties, such as carbon nanotubes, fullerenes and nanocones.

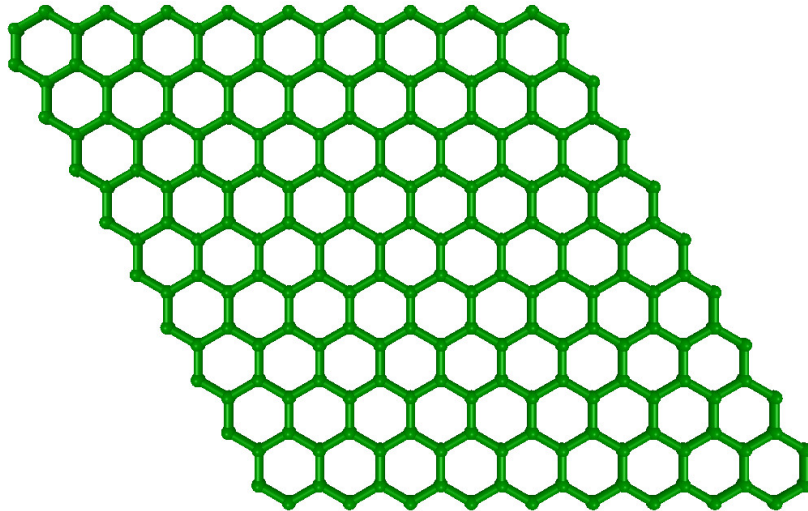


Figure 1 - The graphene honeycomb lattice

2.2. Nanotubes

2.2.1. Definitions and terminology

Carbon nanotubes are allotropes of carbon with a cylindrical shape (Figure 2). They can be thought of as rolled up sheets of graphene. There are several ways in which the sheet can be rolled to form a nanotube, and depending on that, the nanotube will exhibit different properties, such as whether it is metallic or semiconductor [8].

Nanotubes can be single-walled (SWCNT) or multi-walled (MWCNT), where a MWCNT consists of multiple rolled layers of graphene, that is, multiple stacked SWCNTs. Most SWCNTs have a diameter close to 1 nanometer, while being millions of times longer. They have been constructed with length-to-diameter ratio of up to 132,000,000:1 [9].

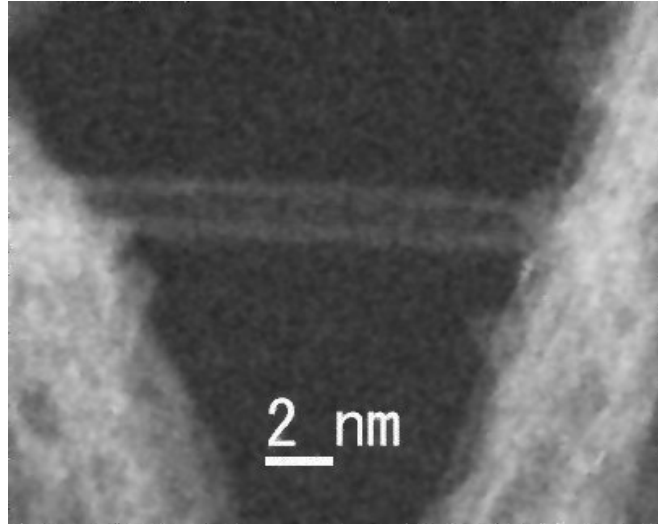


Figure 2 - Electron microscopy image of a single-wall carbon nanotube suspended between two CNT bundles [10]

2.2.2. Modelling

As shown in Figure 3, a carbon nanotube structure can be described through a translation vector $\mathbf{T} = m \cdot \mathbf{a}_1 + n \cdot \mathbf{a}_2$ linking two equivalent atoms of the graphene honeycomb lattice, where \mathbf{a}_1 and \mathbf{a}_2 are two base vectors separated by 60° , and m and n are integers that will identify the nanotube. As \mathbf{T} starts and ends at equivalent points, lines perpendicular to \mathbf{T} , passing through those two terminal points, must also be equivalent. Defining an arbitrary height for these perpendicular lines, a rectangle can be thus constructed where the terminal perpendicular sides are equivalent so the rectangle can be folded and the two perpendicular sides connected in a perfect way, forming a nanotube. The coordinates (m, n) defining the translation vector \mathbf{T} (and the geometrical height of the perpendicular sides) uniquely characterise the molecular structure of a carbon nanotube. Nanotubes are usually classified as zigzag, armchair or chiral, for coordinates of the type $(m, 0)$, (m, m) or (m, n) , respectively. These names have been given because zig-zag nanotubes have a typical zig-zag atomic structure at the termination top and bottom sections, while the armchair nanotubes have an atomic structure that closely resembles a typical sofa design, with a low flat sitting zone by higher flat armchairs. Chiral nanotubes are named this way because they lack the symmetry of the other, producing typical spirals when seen along the tube direction, as the 2D graphene symmetry directions are not aligned anymore with the longitudinal tube direction. [11], [12], [13], [14].

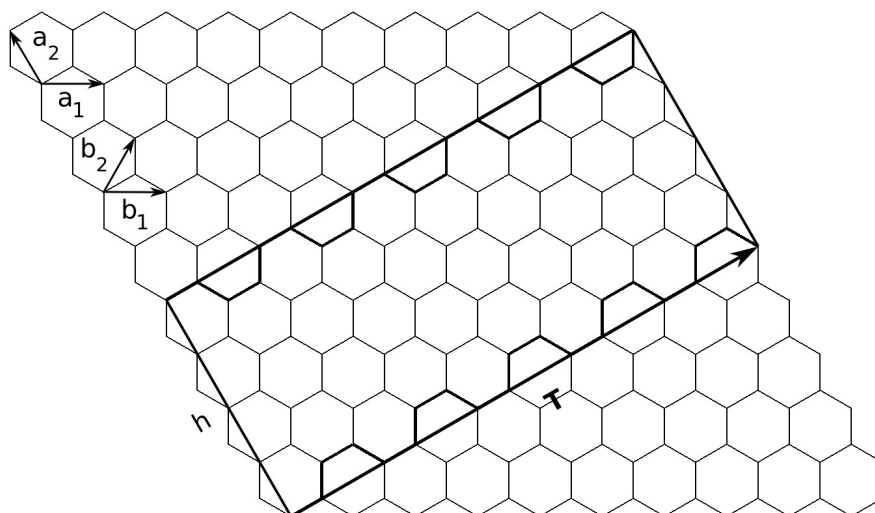


Figure 3 - Selection rectangle for a (5,5) carbon nanotube. \mathbf{T} is the translation vector, b_1 and b_2 the nanotube base vectors, and h is the height

2.2.3. Discovery

Carbon nanotubes (CNTs) have attracted lots of attention from scientists in various fields because they exhibit extraordinary physical and chemical properties due to their intrinsic nano-sized and carbon-based natures.

Nanotubes were first discovered in 1952, by L.V Radushkevich and V.M Lukyanovich, who published the first known transmission electron microscope images of multi-walled carbon nanotubes (MWNT's). However, it happens that not only the article was written in russian and published in a Russian Journal, but this discovery took place during the Cold War. Because of this, their findings were not given due attention [15]. In 1976, Endo, Koyama and Oberlin managed to synthesize and observe hollow tubes of roled up graphite sheets synthesized by a chemical vapour growth technique, and in this case, they were not multi but single-walled (SWNT's) [15], [16]. Endo also managed to observe SWNT's later, in 1988 [17]. In 1979, John Abrahamson et al. presented evidence of carbon nanotubes, where it was suggested that they could be grown in a nitrogen atmosphere at low pressures [18].

Despite these reports on the observation of nanotubes, it was not until 1991, six years after the discovery of fullerenes by Kroto [19], when Iijima [20] discovered this molecule in the insoluble material of arc-burned graphite rods, that carbon nanotubes started to draw great attention from the scientific community. Indeed, coupled with the prediction of Mintmire et al. [21], that if single-carbon nanotubes could be made, they would exhibit remarkable conducting properties, Iijima's discovery sparked the generalized interest in carbon nanotubes that we can observe up to this day.

2.2.4. Synthesis

Most of the processes used to produce CNTs take place in vacuum or with specific process gases. They include, but are not limited to: arc discharge, laser ablation, high-pressure carbon monoxide disproportionation and chemical vapour deposition (CVD), some of which can be seen in Figure 4. These methods allow for large quantities of nanotubes to be fabricated, making CNTs more commercially viable [22].

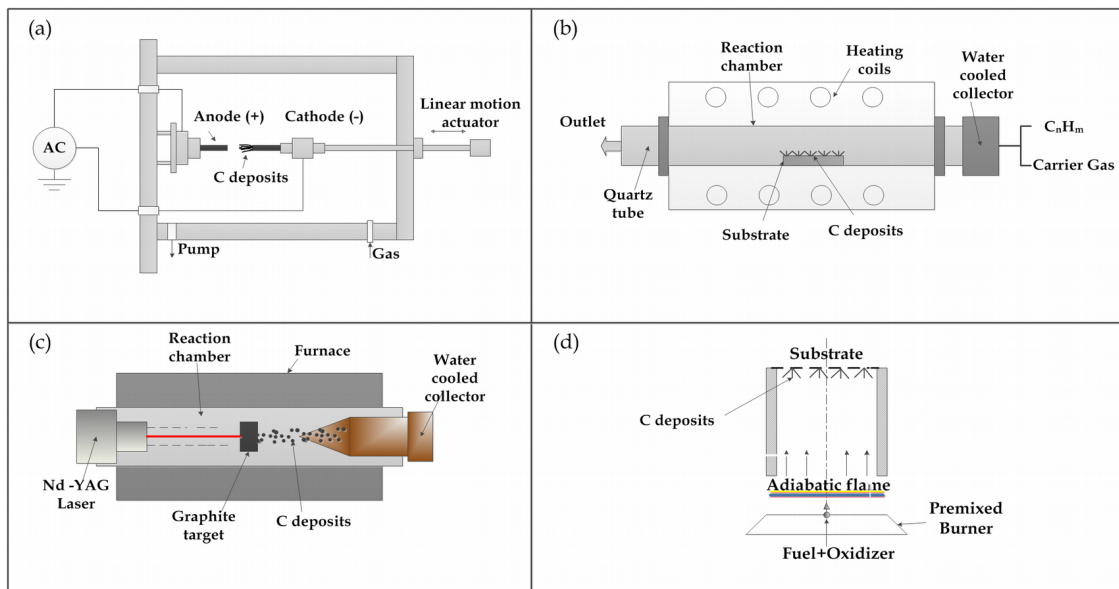


Figure 4 - Schematic representation of methods used for carbon nanotube synthesis (a) Arc discharge (b) Chemical vapor deposition (c) Laser ablation (d) hydrocarbon flames [23]

2.2.5. Properties

Carbon nanotubes exhibit various attractive properties, in part due to being a graphene based structure. These include:

Strength – carbon nanotubes are one of the strongest materials yet discovered in terms of tensile strength and stiffness, due to the covalent sp^2 bonds formed between the individual carbon atoms. Yu et al. [24] have reported that multi-walled carbon nanotubes were tested to have a tensile strength between 11 and 63 GPa, and that the Young's modulus of the outer layer varied between 270 to 950 GPa. However, CNTs are not nearly as strong under compression. Because of their hollow structure and high aspect ratio, they tend to undergo buckling when placed under compressive, torsional, or bending stress [25].

Wettability – the contact angles of most as-synthesized CNT arrays with water are over 160° , exhibiting a super-hydrophobic property. However, it has been found that by applying a voltage as low as 1.3V, the super-hydrophobic surface can be switched to super-hydrophilic [26].

Electrical properties – it is very interesting to note that the classification of CNTs based on electrical conductivity depends of the nanotube indices, m and n . Indeed, if the nanotube has the armchair configuration, it is metallic; if $m - n$ is a multiple of 3, then the nanotube is semiconducting with a very small band gap. In any other case, it is simply a moderate semiconductor [18]. They can carry an electrical current density of $\sim 4 \times 10^9$ A/cm², which is three orders of magnitude higher than a typical metal, such as copper or aluminium [27].

Thermal properties – measurements show that a SWCNT has a room temperature thermal conductivity along its axis of about 3500 W/mK, which is very high when compared, for example, to copper, a metal well known for its good thermal conductivity, which transmits only 385 W/mK. Across its axis, however, the thermal conductivity is very small, of about 1.52 W/mK [23]. The cylindrical shape of the CNTs is responsible for this anisotropy.

2.2.6. Applications

Certain companies [28] have already implemented CNT technology as of today, although nanotubes have only mostly been applied in their bulk form, that is, as a mass of rather unorganized fragments of nanotubes. The gains of tensile strength due to reinforcement will never stand close to the full potential of CNTs, but they may, nonetheless, be sufficient for certain applications. Their use has also been reported in pharmaceutical and medicine [29], for example due to their high surface area that is capable of adsorbing or conjugating with a wide variety of therapeutic and diagnostic agents (drugs, genes, vaccines, antibodies, biosensors, etc.)

2.3. Nanocones

2.3.1. Definition

Carbon nanocones are conical structures which are made predominantly from carbon and which have at least one dimension of the order one micrometer or smaller.

2.3.2. Discovery

Research into CNC's started almost at the same time as the discovery of CNT's in 1991 by Iijima [20]. In 1992, Ge et al. [30] reported the observation of carbon cones, together with tubules. The observed cones were up to 24 nm in length and 8 nm in base diameter, and although they only observed cones with the same opening angle of 19°, they also predicted that there were another four possible angles. The cones were artificially generated by quenching of hot carbon vapor on a graphite substrate. They also reported how a carbon nanocone can be modelled by rolling a sector of a sheet around its apex and joining the two sides.

Later, in 1997, Krishnan et al. [31] successfully confirmed the predictions of Ge et al. that there are five possible opening angles. The CNCs were generated through pyrolysis of hydrocarbons (Figure 5).

Just like in the case of CNTs, CNCs can also be categorized as single-walled or multi-walled. MWCNCs are the common case. SWCNCs, on the other hand, are much harder to obtain. One group [32] found that using the CO₂ laser ablation of carbon at room temperature produced aggregates of SWCNCs.

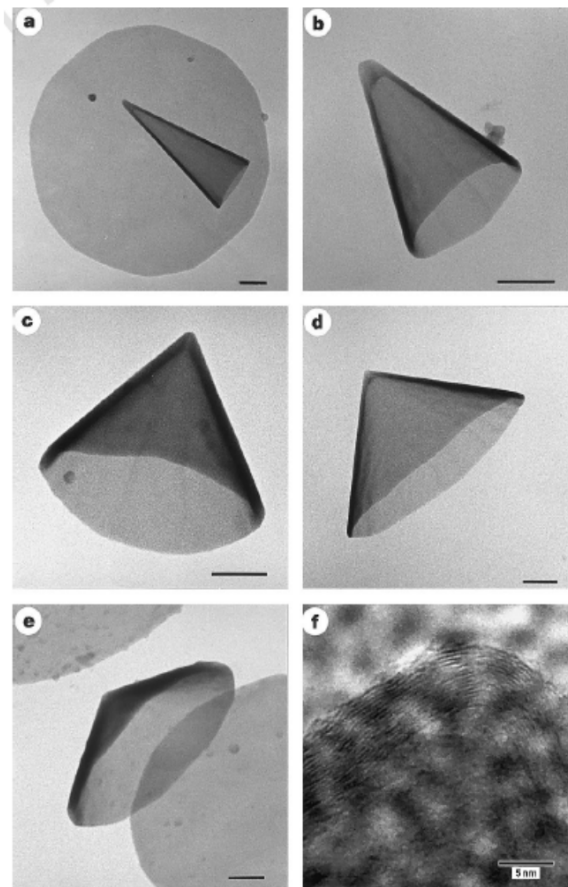


Figure 5 - TEM of the five types of cones in Krishnan's sample, a) apex angle = 19.2°; b) apex angle = 38.9°; c) apex angle = 60°; d. apex angle = 86.6°; e) apex angle = 123.6°; f) High resolution of image of a cone tip. [33]

2.3.3. Modelling of SWCNCs

The modelling of a SWCNC can be carried out by starting out, as in case of SWCNTs, with a graphene sheet. In this case, the sheet should be circular, so as to produce a cone in the end. First, one must cut off a slice of the sheet for some specific angle, and then fold it to close the sheet. As graphene has a hexagonal six-fold symmetry, removing angular slices of 60, 120, 180, 240 or 300° (the so-called disclination angle) starting from the circle center (placed in the center of a hexagon ring), and then connecting the two border lines, a perfect junction should be obtained for the closure

line upon folding, as predicted by Ge et al. [34] (Figure 6). This way, a pentagon (Figure 7), a square, a triangle, a line or a spire structure is formed in the cone center for the disclination angles of 60, 120, 180, 240 or 300° respectively [33], [35], [36], [37], [38], [39], [40].

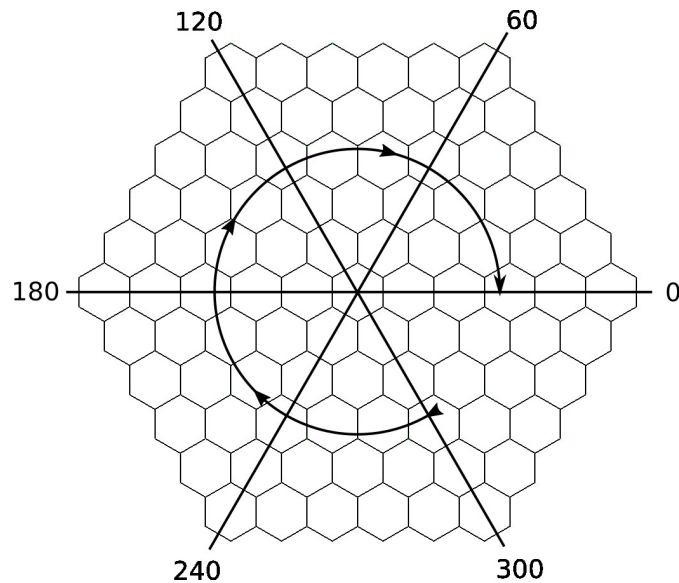


Figure 6 - The five possible nanocone disclination angles

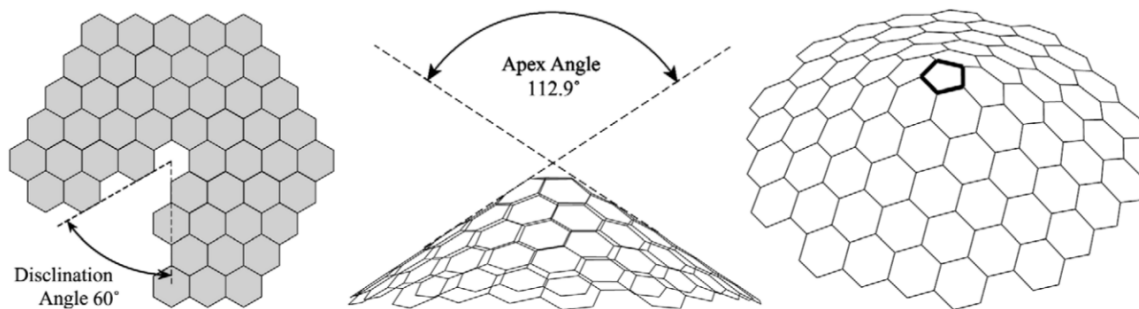


Figure 7 - Formation of a nanocone from a graphene sheet with a 60 disclination. This results in a cone with a single pentagon, located on the cone apex, with an apex angle of 112.9°. Thick lines in the image on the right show the location of the pentagon. There is only one possible nanocone structure with a single pentagon [39]

A different technique consists in removing and folding multiple sectors (from 2 to 5) of 60° starting from different points close enough to the circle center. Curling any of the sectors between the indicated lines leads to perfect closure of the honey comb network. For each sector removed, one pentagon must be formed on the honeycomb lattice, which means that a maximum of five pentagons may be formed, for a disclination angle of 300°. The values of the five possible cone angles are given by $2 \times \arcsin(1 - n / 6)$, or $2 \times \arcsin(1 - \Theta / 360)$, where n is the number of pentagons in the structure

and Θ is the disclination angle in degrees [33], [39], [41], [43]. Thus, for the five possible closed cone structures the apex angles will be 112.9° , 83.6° , 60° , 38.9° and 19.2° (Figure 8).

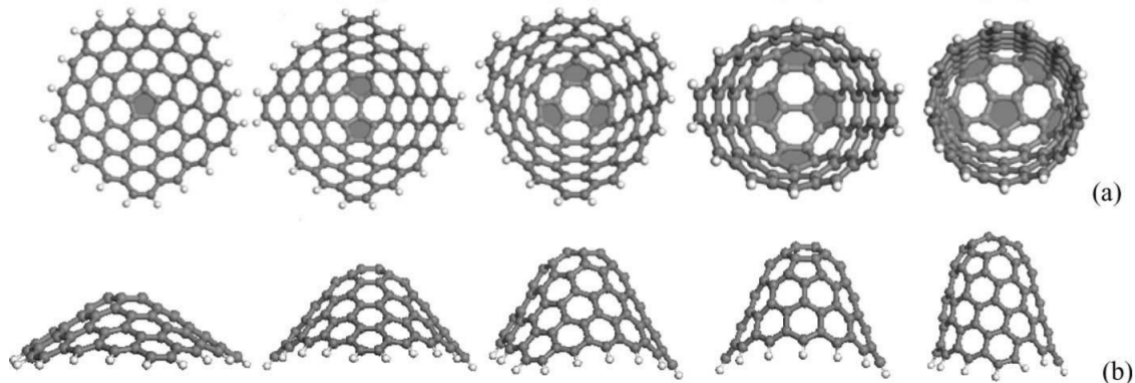


Figure 8 - Optimized structures of five CNCs with one, two, three, four, and five pentagons, respectively. The pentagons colored in light gray are displayed. (a) The top view. (b) The side view. [35]

Unlike the case where there is only one pentagon in the cone, when more pentagons are incorporated there is more than one possible cone structure [39] (Figure 9).

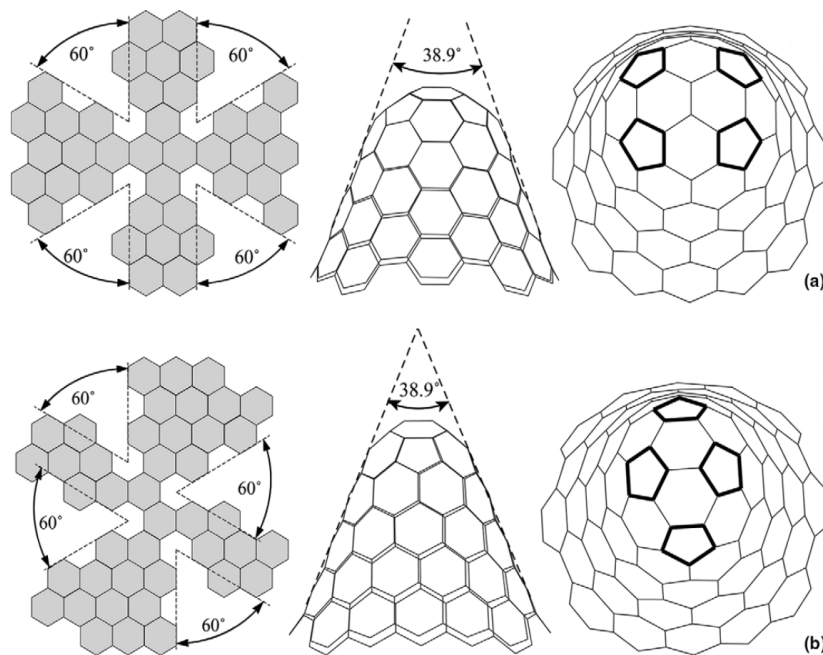


Figure 9 - Formation of nanocones from a graphene sheet with a total of 240° disclination. There are four pentagons in the resulting structure, with an apex angle of 38.9° . Thick lines in the images on the right show the location of the pentagons. There are many possible structures corresponding to different locations of the pentagons, with two shown here: (a) structure with the pentagons arranged in a rectangular fashion with twofold symmetry, (b) structure with pentagons arranged in a diamond pattern with twofold symmetry. [39]

Besides the closed cones described above, there are also open nanocones [39] (Figure 10). Open cones can have the closed cone structures discussed above, with one to five angular sectors removed, the only difference being that they lack the top part, or apex of the cone, and thus there are no pentagons or other smaller rings. Open cones are usually more flexible than closed cones, as they lack the constraints posed by the apex complex structures [33].

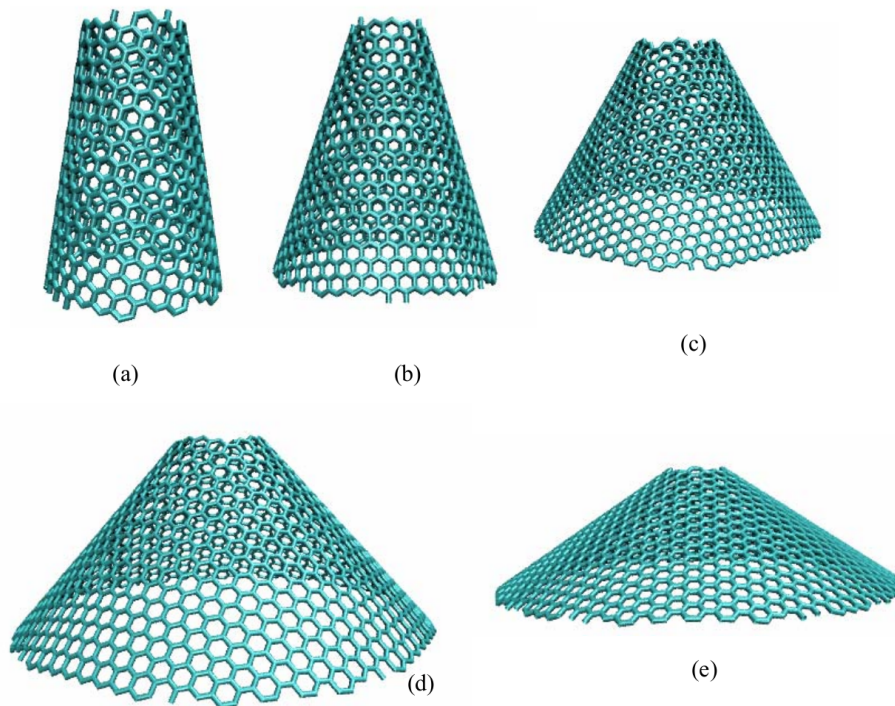


Figure 10 - Schematic illustrations of CNCs with different opening angles (a). TD = 60°, apex angle = 19.2°; (b). TD = 120°, apex angle = 38.9°; (c). TD = 180°, apex angle = 60°; (d). TD = 240°, apex angle = 86.6°; (e). TD = 300°, apex angle = 123.6°. [33]

2.3.4. MWCNCs modelling

Unlike the isolated (SWCNC) cones, multi-walled carbon nanocones (MWCNCs) consist of multiple conical layers that make it harder to verify their structures. There are three structures proposed for this kind of cones:

Stacked closed cone structures – individual cones are stacked on top of each other, being offset from each other in the same way [44]. This structure is referred to as a turbostratic structure. The advantage of this model is its simplicity, but it fails to explain electronic conductivity in the axial direction and the strong axial properties of nanofibers [39]. This structure is also the exception, rather than the rule, as in almost every other case, conical structures with a variety of non-standard apex angles are obtained.

Stacked open cone structures – same as the above case, but with open cones.

Cone-helix structures – Jaszczak et al [36] have reported a broad apex-angle distribution in one of their works. They found that the cone-helix model (Figure 11) proposed by Double and Hellawell [45] fitted their observations. This model is based on growth around a positive disclination with a screw dislocation component. As a graphene sheet wraps around the disclination, adjacent overlapping layers are rotated with respect to one another by an angle equal to the disclination angle. Another parameter, the degree of graphitic alignment, was adopted to justify if the overlapped graphenes are energetically stable enough. It evaluates how many carbon atoms among the layers can achieve the alignment of lattice points in graphite crystal. When the two parameters are combined, a series of conical graphenes with all possible apex angles can be generated [42]. Although this model can explain the broad range of observed apex angles, as well as the electronic and mechanical properties that the stacked cones model fails to predict, the construction of this structure is much more complicated.

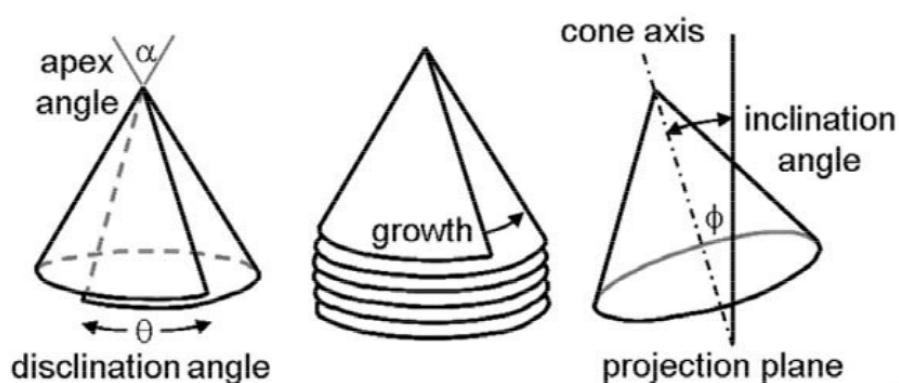


Figure 11-Schematic of cone formation by the cone-helix model [36]

2.3.5. Energetic considerations

Calculations made using the molecular modelling program SPARTAN for various nanocone structures with four pentagons (Figure 9 and Figure 12) were carried out [39], and are shown in Table 1:

Simulation results for various nanocone structures with four pentagons

Nanocone structure	Rings in nanocone	Atoms in nanocone	Energy of nanocone	
			E^{MMFF} (kcal/mol)	E^{HF} (au)
Fig. 2a	20	62	660.13	-1882.6
Fig. 2b	24	72	684.27	-2040.3
Fig. 4b	23	72	631.72	-2192.1
Fig. 4c	25	72	775.36	-2266.1
Fig. 4d	23	72	612.89	-2192.0

The results show the energy of the final optimized structure employing the molecular mechanics force field (MMFF) model and the Hartree-Fock (HF) approximation in SPARTAN.

Table 1 - Results of the SPARTAN simulation [39]. In this work, the nanocones in question are those from figures 9a, 9b, 12b, 12c and 12d, instead of 2a, 2b, 4b, 4c and 4d, respectively

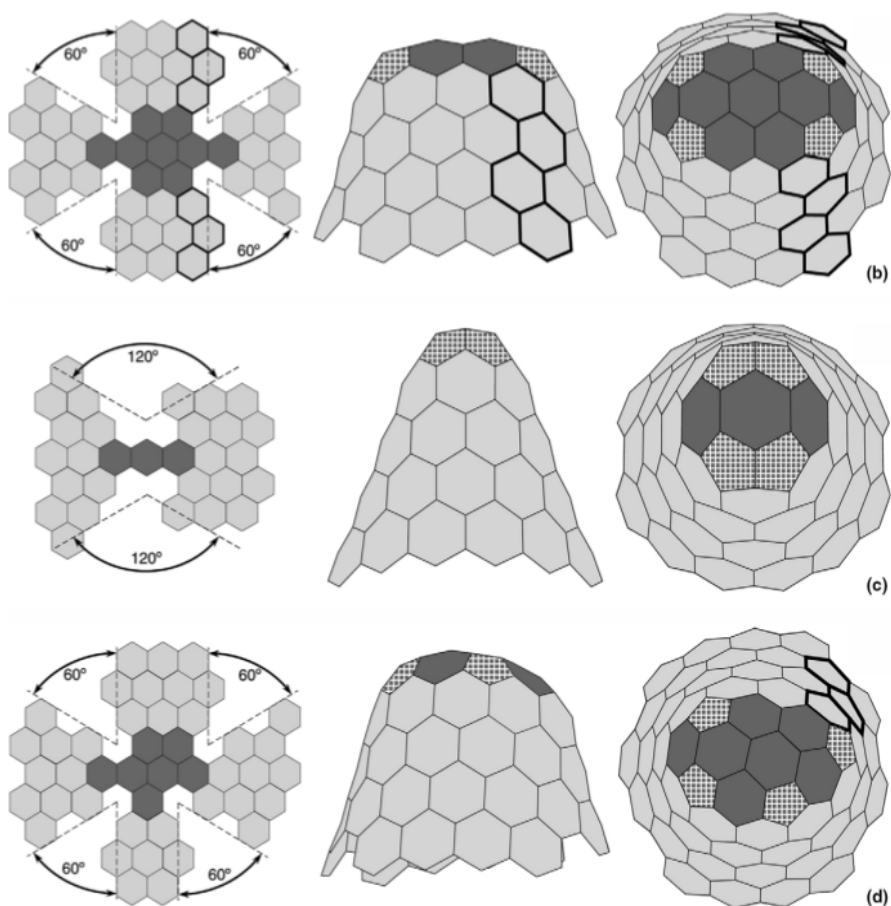


Figure 12 - Remaining Nanocone Isomers used for the simulations carried out in SPARTAN [39]

Although it is generally believed that structures with isolated pentagons are more unstable than those with adjacent pentagons, the results from this work do not allow for any such confirmation. The Hartree-Fock ab-initio calculations suggest that the structures with adjacent pentagons are more stable, whereas the result from the molecular mechanics force field model suggest otherwise.

According to Double and Hellawell [45], cones with smaller apex angles may be disfavored because of the higher elastic energy due to the bending needed to form the corresponding disclinations; however it is also necessary to take into account that cones with smaller apex angles have larger surface areas at their tips, and this should favor smaller apex-angle cones.

Cone angles corresponding to disclination angles that are equal to integer multiples of 60° should be the most favorable, since they preserve the graphite crystal structure, as explained above. But, if every cone would be formed due to the incorporation of pentagons into a graphene sheet, only cones with discrete angles ($60, 120, 180, 240$ or 300°) would be observed, and those cones would be very sharp. However, not only can cones with angles different from the expected discrete values be observed, but they can also present various different shapes [36] (Figures 13 and 14), and this suggests that there is more than one possible formation mechanism for nanotubes.

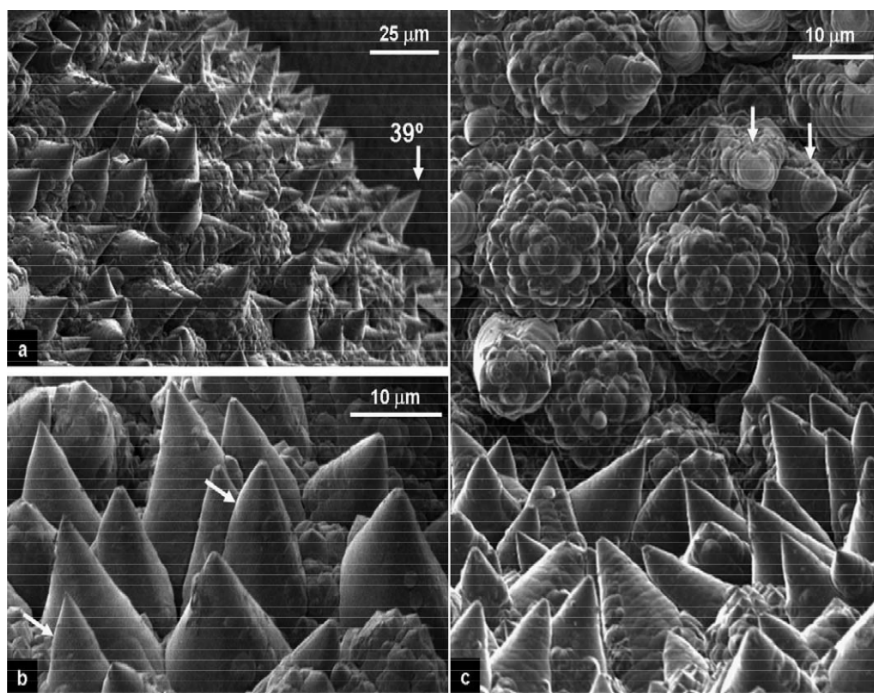


Figure 13 - FESEM images of a cone-covered graphite aggregate. (a) Low-magnification image showing complete coverage of the aggregate surface with conical structures. A $\sim 39^\circ$ cone is marked by an arrow. (b) Higher magnification image of the sample showing a variety of large cones with different apex angles and sharp and blunt tips. Arrows show changes in the apex angle. (c) Close up view of two surfaces which are almost perpendicular and show different cone morphologies—large cones on one surface and globular (artichoke-like) structures on the other. The latter are clusters of large-angle cones. Arrows show some of the cones that are ripped on the side. [36]

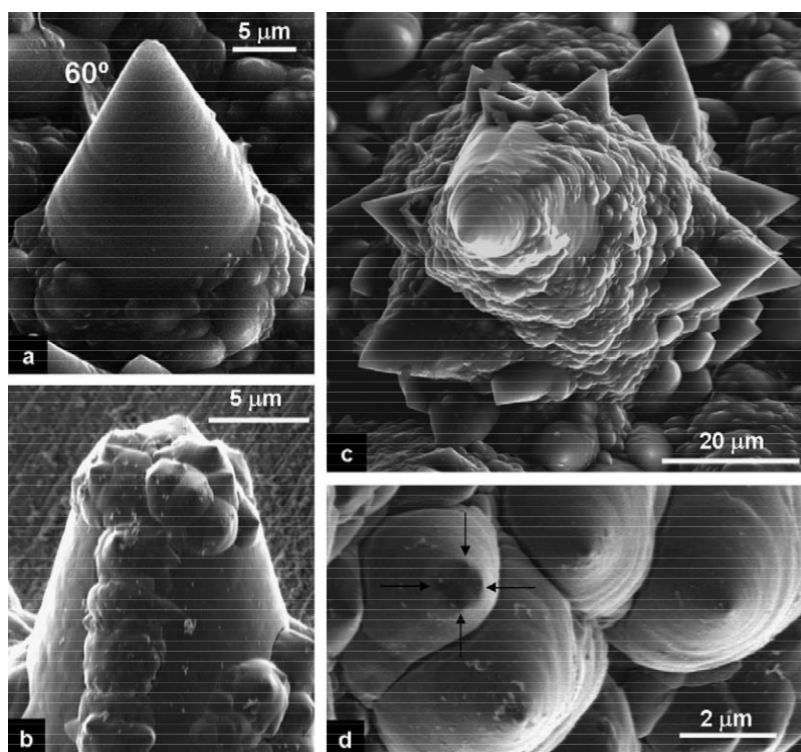


Figure 14 - Typical cone morphologies. (a) SEM image of a cone with a 60° apex angle, the most common apex angle. The slightly uneven surface of the cone suggests layer growth. (b) FESEM and (c) SEM images of large cones with numerous smaller cones growing on their surface. Smaller cones covering surfaces of large cones have a broad distribution of shapes, but large apex angles prevail (c). (d) FESEM image of four cones having sharp and broad tips (multiple tips are marked by arrows). The cones are oriented to reveal their circular cross sections around the tips and layered growth (ripples). [36]

2.3.6. Synthesis

Large-scale production of conical carbon nanostructures is possible through pyrolysis of hydrocarbons in an industrial plasma torch process. This is a patented process, known as the Kvaerner Carbon Black & Hydrogen Process (CBH). The CBH is an emission-free industrial process that decomposes hydrocarbons directly into carbon and H_2 , based on a specially designed plasma torch, with a plasma temperature above $2000^\circ C$. The solid output consists of a significant amount of open-ended carbon nanocones (20%), as well as a large number of flat carbon discs (70%), the rest being carbon black [43].

2.3.7. Properties and applications

Carbon-based electron field emitters have been reported to show good electron field emission properties due to their capability of emitting high currents at low electric field [46], [47]. CNCs have been ideally suited for use as scanning probe tips and electron field emitters due to their small size and high stiffness [34], [48]. Shenderova et al. [49] predicted the good field emission properties for a

nanocone with one pentagon. Qu et al [35] have found that the cohesive and formation energies for various types of CNCs are dependent on the cone angles, while the work function, local density of states, redistribution of the charge, and field emission pattern are sensitive to the morphologies of CNCs that are governed by the position of pentagonal rings in the cone apex. Most importantly, the nanocone with three pentagons in the cone apex exhibits the best field emission property. In this same work, a broad apex-angle distribution has been reported.

2.4. Fullerenes

2.4.1. Discovery

Before its discovery by Kroto et al. [19], the existence of fullerenes had already been proposed [50], when Osawa, after noticing the similarity between a corannulene molecule and a football shape, proposed that a molecule with the exact shape of a football could exist. Attempts were then made to predict the exact structure of C_{60} , but they ended out not receiving the due attention by the scientific community [51], until finally, in 1985, Kroto et al. [19], “during experiments aimed at understanding the mechanisms by which long-chain carbon molecules are formed in interstellar space and circumstellar shells”, ended up “producing a remarkably stable cluster consisting of 60 carbon atoms.”, a discovery which earned them the Nobel Prize in Chemistry in 1996.

Fullerenes are closed carbon-cage molecules containing only pentagonal and hexagonal rings [52], where every atom has bonds with exactly 3 other atoms [53], idealized as sp^2 hybridized atoms [54]. This bonding framework forms a polyhedron, with an atom at each vertex, a bond along each edge, and a ring around each face [52].

2.4.2. Euler’s Theorem

One of the best-known properties of polyhedra is Euler’s theorem, which states that the numbers of vertices (V), edges (E), and faces (F) in a polyhedron are related by:

$$V + F = E + 2 \quad (1)$$

Therefore, a C_n fullerene with n vertices and $3n / 2$ edges, must have a number of faces $F = n / 2 + 2$, as in any special polyhedron with trivalent vertices. However, this result can be taken slightly further by defining the number of pentagons as p and the number of hexagons as h . The total number of vertices is then:

$$(5p + 6h) / 3 = n \quad (2)$$

since there are 5 and 6 atoms for each pentagon and hexagon, respectively, and each atom is shared by 3 faces. The total number of faces is

$$p+h=n/2+2 \quad (3)$$

The solution of the two equation system composed by equations (2) and (3) is $p = 12$ and $h = n / 2 - 10$. Hence all fullerenes contain 12 pentagons and $n / 2 - 10$ hexagons [52].

There is at least one fullerene for each even number of vertices $n \geq 20$, with the sole exception of $n = 22$. If one attempts to construct a trivalent planar graph with a single hexagon at the centre and every other face a pentagon, the construction fails because the final face of the graph has to be another hexagon, giving a total of two hexagons and 24 vertices, i.e., the unique fullerene structure of C_{24} . Odd numbers of vertices are also precluded for fullerenes, because the number of edges and faces must be integer. [52]

2.4.3. Isomers

There are infinitely many fullerenes, the number of isomers growing as N^9 for N carbon atoms [52] (Figure 15). For example, C_{60} has 1812 isomers, but C_{180} , already has 79,538,751 isomers. The order N^9 -growth follows from Thurston's parameterization of triangulations on geodesic domes [55].

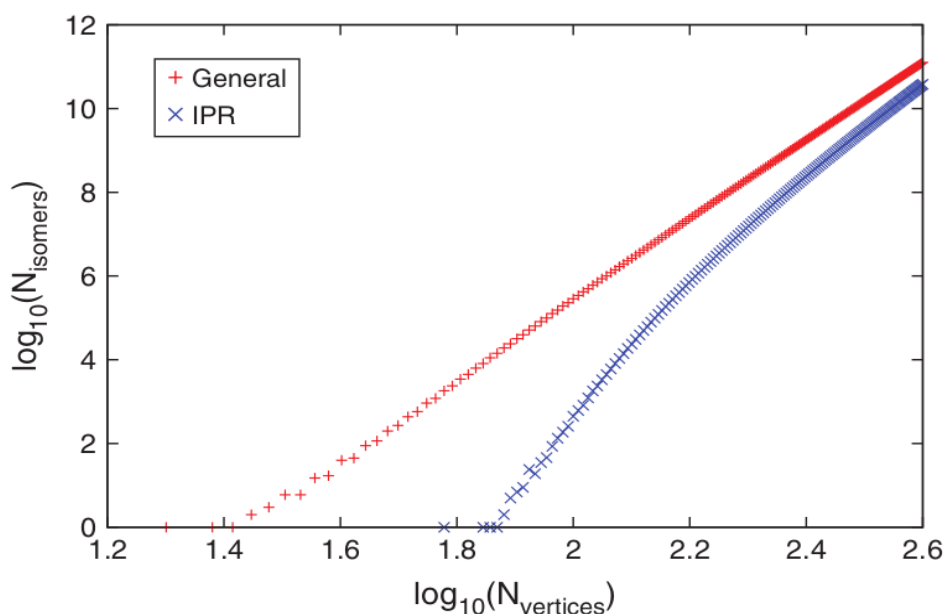


Figure 15 - Number of distinct (non-isomorphic) fullerene isomers C_N (with and without fulfilling the IPR) with increasing number of carbon atoms N up to $N = 400$ (double logarithmic scale). [54]

These isomers come in many different shapes depending on the distribution of the pentagons over the surface of the fullerene, as can be seen in Figure 16.

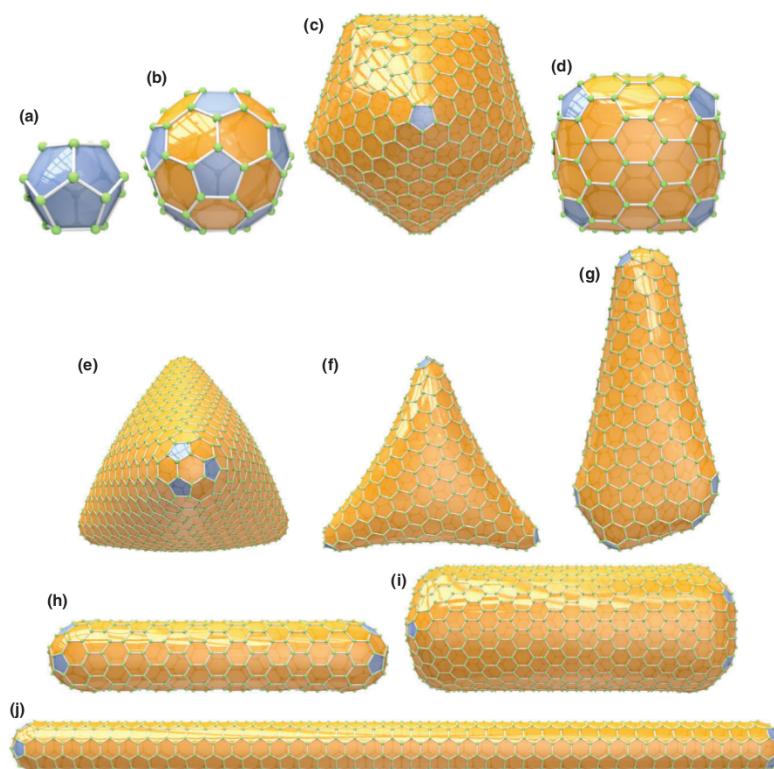


Figure 16 - A selection of different 3D shapes for regular fullerenes (distribution of the pentagons D_P are set in parentheses). 'Spherically' shaped (icosahedral), for example, (a) $C_{20} - I_h$, (b) $C_{60} - I_h$, and (c) $C_{960} - I_h$ ($D_P = 12 \times 1$); barrel shaped, for example, (d) $C_{140} - D_{3h}$ ($D_P = 6 \times 2$); trigonal pyramidally shaped (tetrahedral structures), for example, (e) $C_{1140} - T_d$ ($D_P = 4 \times 3$); (f) trihedrally shaped $C_{440} - D_3$ ($D_P = 3 \times 4$); (g) nanocone ormenhir $C_{524} - C_1$ ($D_P = 5 + 7 \times 1$); cylindrically shaped (nanotubes), for example, (h) $C_{369} - D_{5h}$, (i) $C_{1152} - D_{6d}$, (j) $C_{840} - D_{5d}$ ($D_P = 2 \times 6$). The fullerenes shown in this figure and throughout the paper have been generated automatically using the Fullerene program. [54]

2.4.4. Isolated Pentagon Rule

The single most important consequence of steric strain in the fullerenes is the IPR, which says that the most stable fullerenes are those in which all the pentagons are isolated, that is, no pentagon shares an edge with any of the other pentagons on the fullerene. This rule was first mentioned by Kroto in 1987 [56]. The first fullerene to exhibit an isomer that respects the IPR is C_{60} .

2.4.5. Pentagon and hexagon indices

The IPR is a very effective rule to determine which fullerene isomers are thermodynamically more stable ones. However, for fullerenes with more than 70 carbon atoms, the number of IPR isomers increases rapidly [52].

An additional rule used to determine the most stable fullerene isomers has been proposed by Raghavachari [57], who argues that steric strain will be minimized beyond the IPR when the pentagon-

induced curvature is distributed as uniformly as possible over the fullerene surface. In order to quantify the uniform curvature rule, Raghavachari defines the neighbour index of each hexagon in a fullerene as the number of other hexagons to which it is adjacent. The arrangement of the pentagons in a fullerene can be characterized within the neighbour index scheme by a signature of the form $(h_0, h_1, h_2, h_3, h_4, h_5, h_6)$, where h_k denotes the number of hexagons with neighbour index k . Since in an isolated-pentagon fullerene, every hexagon is adjacent to at least another three, it is only reasonable to dispense the first three indices (since they are all zero).

The sum of the entries in the hexagon neighbour index signature is simply the total number of hexagons. As such, for an isolated-pentagon fullerene, we have:

$$h_3+h_4+h_5+h_6=n/2-10 \quad (4)$$

The sum of the edges of all the hexagons can be obtained by multiplying 6: $3n - 60$. Sixty of those (12×5) are shared with pentagons (for IPR isomers), so:

$$3h_3+4h_4+5h_5+6h_6=3n-120 \quad (5)$$

Eliminating n :

$$3h_3+2h_4+h_5=60 \quad (6)$$

which is simply the number of edges that are shared between pentagons and hexagons.

According to Raghavachari's [57] argument that the indices of all the hexagons in the structure should be as similar to one another as possible and taking into account the constraints of the above equations, it follows that there are three optimum solutions:

- $(20, 0, 0, 0)$ for n equal to 60;
- $(0, 30, 0, 0)$ for n equal to 80;
- $(0, 0, 60, 0)$ for n equal to 140.

Clearly, no $(0, 0, 0, n)$ solution exists because some hexagons must be adjacent to pentagons. It is also very interesting to verify that the three cases correspond to IPR fullerenes with maximum (icosahedral) symmetry. However, these are only three values of n , and it would be interesting to expand the range of values. By allowing for a given fullerene to have two different, but consecutive, hexagon neighbour indices, the solutions can be stated in the following form:

- $(80 - n, 3n/2 - 90, 0, 0)$ for $60 \leq n \leq 80$;
- $(0, 70 - n/2, n - 80, 0)$ for $80 \leq n \leq 140$;

- (0, 0, 60, n/2 - 70) for 60 ≤ n ≤ 80;

The rationale applied above for large fullerenes, with more than 70 atoms, can be used for smaller fullerenes, considering not the hexagon indices, but the pentagon indices, that is, the number of pentagons each pentagon is adjacent to, $(p_0, p_1, p_2, p_3, p_4, p_5)$.

In this case, the total number of fused pentagon pairs (pairs of adjacent pentagons) in an isomer can be given by (dividing by 2, as pentagons are counted twice):

$$N_p = \sum_{k=1}^5 k p_k / 2 \quad (7)$$

The value N_p can be used as a qualitative filter for selecting stable isomers. The lower the value of N_p , the more stable the isomer. Although it can also be used for fullerenes with more than 70 carbon atoms, it does not provide much help, as fused pentagons seldom occur, and for IPR fullerenes, N_p is always 0.

2.4.6. Fullerene Graph Generation Algorithms

There has been an intense activity in the field of topological and graph theoretical descriptions of fullerenes over the past 20 years, to the extent that it has become a major sub-discipline within mathematical chemistry. In order delve into an explanation of some of the proposed algorithms, it is first necessary to go over some concepts [58].

2.4.6.1. Graph Theory

Graph theory covers the study of graphs, which are mathematical structures used to model pairwise relations between objects. A very simple example of a graph can be seen in Figure 17.

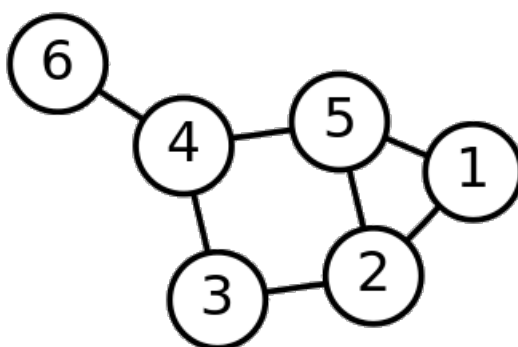


Figure 17 - A 6 vertice graph [59]

Graphs consist of vertices (or nodes) related by edges (or connections). If two vertices are connected by an edge they are said to be adjacent.

The number of edges starting at a vertex (or the number of adjacent vertices) is the valency or degree of that vertex. A node with valency n is said to be n -valent. A graph is called n -regular if every vertex is n -valent.

At this level, a graph represents the information that chemists sometimes call the connection table of a molecule. For a visual representation, it is also necessary to assign (2D or 3D) coordinates to the vertices, and this is known as embedding of the graph (into 2D or 3D space).

When a graph is to be drawn on paper, it is desirable to draw the edges so that they don't cross each other. If this is possible (the graph has a 2D embedding that permits intersection-free edge drawing), the graph is said to be planar. Looking at a drawing of a graph with more or less straight edges, faces are perceived as well as edges and vertices. A face is a 2D region bounded by some of the graph's edges. (the special face defined by the "outside of the graph must also be included"). Depending on the number of edges bounding it, a face can be a triangle, quadrangle, pentagon, hexagon, and so on. The number of bounding edges is the face's size.

The notion of a dual is also a very important one to be aware of. If one pictures a planar graph with its vertices, edges, and faces, one can then obtain a new graph, the dual of the original graph, as follows: first, it is necessary to create a new (dual) vertex in the interior of each face of the original graph. For each original edge, one must find the faces bounded by that edge and connect the dual vertices inside them by a new (dual) edge, crossing the original one. Dualizing exchanges vertex degree and face size: the degree of each dual vertex is clearly equal to the size of the corresponding original face: each edge bounding the original face creates a dual edge starting at the dual vertex. Likewise, a dual face's size is equal to the degree of the corresponding original vertex.

A graph is called k -(vertex)-connected if one must remove at least k vertices in order to separate the graph into disconnected parts. If k vertices are also "enough" to separate the graph (there is some set of k vertices that, when removed, achieves the separation), we say the graph is exactly k -connected. The number k is then called the connectivity of the graph. (Removing a vertex implies removing all edges adjacent to it.)

A fullerene can be represented as a graph, if one considers the vertices of the graph to represent atoms and the edges to represent bonds (Figure 18). Constructing a fullerene graph is equivalent to constructing its polyhedron, because the bonding connectivities of the two are the same. A list of adjacent vertices in the graph is the same thing as a list of adjacent carbon atoms in the polyhedron, for example, and this is how a computer might represent bonding connectivity. [58]

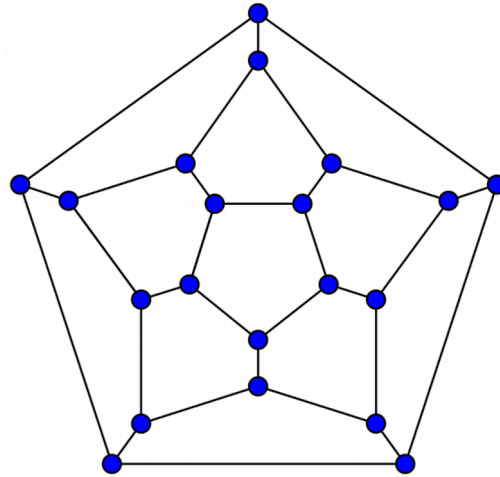


Figure 18 - Graph of C₂₀ [60]

2.4.6.2. Fullerene Duals

The duals of fullerene polyhedra are deltahedra, which are polyhedra made up exclusively of triangular faces. It is easy to see from the above discussion that the dual of a fullerene polyhedron with n vertices has n triangular faces and 12 five- and $n / 2 - 10$ six-valent vertices.

The most often quoted examples of dual pairs are the octahedron and the cube, the icosahedron and the dodecahedron. The tetrahedron completes the set of Platonic solids and is the only regular polyhedron that is self-dual [52] (Figure 19).

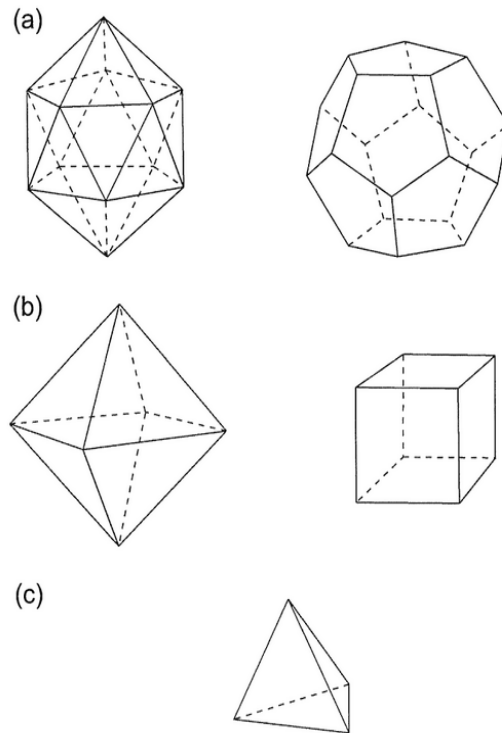


Figure 19 - Two dual pairs are (a) the icosahedron and the dodecahedron, (b) the octahedron and the cube. The tetrahedron (c) is the only self-dual Platonic solid. [52]

One of the many reasons why fullerene duals are interesting is that it is often easier to construct a fullerene by first constructing its dual. This trick forms the basis of several methods for constructing fullerene isomers, such as the leapfrog method. Once the fullerene dual is known, the fullerene can easily be reconstructed, because each set of three mutually adjacent vertices in a fullerene dual encloses a unique vertex of the corresponding fullerene polyhedron (Figure 20). It follows that each vertex of the fullerene can be uniquely associated with three mutually adjacent vertices of its dual, and moreover that two fullerene vertices will be adjacent if and only if two of the three dual vertices with which they are associated are the same. These facts allow one to obtain a list of adjacent vertices in the fullerene from a list of adjacent vertices in its dual, which is tantamount to performing the reconstruction. [52]

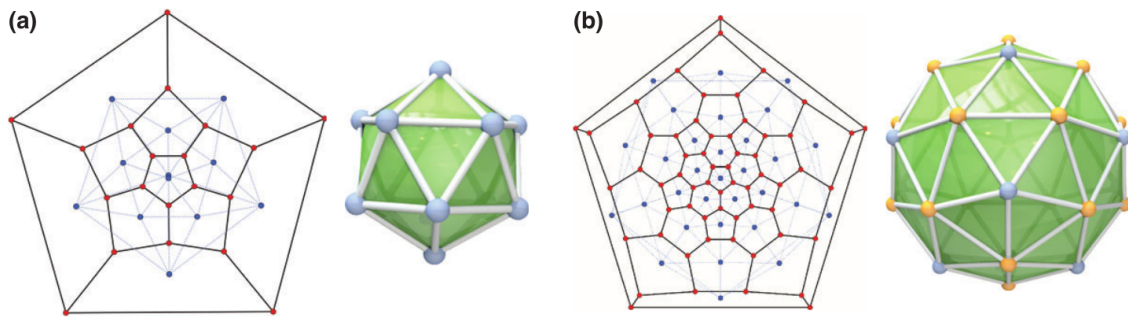


Figure 20 - Planar embeddings of fullerene graph and dual (blue color and dotted lines for the dual representation), and 3D embeddings of the duals: (a) $C_{20} - I_h$, for which the dual is the icosahedron; (b) $C_{60} - I_h$, for which the dual is the pentakis-dodecahedron. [55]

2.4.6.3. Ring-Spiral Algorithm

This algorithm was proposed by Fowler and Manolopoulos in *An Atlas of Fullerenes* [52]. It is based upon the following conjecture: “The surface of a fullerene polyhedron may be unwound in a continuous spiral strip of edge-sharing pentagons and hexagons such that each new face in the spiral after the second shares an edge with both (a) its immediate predecessor in the spiral and (b) the first face in the preceding spiral that still has an open edge.” Any face of the fullerene may serve as a starting point for the unwinding of the spiral. The spiral of the dodecahedron, C_{20} , can be seen in Figure 21.

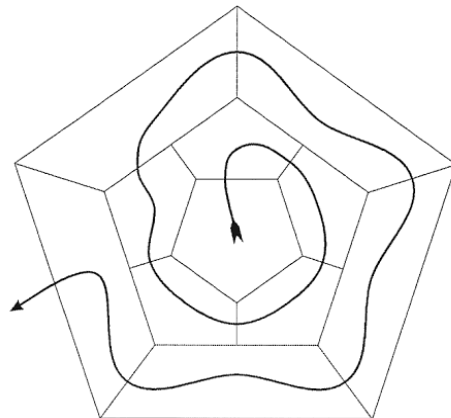


Figure 21 - The unique spiral of the dodecahedron, shown superimposed on the corresponding planar graph. Each new face in the spiral after the second is adjacent to both (a) its immediate predecessor and (b) the first face in the preceding spiral that still has an open edge [52]

Unfortunately, it has been proved that this algorithm fails for very large isomers [52]. The smallest fullerene for which a failing spiral can be found is a D_2 isomer of C_{28} . It has 168 spiral starts, 42 of which are distinct. Out of those 42, one of them fails. Despite the fact that there are fullerenes which fail to be described by a spiral in this manner, this is nonetheless a very attractive algorithm due to its simplicity.

2.4.6.4. Face-adding algorithms

Several authors have proposed fullerene graph generating algorithms in which the approach is to add faces to an existing graph, while considering different sites for addition at each step. It was proposed, for example, to grow a graph starting from a single face, and adding faces at each step [61], as well as using patches, which are sets of faces bounded by a simple cycle, that is, a cycle that traverses no vertex or edge twice [53]. Using C_{24} as a seed, Brinkmann et al. [62] were able to generate almost all fullerene graphs up to C_{200} .

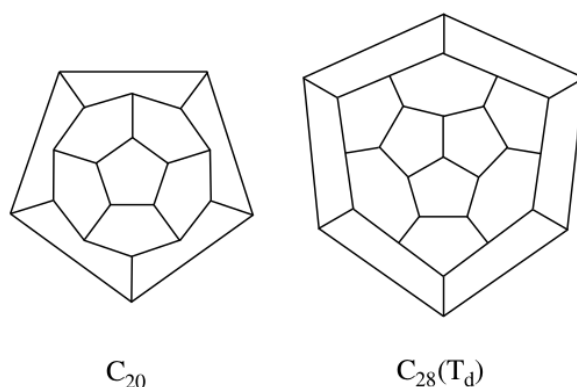


Figure 22 - Two of the basic fullerene isomers [63]

In order to fix the shortcomings of Brinkmann's algorithm, Hasheminezhad et al. [63] defined a set of patch replacements which allows for the generation of every fullerene graph, starting from C_{20} and $C_{28}(T_d)$ (Figure 22).

There are three sets of patch replacements, also called expansions:

- L_i – for $i \geq 0$, adds $i + 2$ faces, using a path of length $2i + 3$ that alternates left and right;
- B_{ij} – for $i, j \geq 0$, adds $i + j + 3$ faces, using a path of length $2i + 2j + 5$ that alternates left and right, except that bends $2i + 2$ and $2i + 3$ have the same orientation;
- F – adds 5 hexagonal faces;

Transformations L_0 , L_3 , $B_{0,0}$, $B_{3,2}$ and F , can be seen in Figure 23. All the transformations' dual representation, except F , in Figure 24:

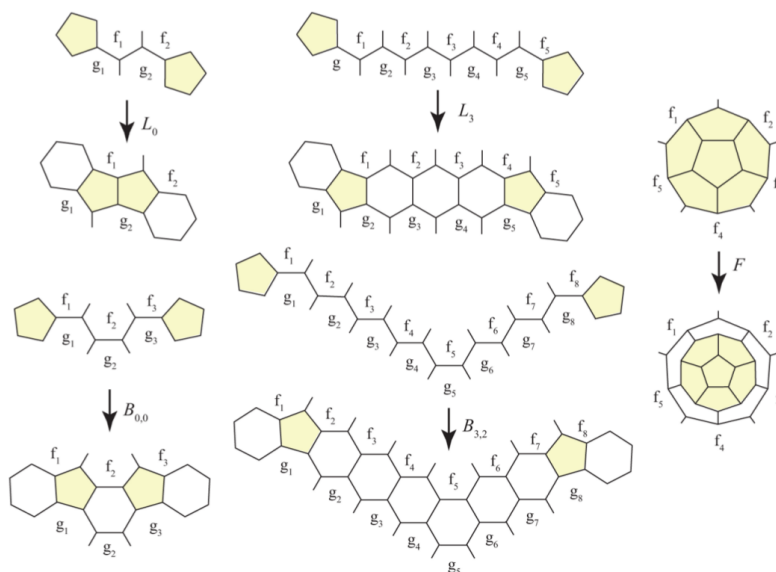


Figure 23 - Expansions L_0 , L_3 , $B_{0,0}$, $B_{3,2}$ and F [63]

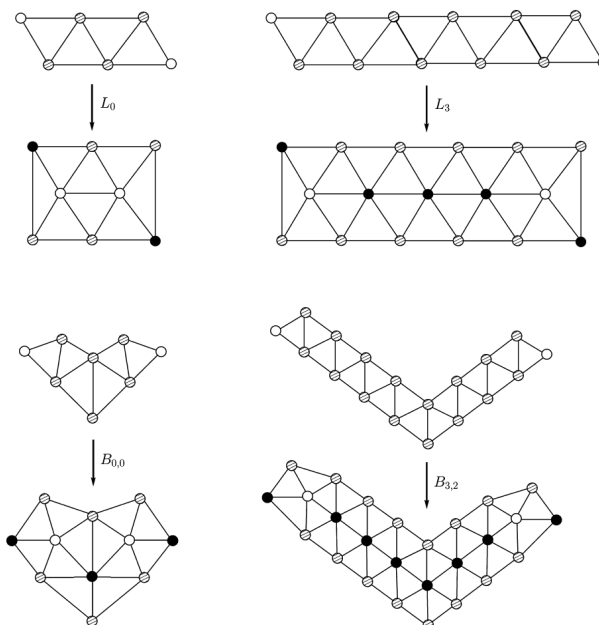


Figure 24 - The L and B expansions in dual representation [64]

They have also mentioned how it would actually be possible to generate every fullerene graph from C_{20} . If, for example, one removes the central atom of C_{28} (T_d) and suppresses the resulting atoms of valence two, resulting in the C_{24} fullerene. So, by generating C_{24} from C_{20} and then applying the inverse of the transformation that was just explained, it is possible to also generate C_{28} .

It is also worth mentioning the three algorithms established:

- **Algorithm 1** – every fullerene isomer except $C_{28}(T_D)$ can be constructed from C_{20} using expansions of type L , B , and F ;
- **Algorithm 2** – every fullerene isomer, except $C_{28}(T_D)$ and type-(5,0) nanotube fullerenes, can be constructed from C_{20} using expansions of type L and B ;
- **Algorithm 3** – every fullerene isomer with at most 300 atoms, except $C_{28}(T_D)$ and type-(5,0) nanotube fullerenes, can be constructed from C_{20} using expansions of type L .

Brinkmann et al. [64] made use of this set of operations to define a fast and complete algorithm that recursively generates all fullerene isomers up to a given maximal N . The algorithm was implemented in a computer program called *buckygen*, and the program was used to generate all fullerenes up to 400 vertices, which is available at the House-of-Graphs website [65].

2.4.6.5. Leapfrog Transformation

This transformation is more easily explained by example. Let us consider the C_{20} fullerene. By placing a vertex on the center of each face and linking them with each of the other vertices in the structure (five in case the vertex is in the center of a pentagon, six in case it is in the center of a hexagon), the result is a dodecahedron with 12 five-valent and 20 six-valent vertices (that is, vertices connected to another 5 and 6 vertices, respectively). Now, by converting the dodecahedron into its dual, the resulting structure is a truncated icosahedron with 12 pentagons and 20 hexagons, and 60 vertices. The different steps of this process can be seen in Figure 25.

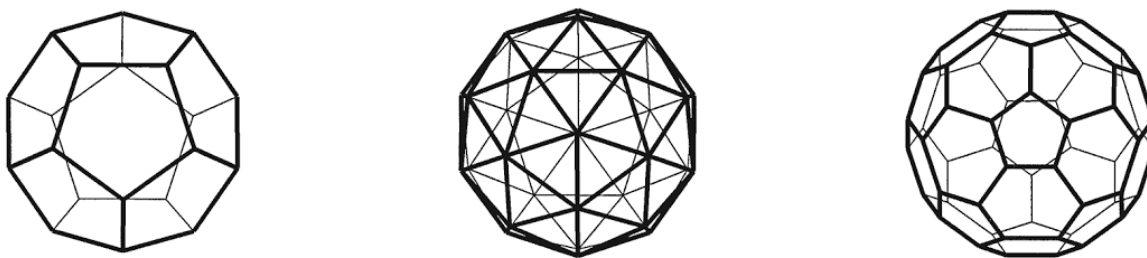


Figure 25 - Leapfrogging a dodecahedron: a dodecahedron is converted to a 32-vertex dodecahedron by omnicaapping, then to a truncated icosahedron by taking the dual [52]

So, to summarize it, a new fullerene can be generated from a smaller one by placing vertices on the center of each face and then dualizing the resulting structure. This method always yields a fullerene with three times the number of atoms of the original one. [52]

2.4.6.6. Fullerene Generation Software

Several programs to create fullerene isomers have been created by researchers. Program *Fullerene*, for example, is a general purpose open-source program that can generate any fullerene isomer, perform topological and graph theoretical analysis, as well as calculate a number of physical and chemical properties [55]. Other programs exist, like *fullgen* [53] or *Buckygen* [64].

2.4.7. Synthesis

The main process used to produce fullerenes is the arc discharge process, which consists in the sublimation of carbon in an inert atmosphere through an electric arc, expansion of the carbon in an inert gas and quenching of the carbon vapour, which leads some of the carbon clusters to assemble into fullerenes.

There are alternatives, however. In 1999, Fulcheri et al. [66] tested a process for fullerene production consisting of treating carbon powders through a 3-phase thermal plasma, and reported that the preliminary experiments carried out had confirmed that their process was a promising option for the mass production of fullerenes.

2.4.8. Applications

In a 2013 review paper [67], Lalwani and Sitharaman analyzed the advances in the development of fullerene-based magnetic resonance imaging and X-ray imaging contrast agents, drug and gene delivery vehicles and photodynamic therapy agents. They presented the examples of certain companies that make use of fullerenes in their products, such as for example an anti-aging cream, fullerene-based drugs for AIDS and neurodegenerative diseases such as Parkinson and Alzheimer's.

Bakry, Rania et al. [68], report the aspects of medical applications of fullerenes. A similar review, by Bosi, Susanna et al. [69], focuses on the biological applications of fullerenes.

3. Algorithms and Results

The algorithms developed in this work were based upon existing scientific research, and implemented in the C programming language [70]. The source code, totalling a number of 2356 lines, is divided among six files. In this section, the algorithms used to generate nanotubes, nanocones and fullerenes will be explained and parts of the source code will be presented in code text segments. The results of the implemented algorithms are also shown, (the molecules that GAMGI can generate).

3.1. Nanotubes

3.1.1. Graphene Base Vectors and Atomic Positions

As mentioned in section 2.1, graphene can be described by a 2D hexagonal lattice with vectors $\mathbf{a}_1 = a \cdot \mathbf{x}$ and $\mathbf{a}_2 = a \cdot \sin(120^\circ) \cdot \mathbf{x} + a \cdot \cos(120^\circ) \cdot \mathbf{y}$, with carbon atoms at positions (0, 0) and (2/3, 1/3). The lattice parameter $a = 2.467 \text{ \AA}$ is thus simply related with the CC bond distance between the two atoms: $a = \text{bond} \cdot \sqrt{3}$.

```
void gamgi_chem_graphene_vectors (double bond, double *a, double *b)
{
/*****
 * graphene lattice vectors *
*****/

a[0] = bond * sqrt (3.0);
a[1] = 0.0;
b[0] = -0.5 * bond * sqrt (3.0);
b[1] = 1.5 * bond;
}

/*****
 * each primitive cell (i,j), with length a and      *
 * angle 120o, is formed by two equilateral triangles *
 *                                                    *
 * atoms A are in the vertices,                      *
 * atoms B are centered in the triangles below       *
 *                                                    *
 * the distance between atoms is the distance from   *
 * a vertex to the center of a triangle: a/sqrt(3)  *
*****/

for (j = 0; j < max; j++)
{
  for (i = -max; i < max; i++)
  {
/*****
 * A atoms (type = FALSE): (0, 0, 0) *
*****/

x = i * a_rotated[0] + j * b_rotated[0];
z = i * a_rotated[1] + j * b_rotated[1];
static_atom_3d (molecule, driver, max, i, j, FALSE,
n1, n2, perimeter, height, element1, x, z);

/*****
 * B atoms (type = TRUE): (2/3, 0, 1/3) *
*****/

x += 2 / 3.0 * a_rotated[0] + 1 / 3.0 * b_rotated[0];
z += 2 / 3.0 * a_rotated[1] + 1 / 3.0 * b_rotated[1];
static_atom_3d (molecule, driver, max, i, j, TRUE,
n1, n2, perimeter, height, element2, x, z);
}
}
}
```

Code Segment 1 - Definition of the graphene base vectors and creation of graphene atoms

3.1.2. Nanotube Base Vectors, Translation Vector and Height

Nanotubes are usually described using similar base vectors but separated by a 60° angle: $\mathbf{b}_1 = a \cdot \mathbf{x}$ and $\mathbf{b}_2 = a \cdot \sin(60^\circ) \cdot \mathbf{x} + a \cdot \cos(60^\circ) \cdot \mathbf{y}$. Clearly these are also lattice vectors because the lattice is hexagonal. Therefore any (m,n) integer combination of these two vectors represents a translation vector \mathbf{T} of the lattice, linking two equivalent points:

$$\mathbf{T} = m \cdot \mathbf{b}_1 + n \cdot \mathbf{b}_2 \quad (8)$$

```

/*****
 * nanotube lattice vectors *
 *****/

a[0] = bond * sqrt (3.0);
a[1] = 0.0;
b[0] = 0.5 * bond * sqrt (3.0);
b[1] = 1.5 * bond;

/*****
 * translation vector: T is always between *
 * 0 and PI/2, angle is set negative so T *
 * vector is rotated down, to be horizontal *
 * *
 * rectangle dimensions: perimeter x height *
 *****/

t[0] = n1 * a[0] + n2 * b[0];
t[1] = n1 * a[1] + n2 * b[1];

```

Code Segment 2 - Definition of nanotube base vectors and translation vector

Two lines perpendicular to \mathbf{T} passing through these two equivalent points, must also be equivalent, allowing us to define a rectangle with length $|\mathbf{T}|$ and equivalent sides of height h , as shown in Figure 3. As the two sides are equivalent, the rectangle can be folded and the two sides linked, forming a perfect seam, thus forming the final nanotube structure.

```

static double static_height (double bond, int rings, int n1, int n2,
double angle)
{
/*****
 * armchair (angle = 30): *
 * h = 2 * rings * bond * sin (PI / 3.0) *
 * *
 * zigzag (angle = 0): *
 * h = (1 + rings) * bond * sin (PI / 6.0) + rings * bond *
 *****/

return
bond * sin (GAMGI_MATH_PI / 6.0 - angle) +
rings * bond * sin (GAMGI_MATH_PI / 2.0 + angle) +
rings * bond * sin (GAMGI_MATH_PI / 6.0 + angle);
}

```

Code Segment 3 - Definition of height based on number of rings

3.1.3. Lattice Rotation

As mentioned in section 2.2.2, the integer coordinates (m,n) uniquely identify the whole atomic structure of a nanotube, including its radius and type (armchair, zig-zag, chiral). Depending of the nanotube type, the 2D rectangle makes an angle θ with the horizontal x axis (0° for zig-zag, 30° for armchair, between 0° and 30° for chiral nanotubes). To select the atoms within the 2D rectangle and then build the nanotube, it is easier to first rotate the rectangle in order to align it with the 2D x, y axes. This can be accomplished simply rotating the $\mathbf{a}_1, \mathbf{a}_2$ graphene base vectors of the angle $-\theta$, before generating the graphene atom coordinates (Figure 26). The atoms inside the transformed, horizontal, rectangle can now be selected simply by checking whether the x coordinate is between 0 and T (the length of the vector \mathbf{T}) and the y coordinate is between 0 and h . Atoms not fitting these conditions are discarded.

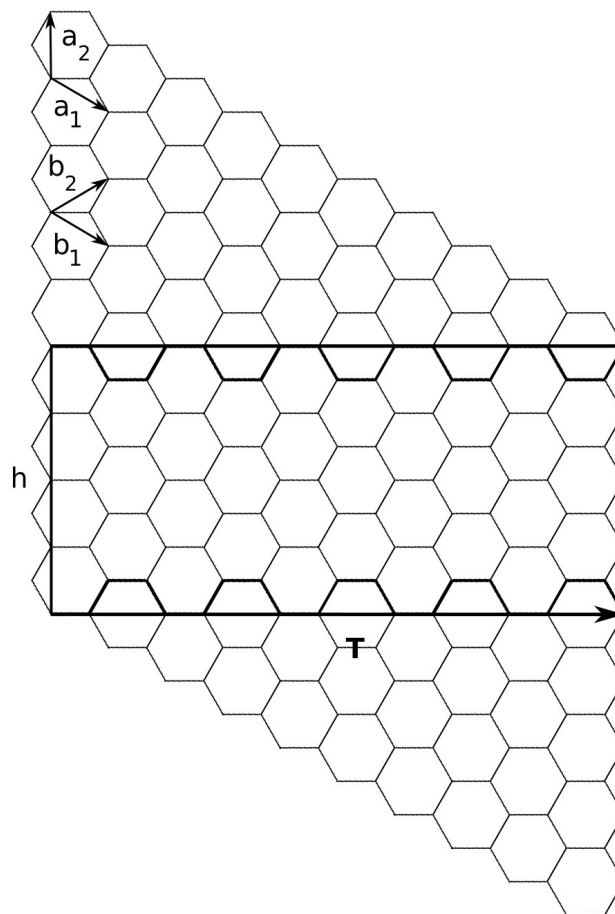


Figure 26 - Graphene lattice and selection rectangle of figure 3 after applying 30° clockwise rotation

```

static void static_vectors (double *a, double *b,
double angle, double *a_rotated, double *b_rotated)
{
/*****
* rotated graphene lattice vectors *
*****/

a_rotated[0] = cos (angle) * a[0] - sin (angle) * a[1];
a_rotated[1] = sin (angle) * a[0] + cos (angle) * a[1];
b_rotated[0] = cos (angle) * b[0] - sin (angle) * b[1];
b_rotated[1] = sin (angle) * b[0] + cos (angle) * b[1];
}

```

Code Segment 4 - Rotation of the graphene base vectors

3.1.4. Scanning Region

To find the range of graphene cells that must be considered, in order to generate all the relevant atoms, we notice that the farthest point in the rectangle (its top right corner) is given by:

$$r = \sqrt{T^2 + h^2} \quad (9)$$

This distance sets the radius of a circumference defining the maximum distance from the origin that must be considered. Moreover, as the rectangle is above the x axis, only atoms with positive y coordinates must be considered. By direct inspection (see Figure 27) it can be seen that the range of hexagonal graphene cells that must be considered to include all the relevant atoms closer than r can be given by $[-max, +max]$ (horizontal) and $[0, +max]$ (vertical) with:

$$max = 1 + (1.2 * r) / a \quad (10)$$

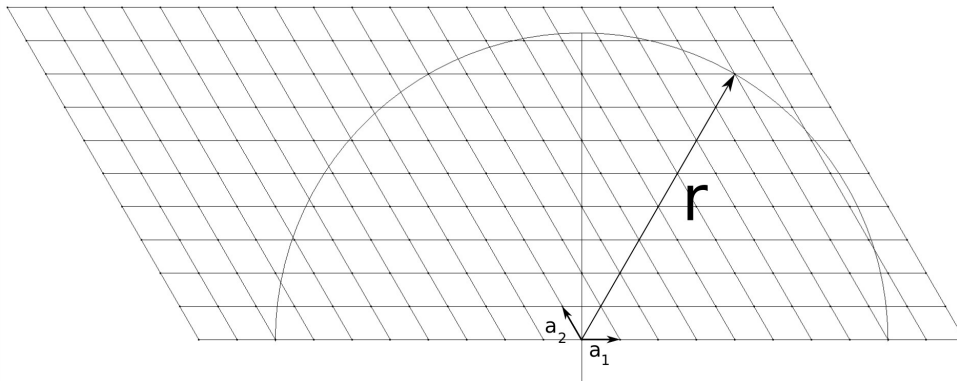


Figure 27 - The semi-circle area and lozenge of graphene cells for a nanotube. r is the rectangle diagonal

```

/*****
* scanning region: a hexagonal lozenge with *
* [-max,max] x [0, max] cells, covering *
* completely a half-circle with radius radius *
*****/

radius = sqrt (p * p + h * h);
*max = m = 1.0 + (1.2 * radius) / (bond * sqrt (3.0));

```

Code Segment 5 - Definition of half-circle radius and maximum number of cells

3.1.5. Coordinate Transformation

To convert the 2D rectangle of atoms into the 3D nanotube, we notice that the rectangle length T is simply the perimeter of the nanotube, so the nanotube radius (unrelated to the previous one) becomes $r = T / 2\pi$. The 3D atom coordinates are then obtained applying the following transformation (see Figure 28 and Figure 29) to all the atoms inside the 2D rectangle:

$$(x, y) = (r \cdot \cos(\alpha), r \cdot \sin(\alpha)) \quad (11)$$

with

$$\alpha = 2\pi \cdot x / T \quad (12)$$

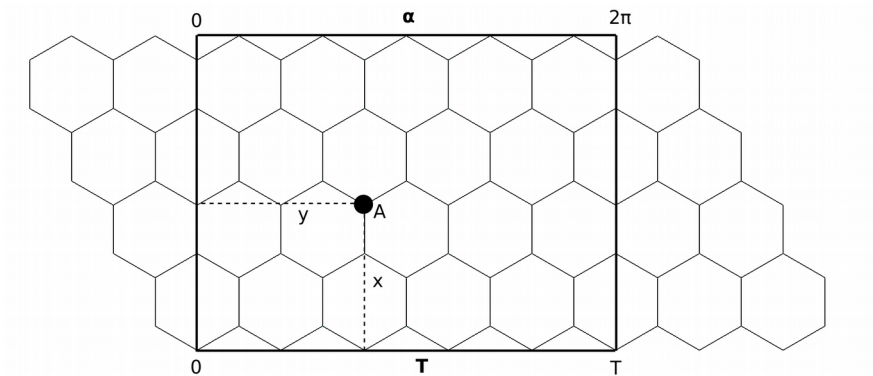


Figure 28 - Selection rectangle with length T . A is an atom with coordinates (x, y)

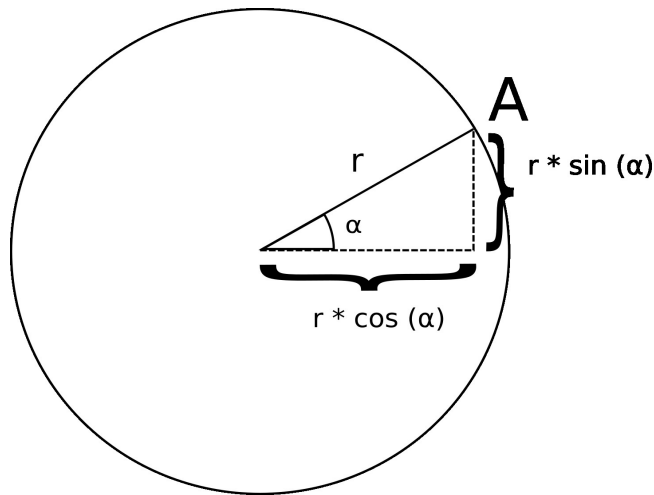


Figure 29 - Coordinate transformation for atom A of figure 29

```
static void static_fold (gamgi_atom *atom, double perimeter, double x, double z)
{
double radius, angle;

radius = perimeter / (2 * GAMGI_MATH_PI);
angle = x / radius;

atom->position[0] = radius * cos (angle);
atom->position[1] = radius * sin (angle);
atom->position[2] = z;
}
```

Code Segment 6 - Nanotube radius and angle definitions and coordinate transformation

3.1.6. Periodic Boundary Conditions

In particular, atoms with $x = T$ are automatically superimposed over atoms with $x = 0$ and must be discarded. This can be accomplished noticing that atoms in graphene cells (i,j) and $(i,j) + \mathbf{T}$ or $(i,j) - \mathbf{T}$ must be equivalent. Using the graphene cell vectors, the translation vector from equation (8) can be written as:

$$\mathbf{T} = (m+n) \cdot \mathbf{a}_1 + n \cdot \mathbf{a}_2 \quad (13)$$

so an atom in a graphene cell (i, j) must be discarded if another atom has been previously allocated in a lower cell $(i - m - n, j - n)$. (the argument could also be made for cells $(i + m + n, j + n)$, but these never occur because the cells are scanned upwards).

```
/******
 * ignore atoms related with periodic boundary atoms *
 * with -T vector, so boundary atoms (that would be *
 * superimposed after folding) are not created twice *
 * *
 * with +T vector is not needed because atoms are *
 * generated in increasing order, from -max to + max *
 *****/
```

```
i_pbc = i - n1 - n2;
```



```

j_pbc = j - n2;
pbc = TRUE;
if (j_pbc < 0 || i_pbc >= max) pbc = FALSE;
if (i_pbc < -max || i_pbc >= max) pbc = FALSE;
if (pbc == TRUE)
{
    number = gamgi_chem_graphene_index (2 * max, max, 0, i_pbc, j_pbc, type);
    if (driver[number] == NULL) pbc = FALSE;
}

/*****
 * create atom *
*****/

if (pbc == FALSE)
{
    atom = gamgi_chem_atom_create (element);
    gamgi_engine_link_atom_molecule (atom, molecule);

    static_fold (atom, perimeter, x, z - height / 2);
    number = gamgi_chem_graphene_index (2 * max, max, 0, i, j, type);
    driver[number] = atom;
}

```

Code Segment 7 - Creation of graphene atoms with discarding of atoms with $x = T$

3.1.7. Bond Creation

The final step in generating nanotubes is to create the bonds between the atoms. As mentioned before, each graphene cell contains two atoms, at positions (0, 0) and (2/3, 1/3), from now on called atoms A and B respectively. Through direct inspection, it can be seen that each atom A in a cell (i, j) should be connected to 3 atoms B at cells (i, j), (i - 1, j) and (i - 1, j - 1).

However, it must be recognized that atoms close to the left and right borders of the 2D rectangle may be bonded to atoms in the other border, when these two borders are connected, forming the seam that closes the 3D nanotube. Thus periodic boundary conditions must be applied, described by translation vectors $+T$ or $-T$, corresponding to (m + n, n) or (-m - n, -n) translation cells.

Therefore, the general procedure to create the nanotube bonds is: for each atom A in a cell (i, j), try to find an atom B in cells (i, j), (i - 1, j) and (i - 1, j - 1) (local atoms) or in cells (i + m + n, j + n), (i + m + n - 1, j + n) and (i + m + n - 1, j + n - 1) (atoms close to the left border: $\alpha \rightarrow 0$, applying $+T$ periodic boundary conditions) or in cells (i - m - n, j - n), (i - m - n - 1, j - n) and (i - m - n - 1, j - n - 1) (atoms close to the right border: $\alpha \rightarrow 2\pi$, applying $+T$ periodic boundary conditions). Every time an atom is found, a bond must be created between the two atoms, A and B. Figure 30 exemplifies these rules.

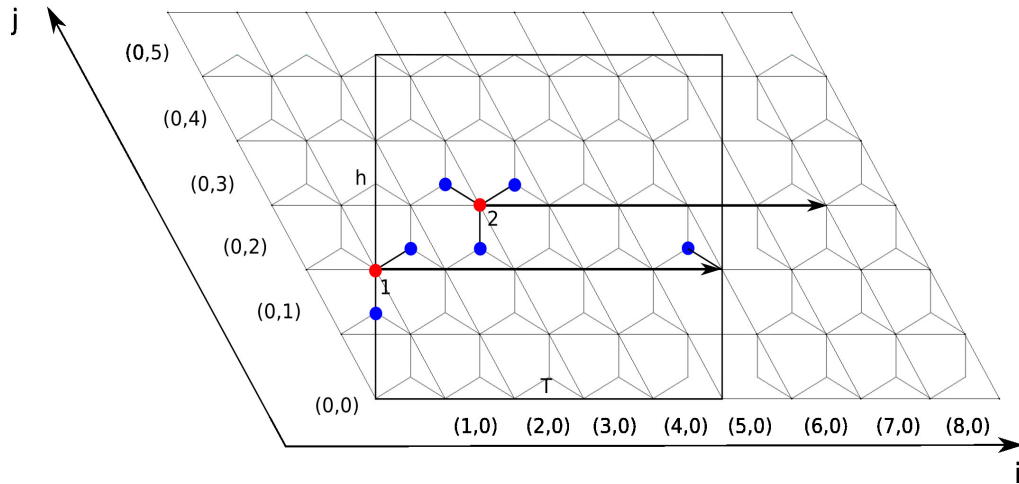


Figure 30 - Bonding scheme. Atoms 1 and 2 (in red) are A atoms. Atom 1 in cell (1,2) creates local bonds with atoms B (in blue) in cells (1,2) and (0,1), and a bond with atom B of cell (5,2), close to the right border. Atom B in cell (3,3) creates all possible local bonds, but no bonds with atoms close to the border.

```

static void static_bond_3d (gamgi_atom **driver, int max, int n1, int n2)
{
gamgi_atom *atom;
int i, j, number;

/*****
 * build bonds between atoms that become
 * neighbours after folding the graphene sheet
 *****/

for (j = 0; j < max; j++)
{
for (i = -max; i < max; i++)
{
/*****
 * type = FALSE for atoms A, type = TRUE for atoms B *
 *****/

number = gamgi_chem_graphene_index (2 * max, max, 0, i, j, FALSE);
atom = driver[number]; if (atom == NULL) continue;

/*****
 * periodic boundary bonds: -T vector *
 *****/

static_bond (atom, driver, max, i - n1 - n2, j - n2, TRUE);
static_bond (atom, driver, max, i - 1 - n1 - n2, j - n2, TRUE);
static_bond (atom, driver, max, i - 1 - n1 - n2, j - 1 - n2, TRUE);

/*****
 * periodic boundary bonds: +T vector *
 *****/

static_bond (atom, driver, max, i + n1 + n2, j + n2, TRUE);
static_bond (atom, driver, max, i - 1 + n1 + n2, j + n2, TRUE);
static_bond (atom, driver, max, i - 1 + n1 + n2, j - 1 + n2, TRUE);
}
}
}

```

Code 8 - Nanotube bond creation

3.1.8. Results

With the implementation of these algorithms, it is now possible to generate nanotubes using GAMGI. Examples of the three kinds of nanotubes that may now be generated are shown in figures 31 to 33. For each case, a top down and a side view are provided.

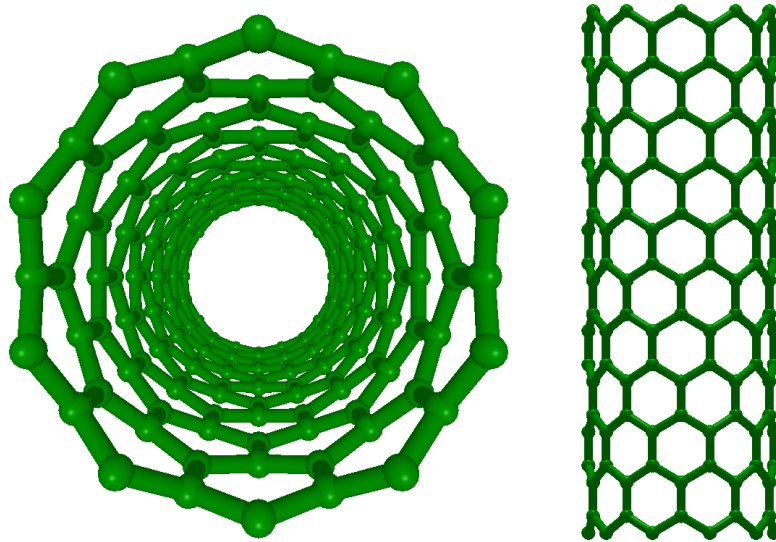


Figure 31 - Zig-zag (10,0) nanotube

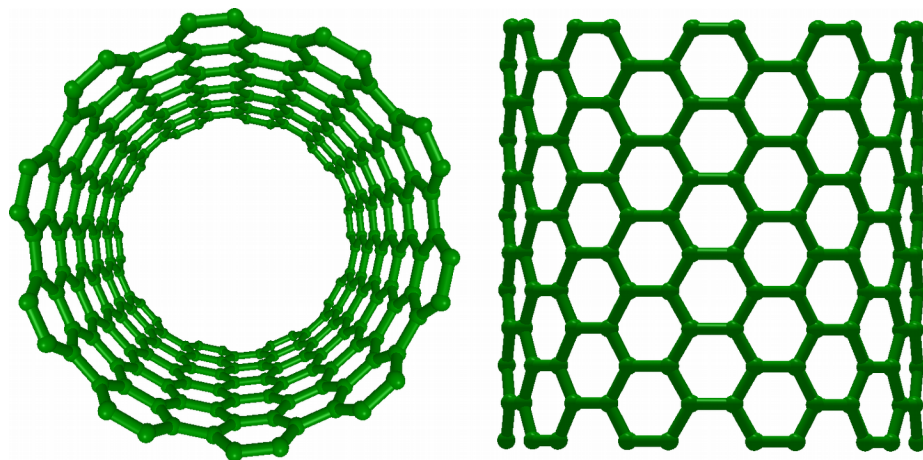


Figure 32 - Armchair (10,10) nanotube

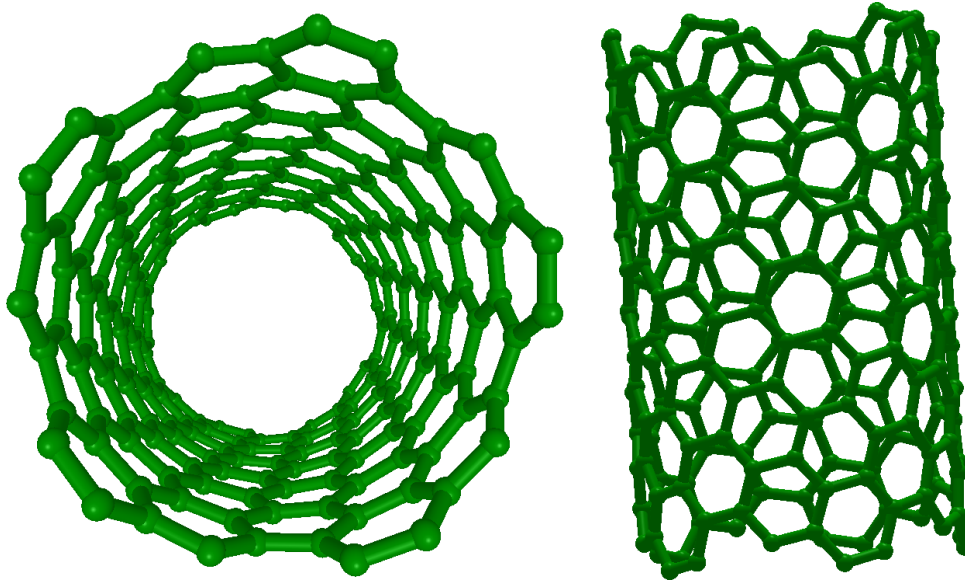


Figure 33 - Chiral (10,5) nanotube

Nanotubes with any combination of (m,n) indices can be generated without restrictions. The atoms are placed in the correct positions and there is barely any bond distortion.

3.2. Nanocones

3.2.1. General Considerations

As in the case of carbon nanotubes, a graphene sheet is used to create carbon nanocones. However, in the case of nanocones, we want the initial 2D sheet to be circular, in the simpler cases with the origin at the center of a hexagon instead of an atom, so the two carbon atoms per graphene cell are at positions $(2/3, 1/3)$ and $(1/3, 2/3)$. Nanocones can be described by a disclination angle, Δ , which is the angle of the circular slice, starting from the circle center, that is removed before folding the sheet until the two slice borders are reunited again, forming a 3D nanocone. As graphene has hexagonal symmetry, this angle Δ can have the value of 60° , 120° , 180° , 240° and 300° . Nanocones may be closed or open, in which case they lack the top part, or apex of the cone, so two radii are defined: the outer radius of the sheet (r_{out}) and the inner radius of the sheet (r_{in}). This allows us to define a circular region (an annulus) from which an angular slice having one of the five values just mentioned has been removed. Additionally, instead of removing a single slice with an angle Δ , it is also possible to remove up to five slices with angle Δ / n (usually 60°), where n is the number of slices, and in this case, the slices have their origin in hexagons close to the center of the cone. The sheet can then be folded by closing all the angular gaps that exist, resulting in the final nanocone structure.

3.2.2. Single Disclination Cones

3.2.2.1. Atom Selection

The apex angle and the shape at the top of the nanocone will be different for each disclination, as mentioned in section 2.2.2. As shown in Figure 34, to select the atoms within the circular region that is left after cutting the slice corresponding to the disclination angle, it is necessary to check if:

- their distance from the center is smaller than or equal to r_{out} and greater than or equal to r_{in} :

$$r_{out} \geq \sqrt{x^2 + y^2} \geq r_{in} \quad (14)$$

```
static double static_circle (double bond, int rings)
{
double h, v;

h = 0;
if (rings > 0)
    h = (rings - 1) * bond * cos (GAMGI_MATH_PI / 6.0);

v = 0;
if (rings > 0)
    v = bond / 2.0 + bond * sin (GAMGI_MATH_PI / 6.0) +
        (rings - 1) * (bond + bond * sin (GAMGI_MATH_PI / 6.0));

return sqrt (h * h + v * v);
}

/*****
 * remove inner and outer atoms *
*****/

d = sqrt (x * x + y * y);
if (d < r_in || d > r_out) return FALSE;
```

Code 9 - Nanocone radii definition and atom discarding

- the angle they make with the x axis is greater than the disclination angle:

$$d \leq \arccos\left(\frac{x}{\sqrt{x^2 + y^2}}\right) \quad (15)$$

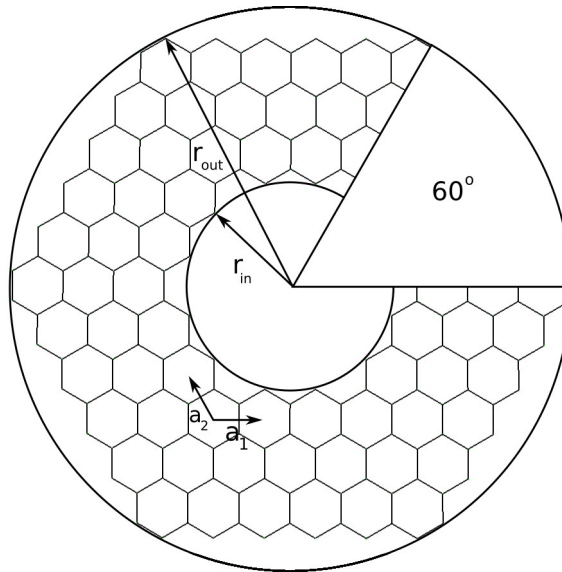


Figure 34 - The selection area for a nanocone is the circular crown with maximum and minimum radii r_{out} and r_{in} , minus the 60° disclination gap

3.2.2.2. Scanning region

An exercise similar to the case of nanotubes is applied to find the range of graphene cells that must be considered in order to generate all the relevant atoms. In this case, the nanocone outer radius, r_{out} , is used to set the radius of the circumference that defines the maximum distance from the origin that must be considered. Again, by direct inspection (see Figure 35), it can be seen that the range of hexagonal graphene cells that must be considered in order to include all the relevant atoms closer than r can be given by $[-max, +max]$ (horizontal) and $[-max, +max]$ (vertical) with:

$$max = 1 + (r_{out} * 1.2) / a \quad (16)$$

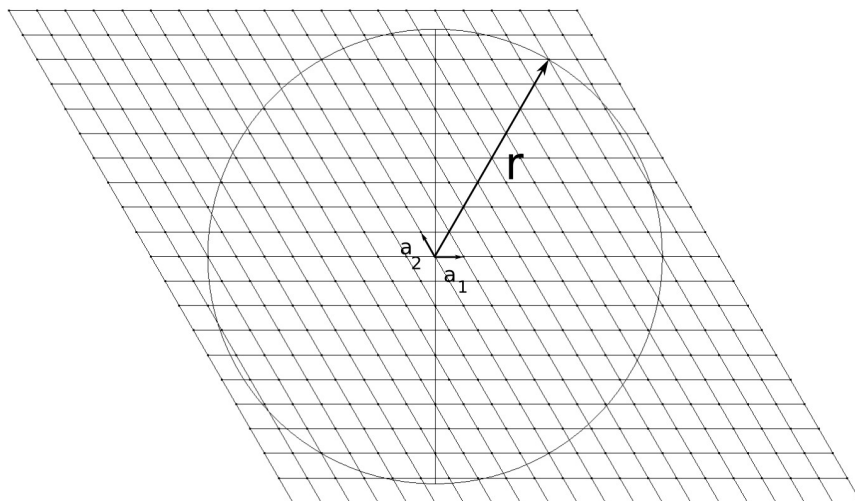


Figure 35 - The circle area and lozenge of graphene cells for a nanocone. r is the rectangle diagonal

3.2.2.3. Coordinate Transformation

To convert the 2D circular slice of atoms into the 3D nanocone, one side of the slice remains fixed, while the other rotates an angle equal to $-\Delta$, in order to fill the gap left by the removed slice. First, we note that atoms whose angle, α_{atom} , is equal to Δ should be rotated by $-\Delta$, that is, -100% of the value of Δ . Atoms whose angle is equal to 360° should remain fixed, that is, rotated 0% of the value of Δ . Atoms with an angle between Δ and 360° should be rotated between 0% and 100% of Δ , and the closer their angle is of Δ , the greater the rotation. The ratio that gives the percentage of the angle Δ that an atom should be rotated is given by: $2\pi - \alpha_{atom} / 2\pi - \Delta$. The value thus obtained is subtracted to 1 and the result multiplied by 2π in order to obtain the new value of the angle the atom will have after rotation. The formula used to calculate the value of the new angle α_{cone} of the atoms is:

$$\alpha_{cone} = 2\pi \left(1 - \frac{2\pi - \alpha_{atom}}{2\pi - \Delta}\right) \quad (17)$$

The radius of the base of the cone, r_{cone} , is given by:

$$r_{cone} = (2\pi - \Delta) r_{out} / 2\pi \quad (18)$$

while the height, h , of the nanocone can be calculated from Pythagoras' theorem (see Figure 36):

$$h = \sqrt{r_{out}^2 - r_{cone}^2} \quad (19)$$

When the graphene sheet is folded, the distance of the atoms from the axis of the cone (in the xy plane) is smaller than their original distance from the center of the sheet and the atoms are rotated by the angle α_{cone} . By multiplying the original distance to the center, $l = \sqrt{x^2 + y^2}$, by the ratio between the outer radius of the sheet and the base radius of the cone, $f = r_{cone} / r_{out}$, the new distance to the axis is obtained. The transformation of coordinates x and y is thus given by:

$$x = l \cdot f \cdot \cos(\alpha_{cone}) \quad (20)$$

$$y = l \cdot f \cdot \sin(\alpha_{cone}) \quad (21)$$

To obtain the z coordinate, we note that the top and bottom z coordinates of the cone should be $z = h / 2$ and $z = -h / 2$, respectively, and this is easily obtained by:

$$z = h \cdot (1 - l / r_{out}) - h / 2 \quad (22)$$

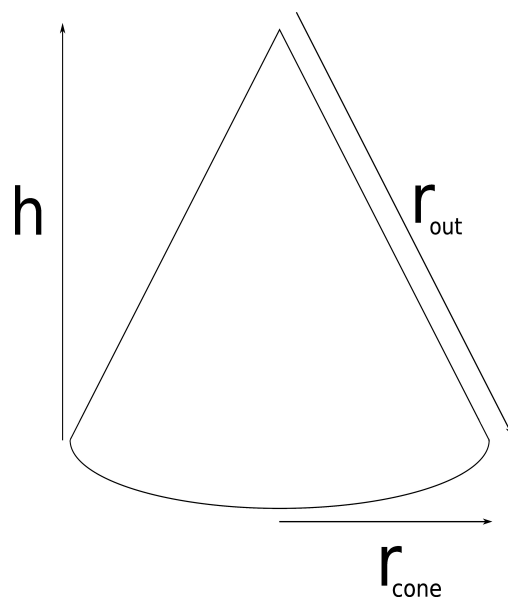


Figure 36 - Cone height, h , base radius, r_{cone} , and distance from apex to outer point of base, r_{out}

```

/*****
 * calculate cone radius and height *
 *****/

r_cone = static_cone (r_out, disclination);
f = r_cone / r_out; if (f > 1.0) f = 1.0;
height = r_out * sqrt (1 - f * f);

/*****
 * 1 ring, single rotation *
 *****/

angle = acos (x / d); if (y < 0) angle = 2 * GAMGI_MATH_PI - angle;
if (angle > disclination)
{
  angle = 2 * GAMGI_MATH_PI *
  (1 - (2 * GAMGI_MATH_PI - angle) / (2 * GAMGI_MATH_PI - disclination));
  *x_new = d * f * cos (angle);
  *y_new = d * f * sin (angle);
  *z_new = height * (1.0 - d * f / r_cone) - height / 2;
  return TRUE;
}

return FALSE;
}

```

Code Segment 10 - Cone radius and height definitions and coordinate transformation for single disclination

3.2.3. Multiple Disclinations

For several partial disclinations of 60° each, acting together, the hexagons at which they are centered will be transformed into pentagons, as one of the six atoms of the original hexagon will be removed. Depending on the center of the disclinations, different isomers of a nanocone with a specific total disclination may be generated, since the pentagons will be located at different places of the

structure. Moreover, rotating only one end or both ends of the angular gaps left by disclinations produce slightly different results. Therefore, several different isomers can be created for the same total disclination.

3.2.3.1. Origin Translation and Radii

The general procedure to model nanocones is direct when a single disclination is applied, becoming progressively more complex when more disclinations are introduced. First, the origin of the nanocone may be moved to the center of a bond, if the nanocone model so requires. Then, atoms with a distance from the center greater than r_{out} and smaller than r_{in} are removed. The next step is applied to every disclination in succession.

```

/*****
 * change global origin *
 *****/

x += o_x;
y += o_y;

/*****
 * remove inner and outer atoms *
 *****/

d = sqrt (x * x + y * y);
if (d < r_in || d > r_out) return -1;

```

Code Segment 11 - Nanocone origin translation and atom discarding

3.2.3.2. Origin Conversion and Angle

The center of each disclination becomes the new rotation origin, and the atom coordinates are converted so that they become relative to the new origin, that is, $(x, y) \rightarrow (x - c_x, y - c_y)$, where c_x and c_y depend on the disclination center's coordinates. The angle that the atoms make with the new horizontal x axis is calculated through the formula:

$$\alpha_c = \arccos\left(\frac{x_c}{\sqrt{x_c^2 + y_c^2}}\right) \quad (23)$$

```

/*****
 * get cartesian and cylindrical new atom *
 * coordinates, from the rotation center *
 *****/

x_c = x - c_x;
y_c = y - c_y;
d_c = sqrt (x_c * x_c + y_c * y_c);
angle = acos (x_c / d_c);
if (y_c < 0) angle = 2 * GAMGI_MATH_PI - angle;

```

Code Segment 12 - Origin conversion and angle calculation

3.2.3.3. Atom Removal from Disclination Gaps

To check whether an atom is inside the disclination area, we start by noting that a bisector can be defined for every disclination, which is separated by the same angle from both ends of the disclination, and we call this angle *min*. Let a_0 and a_1 be the minimum and maximum angles of the disclination, given by: $a_0 = \text{bisector} - \text{min}$ and $a_1 = \text{bisector} + \text{min}$. Figure 37 illustrates these concepts.

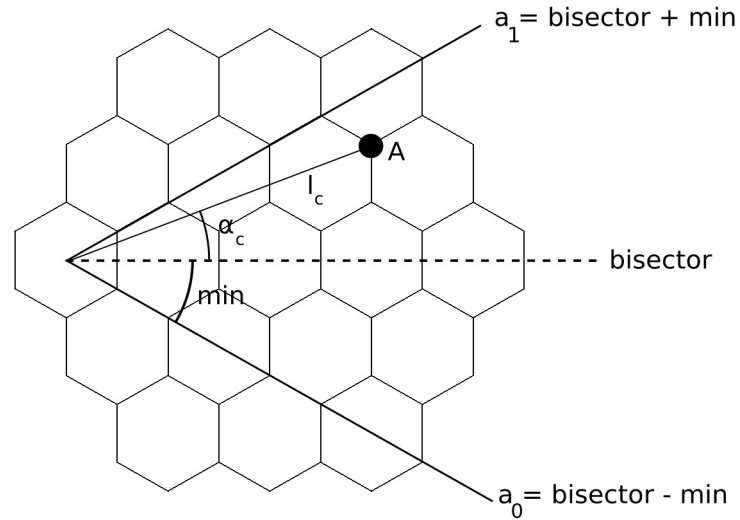


Figure 37 - Bisector, a_1 and a_2 for a generic disclination. A is an atom at a distance l_c from the disclination center, making an angle α_c with the horizontal direction

Let t_0 and t_1 be the perpendicular distance of l_c relative to a_0 and a_1 , respectively (see Figure 38):

$$t_0 = l_c \cdot \sin(\alpha_c - a_0) \quad (24)$$

$$t_1 = l_c \cdot \sin(\alpha_c - a_1) \quad (25)$$

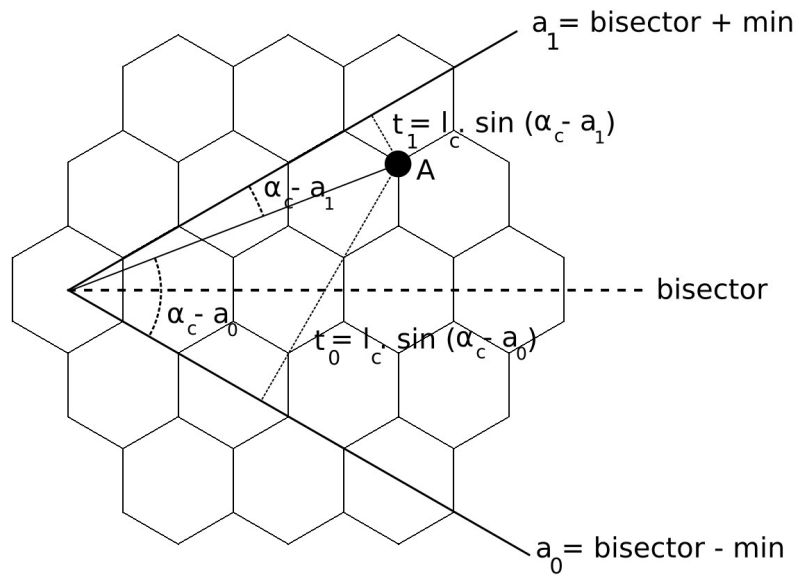


Figure 38 - t_0 and t_1 for atom A. Note that t_1 is actually negative.

It is possible to verify that $t_0 > 0$ and $t_1 < 0$ for atoms inside the disclination area, so that the criterion $t_0 > 0 > t_1$ is verified, then it is inside the disclination area, so its angle is: $a_1 \geq \alpha_c \geq a_0$ and it must be removed. However, when there are atoms along the angles a_0 and a_1 , only atoms with $\alpha_c = a_0$ should be included, whilst removing atoms with $\alpha_c = a_1$, or vice-versa. In order to achieve this, the selection criterion is modified in the following way:

- $t_0 > 0.1 > t_1$ - in case atoms with $\alpha_c = a_1$ should be removed;
- $t_0 > -0.1 > t_1$ - in case atoms with $\alpha_c = a_0$ should be removed.

0.1 is simply a tolerance value which is found to fit well its purpose. It is multiplied by a value, called *side*, which is 1 for the first case and -1 for the second case. Figure 39 illustrates this concept.

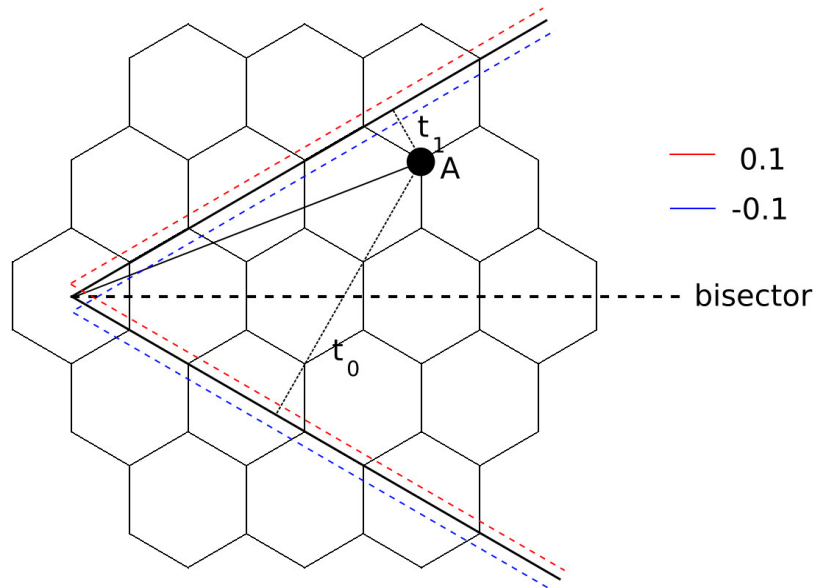


Figure 39 - New disclination borders for the cases where atoms at a_0 are kept (red dashed lines) and atoms at a_1 are kept (blue dashed lines)

```

/*****
 * remove atoms in negative, positive or *
 * both arcs from bisector to min angles *
 *****/

a0 = bisector - min;
a1 = bisector + min;
t0 = d_c * sin (angle - a0);
t1 = d_c * sin (angle - a1);
if (t0 > side * TOLERANCE_T && t1 < side * TOLERANCE_T) return -1;

```

Code Segment 13 - Definition of a_0 , a_1 , t_0 and t_1 and atom discarding from angular gap in the case of multiple disclinations

3.2.3.4. Rotation Check

The last step consists in determining whether an atom should be rotated in order to fill the gap left by the disclination. Either atoms from both sides of the disclinations are rotated and joined in the middle (Figure 40), or one side remains fixed while the other is rotated. Since there is more than one disclination, it is not possible to rotate all the atoms of the structure, so an angle max , relative to the bisector, is defined for each nanocone model, such that atoms with $\alpha_c > max$ ou $\alpha_c < -max$ aren't rotated.

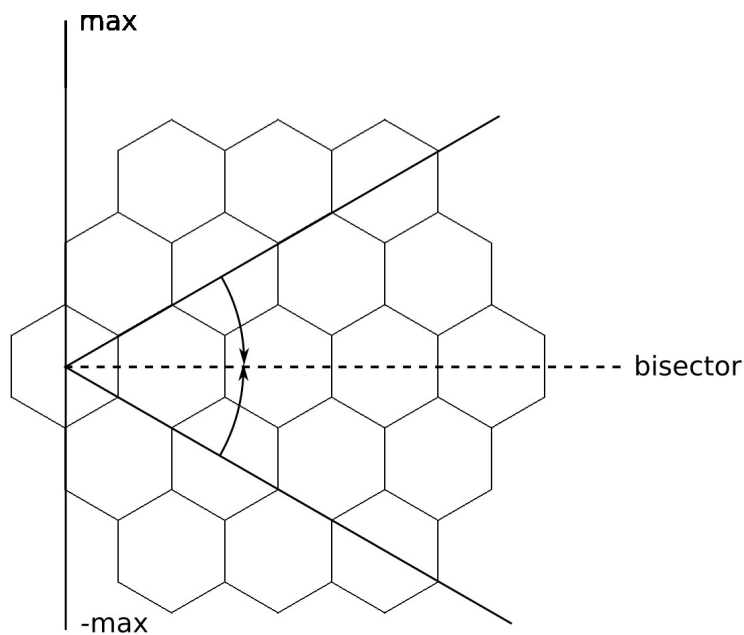


Figure 40 - Disclination closed by rotating both edges. Atoms between $-max$ and a_0 and atoms between max and a_1 are rotated

In case only one side is to be rotated, it is necessary to choose which side, between a_0 and a_1 is to remain fixed and which is to be rotated. If a_0 is to be fixed (Figure 41), only atoms separated less than 90° from a_0 are rotated. If a_1 is to be fixed (Figure 42), only atoms separated more than -90° are rotated.

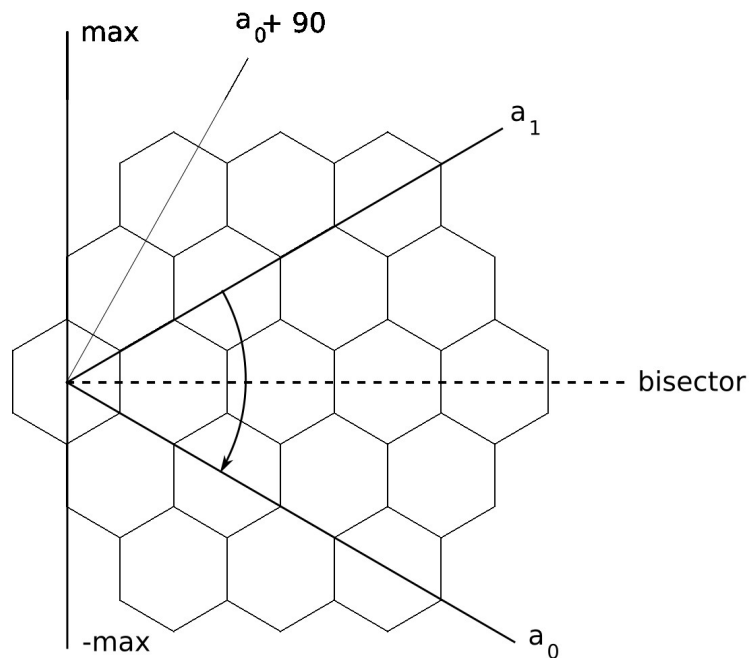


Figure 41 - Disclination closed by rotating one edge. Atoms at a_0 remain fixed. Atoms between $a_0 + 90^\circ$ and a_1 are rotated.

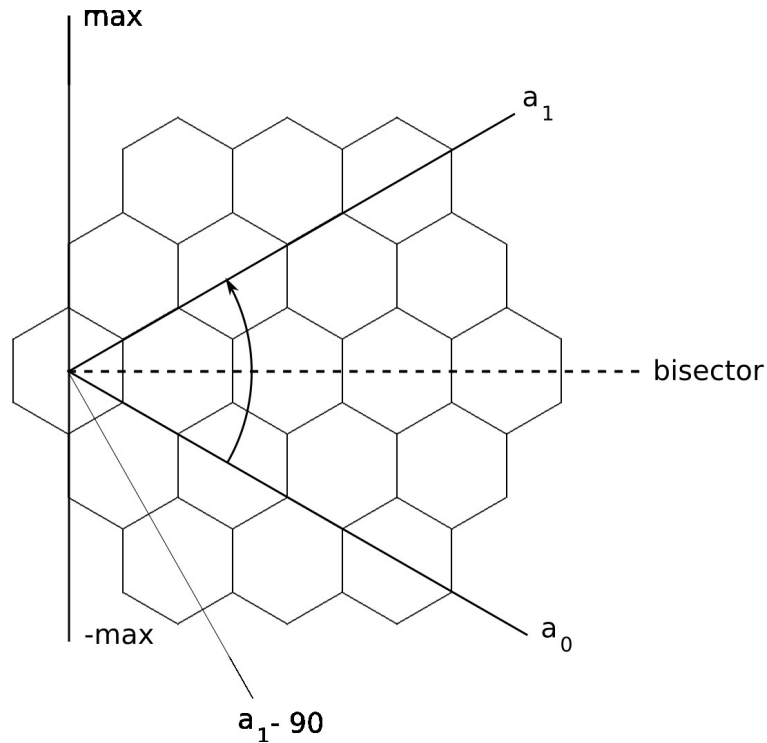


Figure 42 - Disclination closed by rotating one edge. Atoms at a_1 remain fixed. Atoms between $a_1 - 90^\circ$ and a_1 are rotated.

In case an atom is rotated, the new angle that he should have, α_{new} , is calculated. It will be rotated of at most 30° in case both sides are rotated, and 60° in case one side is fixed.

$$\alpha_{new} = \alpha_c - \min(max - \alpha_c / max - min) \quad (26)$$

```

if (positive == FALSE && negative == FALSE) return 0;
if (min == max) return 0;

if (positive == TRUE && negative == FALSE)
  { bisector -= min; min *= 2; }

if (positive == FALSE && negative == TRUE)
  { bisector += min; min *= 2; }

/*****
 * convert angle to local values: [-max to +max], bisector = 0 *
 *****/

angle = angle - bisector;
if (positive == TRUE && negative == TRUE)
  {
  if (angle > GAMGI_MATH_PI) angle -= 2 * GAMGI_MATH_PI;
  if (angle < -GAMGI_MATH_PI) angle += 2 * GAMGI_MATH_PI;
  }

/*****
 * return atom if outside the rotation zone *
 *****/

if (fabs (angle) < min / 2) return 0;
if (fabs (angle) > max) return 0;

```

```

if (negative == FALSE && angle < 0) return 0;
if (positive == FALSE && angle > 0) return 0;

/*****
 * scale angle from [max, min] to [max, 0] *
 *****/

if (angle < 0) {min = -min; max = -max; }
angle -= min * (max - angle)/ (max - min);

/*****
 * convert angle back to absolute values: [0, 360[ *
 *****/

angle += bisector;
if (angle < 0) angle += 2 * GAMGI_MATH_PI;

```

Code Segment 14 - Multiple disclination rotation process. First, check what edges are to be rotated and adjust the values of *bisector* and *min* accordingly. Second, convert angle to local values relative to *bisector*, with bisector = 0°. Third, return atom coordinates if they lie outside the rotation zone. Fourth, get new angle for atoms. Finally, convert angle back to absolute values.

3.2.3.5. Coordinate Transformation

The 3D nanocone coordinates may finally be calculated by following an identical logic to the transformations for a single disclination, with the exception that the *x* and *y* coordinates of the atom should be reconverted back so that they become relative to the center of the cone. The transformation of coordinate *z* is the same as in equation 22.

$$x = l \cdot f \cdot \cos(\alpha_{new}) + f \cdot c_x \quad (27)$$

$$y = l \cdot f \cdot \sin(\alpha_{new}) + f \cdot c_y \quad (28)$$

```

/*****
 * rotate atom with new angle to close arc gap *
 *****/

*x_new = f * d_c * cos (angle) + f * c_x;
*y_new = f * d_c * sin (angle) + f * c_y;

return 1;

```

Code Segment 15 - Multiple disclination *x* and *y* coordinate conversion for atoms that are to be rotated

If an atom is not rotated to be rotated, the coordinate conversion is made so that the atom is pulled towards the center, and its height changed:

$$x_{new} = x \cdot f \quad (29)$$

$$y_{new} = y \cdot f \quad (30)$$

```

/*****
* set default atoms *
*****/

*x_new = f * x;
*y_new = f * y;
*z_new = height * (1.0 - d * f / r_cone) - height / 2;

```

Code Segment 16 - Multiple disclination x, y and z coordinate conversion for atoms which are not to be rotated

3.2.3.6. Analysis of Nanocone 120_E

For a better understanding, let us examine the case of the nanocone model that we have labeled as *120_E*, a nanocone with a total disclination of 120° in which the two pentagons (resulting from the application of the two 60° disclinations) are separated by an atomic bond (edge), and the gaps are closed by rotating both sides (Figure 43).

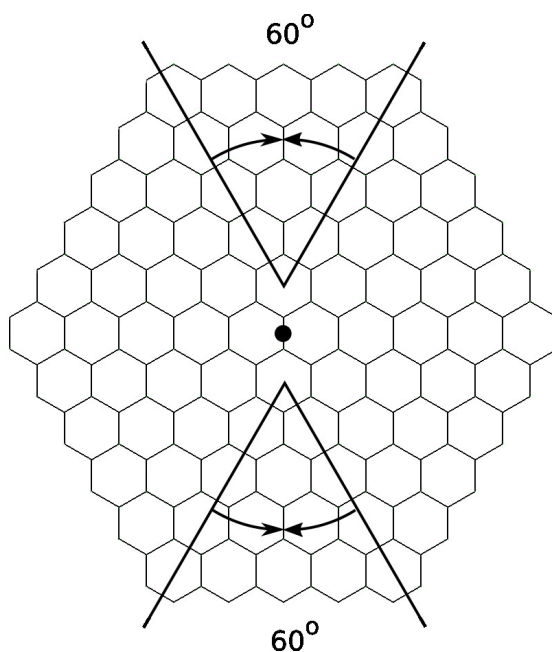


Figure 43 - Schematic for the creation of isomer *120_E*. The origin of the cone is marked by a black dot. Two 60° disclinations are applied, and both of them are closed by rotating both sides.

First, the origin is changed so that it is in the center of a bond, and this is done simply by adding $a/2$ to the x position of the origin. The following steps are carried out first for the top disclination center and repeated for the bottom one (we describe the procedure only for the top one). The coordinate transformation $(x, y) \rightarrow (x_c, y_c)$ is applied, with $x_c = 0$ and $y_c = \mathbf{a}_2$ (graphene lattice base vector) and the angle α_c is calculated. The bisector is located at 90° , with $\min = 30^\circ$, so $a_0 = 60^\circ$ and $a_1 = 120^\circ$. Atoms with $60^\circ < \alpha_c \leq 120^\circ$ are removed, so the criterion $t_0 > 0.1 > t_1$ is used, with $\text{side} = 1$. The value of \max is 90° , so only atoms from 0° to 180° should be rotated. Since both sides of the disclination are rotated, they meet in the mid point, 90° . Atoms with $\alpha_c = 60^\circ$ are to be rotated of *min*

so that they are placed at 90° . Atoms with $\alpha_c = 0^\circ$ or $\alpha_c = 180^\circ$ remain still. Atoms with $0 < \alpha_c < 60^\circ$ are to be rotated of an angle between 0° and min , and atoms with $120^\circ < \alpha_c < 180^\circ$ are to be rotated of an angle between 0° and $-min$.

3.2.3.7. Bond Creation

To create bonds between atoms, a scheme similar to the one used for nanotubes can be applied although each nanocone requires a slightly different bonding scheme. Upon analyzing each individual case, bonding patterns become clear, so it is possible to establish a common basis.

Considering the initial indices i and j of a 2D graphene sheet, each bond in a seam region (the region where the two loose edges come together through the bonding of the atoms) is always established between an A or B atom of an (i_0, j_0) cell on one side of the seam, and another A or B atom in an (i_1, j_1) cell on the other side of the seam. Also, if the atom on cell (i_0, j_0) is an atom A, then $k_0 = 0$, otherwise $k_0 = 1$. Identically, if the atom on cell (i_1, j_1) is an atom A, then $k_1 = 0$, otherwise $k_1 = 1$.

It is necessary to determine the bond closer to the center of the nanocone disclination through direct inspection. From this inspection, i_0, j_0, k_0, i_1, j_1 and k_1 are determined. Since the disclination area that is subtracted corresponds to a symmetry rotation, there is a natural, predictable progression of these indices. This progression is identified by six other indices: $ii_0, jj_0, kk_0, ii_1, jj_1,$ and kk_1 , which are used in the algorithm: 0 when the index remains constant, +1 when it increases by 1, -1 when it decreases by -1, +2 when it increases by 1 every 2 iterations and the resulting atom is of type A, -2 when it decreases by 1 every 2 iterations and the resulting atom is of type A, +3 when it increases by 1 every 2 iterations and the resulting atom is of type B, -3 when it decreases by 1 every 2 iterations and the resulting atom is of type B (kk_0 and kk_1 are either 0 or 1, depending on whether the atom type changes or not).

```
static void static_link (gamgi_atom **driver, int max,
int i0, int j0, int k0, int i1, int j1, int k1,
int ii0, int jj0, int kk0, int ii1, int jj1, int kk1)
{
gamgi_atom *atom;
int number;

while (i0 < max - 1 && j0 < max - 1 && i0 > -max && j0 > -max)
{
/*****
* update atom 0 *
*****/

i0 = static_update (i0, ii0, k0);
j0 = static_update (j0, jj0, k0);
k0 = abs (kk0 - k0);

/*****
* update atom 1 *
*****/

i1 = static_update (i1, ii1, k1);
j1 = static_update (j1, jj1, k1);
k1 = abs (kk1 - k1);
```

```

/*****
 * create bond between atoms (i0,j0,k0) and (i1,j1,k1) *
 *****/

number = static_index (i0, j0, max) + k0;
if (number >= 0 && number < 8 * max * max)
{
  atom = driver[number]; if (atom != NULL)
  static_bond (atom, driver, max, i1, j1, k1);
}
}
}

```

Code Segment 17 - Nanocone bond creation

3.2.3.8. Analysis of Nanocone 60_O

Let us examine the bond creation process of the nanocone model that was labeled as 60_O, a nanocone with a total disclination of 60° in which the pentagon is located at the top and center of the nanocone, and the gaps are closed by rotating one side (Figure 44). The origin of the nanocone is marked with a black dot.

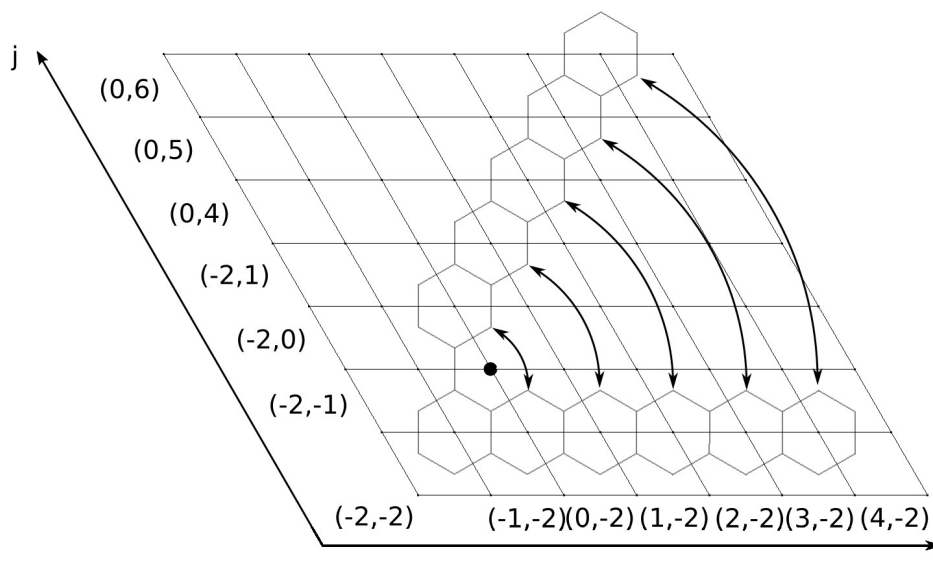


Figure 44 - Bonding scheme for the 60_o nanocone isomer

- The first bond to be created is between atom B of cell (0, -1), and atom B of cell (0, 0). The starting indices are now known;
- The second bond is between B atoms of cells (1, -1) and (1, 1);
- The third bond is between B atoms of cells (2, -1) and (2, 2);
- The fourth bond is between the B atoms of cells (3, -1) and (3, 3);

The pattern has now been established.

- The atom is always a B atom, in both cases, so $k_0 = k_1 = 0$ always. Because there is no change in these indices, $kk_0 = kk_1 = 0$;
- The value of 1 is added to i_0 for each bond, while $j_0 = -1$ always. As such, $ii_0 = 1$, while $jj_0 = 0$;
- The value of 1 is added to both i_1 and j_1 for each bond. As such, $ii_1 = jj_1 = 1$.

This analysis was repeated on every nanocone. A table with all the indices can be seen in the appendix B.

3.2.4. Results

With the implementation of these algorithms, it is now possible to generate nanocones using GAMGI. Figures 45 to 52 show the most relevant examples of nanocones. In the cases where more than one disclination was introduced, the pentagons are colored red. Nanocones are identified by a number, which is equal to the total disclination, and a letter, which either regards the way in which gaps are closed or the the way pentagons are linked. Nanocones labeled “edge” or “hexagon” have their pentagons linked by edges (bonds) or hexagons, and “double” if the gaps are closed by rotating both sides. Nanocones with a single disclination are identified by the letter O, meaning that the disclination is centered in the nanocone origin.

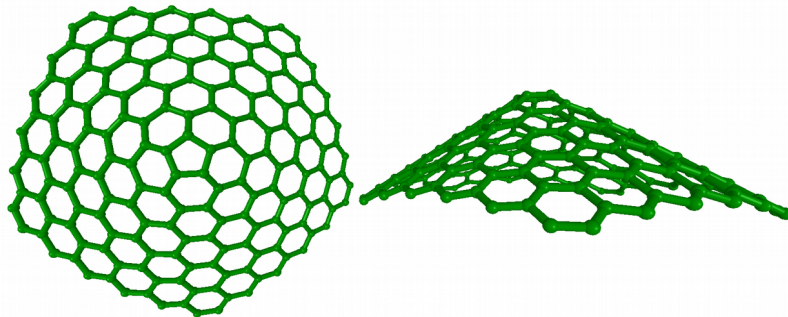


Figure 45 - Nanocone 60_O

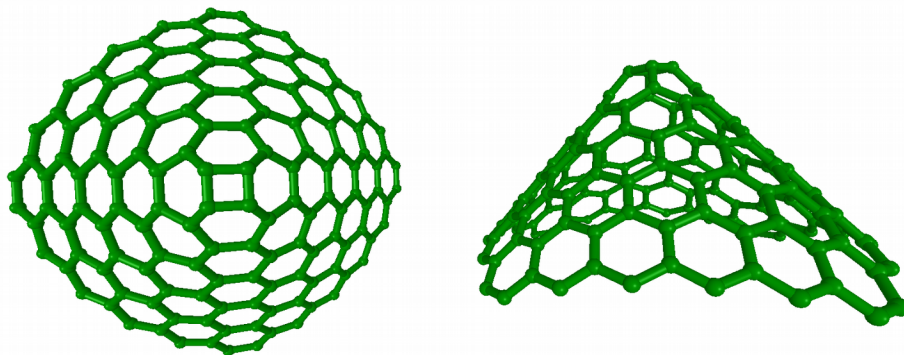


Figure 46 - Nanocone 120_O

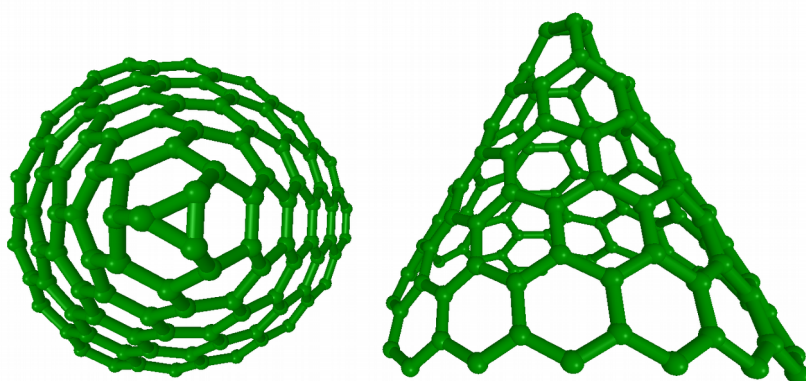


Figure 47 - Nanocone 180_O

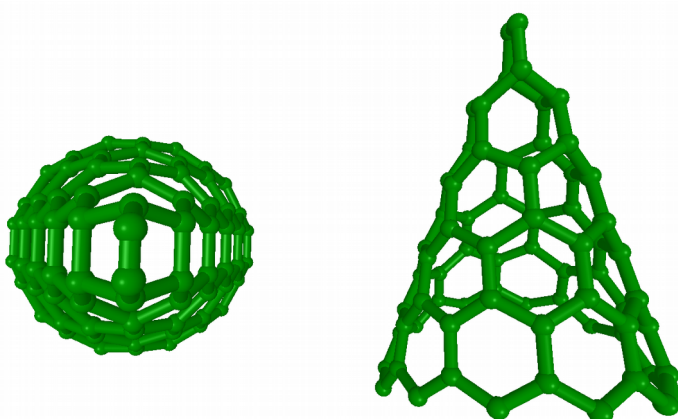


Figure 48 - Nanocone 240_O

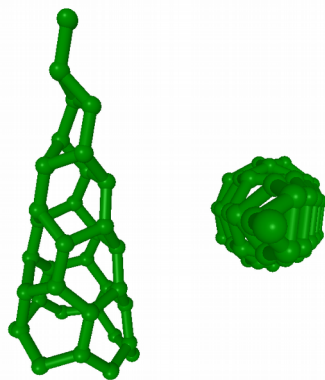
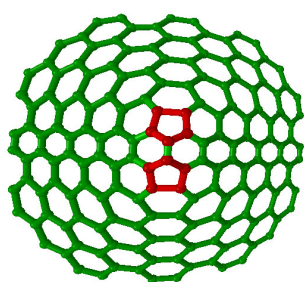
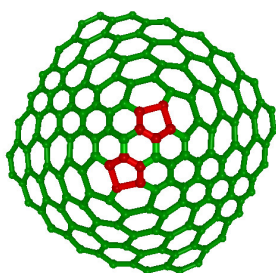


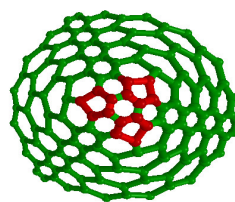
Figure 49 - Nanocone 300_O



120 Edge

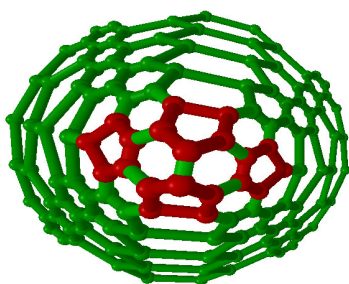


120 Hexagon

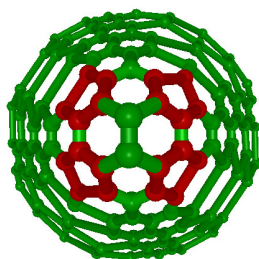


180 Double

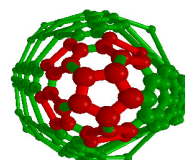
Figure 50 - Nanocones 120_E, 120_H and 180_D



240 Edge



240 Hexagon



300 Edge

Figure 51 - Nanocones 240_E, 240_H and 300_E

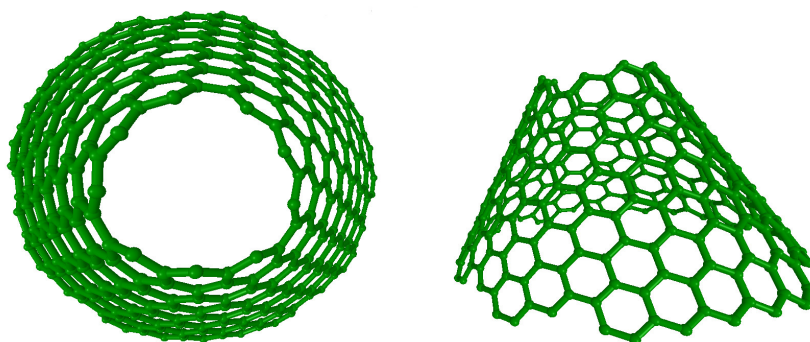


Figure 52 - Open nanocone 180_O

It is now possible to generate any nanocone with a single disclination with a discrete value between 60° and 300° . The higher the disclination value, the more distorted the structure becomes, but it is still clearly defined.

A total of 10 different models of nanocones with multiple disclinations can be generated. Just like in the case of single disclinations, the structures become more distorted with the increase of the disclination (which corresponds to introducing more disclinations). However, since not all the atoms of the structure are rotated, and those that are become more dislocated from their original position, the structures are more heavily distorted in the regions where the disclination gaps were removed. Thus, at this moment, the generation of nanocones with multiple disclinations may produce heavily distorted structures, of which the most striking example is nanocone 300_E.

3.3. Fullerenes

3.3.1.1. L_0 Transformation

As mentioned in section 2.4.6.4, according to Hasheminezhad et al. [63], all fullerene structures can be constructed applying only three classes of atomic expansions: L_i , $B_{i,j}$ and F , except C_{28} (which needs a special transformation, discussed later). As stated in algorithm 3, expansions L_i alone suffice to build fullerene isomers up to 300 atoms. For example, by applying expansion L_0 , which adds 2 faces and 4 atoms to a fullerene, C_{24} is obtained. The first step is to identify and select the atoms that participate in an L_0 expansion, as seen in Figure 53.

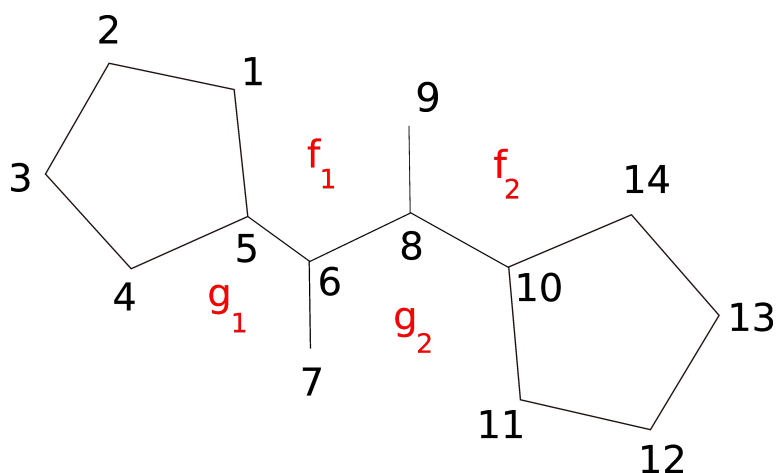


Figure 53 - Atoms that participate in transformation L_0

There are 14 atoms. Two of the faces must be pentagons. Faces labeled g_1 , g_2 , f_1 and f_2 may be either pentagons or hexagons.

```
int l0 (double *atom_in, int *bond_in, int size,
double *atom_out, int *bond_out, int central_1, int *hexagon)
{
double x, y, z;
int central_1n[3];
int central_2, central_2n[3];
int single_1, single_1n[3];
int single_2, single_2n[3];
int junction_1, junction_1n[3];
int junction_2, junction_2n[3];
int near_1, near_1n[3];
int near_2, near_2n[3];
int far_1, far_1n[3];
int far_2, far_2n[3];
int hexagon_1, hexagon_2;
int edge_1, edge_2;

copy_matrix_bond (bond_in, size, bond_out, size + 4);
copy_matrix_atom (atom_in, size, atom_out, size + 4);

/*****
 * identify L0 changing atoms for side 1 *
 *****/

/*****
 * get neighbours of central 1: *
 * position 0 = central_2      *
 * position 1 = single_1      *
 * position 2 = junction_1    *
 *****/

neighbour_find (bond_in, size, central_1, -1, central_1n);
central_2 = central_1n[0];
single_1 = central_1n[1];
junction_1 = central_1n[2];

/*****
 * get neighbours of junction_1: *
 * position 0 = central_1      *
 * position 1 = near_1        *
 * position 2 = far_1         *
 *****/
```

```

neighbour_find (bond_in, size, junction_1, central_1, junction_1n);
if (ring_check (bond_in, size, single_1, junction_1n[1]) == FALSE)
    neighbour_swap (junction_1n, 1, 2);

near_1 = junction_1n[1];
far_1 = junction_1n[2];

/*****
 * identify L0 changing atoms for side 2 *
*****/

/*****
 * get neighbours of central_2: *
 * position 0 = central_1      *
 * position 1 = single_2       *
 * position 2 = junction_2     *
*****/

neighbour_find (bond_in, size, central_2, central_1, central_2n);
if (ring_check (bond_in, size, single_1, central_2n[1]) == TRUE)
    neighbour_swap (central_2n, 1, 2);

single_2 = central_2n[1];
junction_2 = central_2n[2];

/*****
 * get neighbours of junction_2: *
 * position 0 = central_2       *
 * position 1 = near_2         *
 * position 2 = far_2          *
*****/

neighbour_find (bond_in, size, junction_2, central_2, junction_2n);
if (ring_check (bond_in, size, single_2, junction_2n[1]) == FALSE)
    neighbour_swap (junction_2n, 1, 2);

near_2 = junction_2n[1];
far_2 = junction_2n[2];

/*****
 * check that junction_1, near_1, far_1 are part of a pentagon *
*****/

neighbour_find (bond_in, size, near_1, junction_1, near_1n);
neighbour_find (bond_in, size, far_1, junction_1, far_1n);
if (neighbour_check (bond_in, size, near_1n, far_1n, 1) == FALSE)
    return FALSE;

/*****
 * check that junction_2, near_2, far_2 are part of a pentagon *
*****/

neighbour_find (bond_in, size, near_2, junction_2, near_2n);
neighbour_find (bond_in, size, far_2, junction_2, far_2n);
if (neighbour_check (bond_in, size, near_2n, far_2n, 1) == FALSE)
    return FALSE;

```

Code Segment 18 - L_0 transformation atom selection

Expansion L_0 essentially consists in adding four atoms in a way that leads to the creation of two adjacent pentagons, while transforming the original pentagons into hexagons, as can be seen in Figure 54.

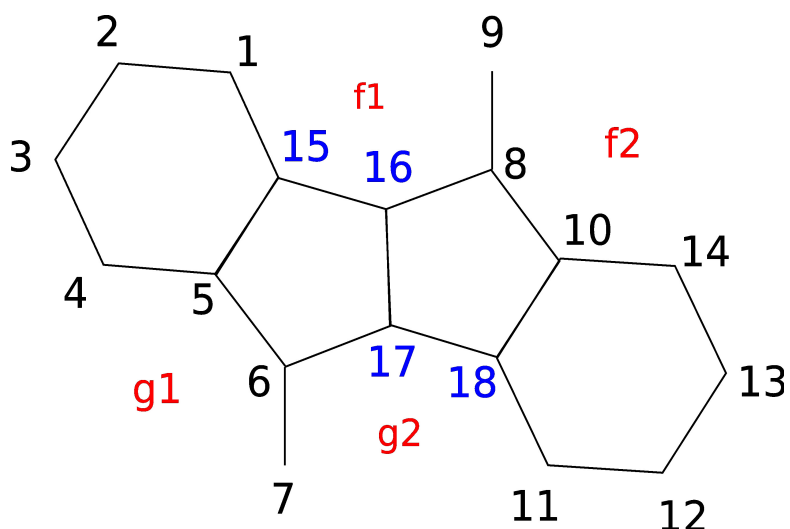


Figure 54 - Atoms from figure 52 after applying transformation L_0 . The added atoms are numbered in blue.

By analyzing both figures thoroughly, the necessary steps to carry out the transformation become evident:

- break the bonds between atoms:
 - 1 – 5
 - 6 – 8
 - 10 – 11
- create bonds between atoms:
 - 1 – 15
 - 5 – 15
 - 6 – 17
 - 8 – 16
 - 10 – 18
 - 11 – 18
 - 15 – 16
 - 16 – 17
 - 17 – 18

```

/*****
* remove 3 old bonds *
*****/

```

```

bond_create (bond_out, size, central_1, central_2, 0);
bond_create (bond_out, size, junction_1, far_1, 0);
bond_create (bond_out, size, junction_2, far_2, 0);

```

```

/*****
 * create 9 new bonds *
 *****/

bond_create (bond_out, size, hexagon_1, junction_1, 3);
bond_create (bond_out, size, hexagon_1, far_1, 3);
bond_create (bond_out, size, hexagon_2, junction_2, 3);
bond_create (bond_out, size, hexagon_2, far_2, 3);

bond_create (bond_out, size, hexagon_1, edge_1, 3);
bond_create (bond_out, size, hexagon_2, edge_2, 3);
bond_create (bond_out, size, edge_1, edge_2, 3);
bond_create (bond_out, size, edge_1, central_2, 3);
bond_create (bond_out, size, edge_2, central_1, 3);

```

Code Segment 19 - L_0 transformation bond creation

These transformations are meant to be applied to fullerene graphs, hence the fact that they only provide information in regards to the atom bonds (topological information).

3.3.1.2. $C_{28} \rightarrow C_{24}$ Transformation

As shown in Figure 55, C_{28} can be obtained from C_{24} by adding four additional atoms, one at the center of a hexagon, and another three in alternating pentagons around, so three new pentagons are created at the center and three adjacent pentagons are converted into hexagons, preserving the total number of pentagons (12).

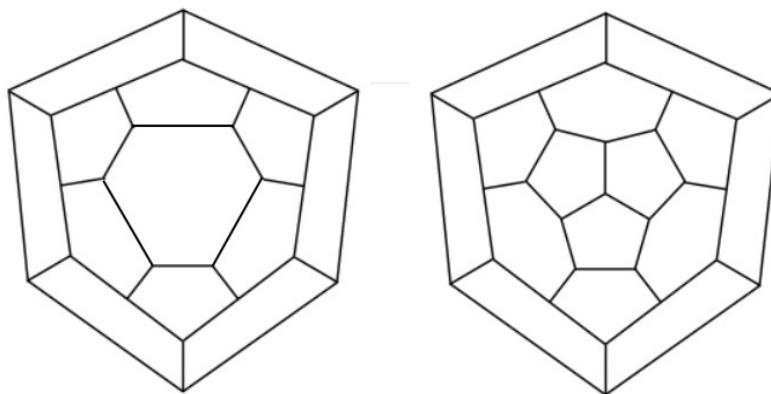


Figure 55 - Graphs of C_{24} (left) and C_{28} (right), obtained by adding the central atom and three atoms in alternating pentagons

```

void ma (double *atom_in, int *bond_in, int size,
double *atom_out, int *bond_out, int *hexagon)
{
double x, y, z;
int near_1, near_2, near_3;
int center_1;

```

```

copy_matrix_bond (bond_in, size, bond_out, size + 4);
copy_matrix_atom (atom_in, size, atom_out, size + 4);

/*****
 * new 4 atoms *
*****/

near_1 = size++;
x = (atom_in[3 * hexagon[0] + 0] + atom_in[3 * hexagon[1] + 0]) / 2;
y = (atom_in[3 * hexagon[0] + 1] + atom_in[3 * hexagon[1] + 1]) / 2;
z = (atom_in[3 * hexagon[0] + 2] + atom_in[3 * hexagon[1] + 2]) / 2;
atom_out[3 * near_1 + 0] = x;
atom_out[3 * near_1 + 1] = y;
atom_out[3 * near_1 + 2] = z;

near_2 = size++;
x = (atom_in[3 * hexagon[2] + 0] + atom_in[3 * hexagon[3] + 0]) / 2;
y = (atom_in[3 * hexagon[2] + 1] + atom_in[3 * hexagon[3] + 1]) / 2;
z = (atom_in[3 * hexagon[2] + 2] + atom_in[3 * hexagon[3] + 2]) / 2;
atom_out[3 * near_2 + 0] = x;
atom_out[3 * near_2 + 1] = y;
atom_out[3 * near_2 + 2] = z;

near_3 = size++;
x = (atom_in[3 * hexagon[4] + 0] + atom_in[3 * hexagon[5] + 0]) / 2;
y = (atom_in[3 * hexagon[4] + 1] + atom_in[3 * hexagon[5] + 1]) / 2;
z = (atom_in[3 * hexagon[4] + 2] + atom_in[3 * hexagon[5] + 2]) / 2;
atom_out[3 * near_3 + 0] = x;
atom_out[3 * near_3 + 1] = y;
atom_out[3 * near_3 + 2] = z;

center_1 = size++;
x = (atom_out[3 * near_1 + 0] + atom_out[3 * near_2 + 0] + atom_out[3 * near_3 + 0]) / 3;
y = (atom_out[3 * near_1 + 1] + atom_out[3 * near_2 + 1] + atom_out[3 * near_3 + 1]) / 3;
z = (atom_out[3 * near_1 + 2] + atom_out[3 * near_2 + 2] + atom_out[3 * near_3 + 2]) / 3;
atom_out[3 * center_1 + 0] = x;
atom_out[3 * center_1 + 1] = y;
atom_out[3 * center_1 + 2] = z;

```

```

/*****
 * remove 3 old bonds *
 *****/

bond_create (bond_out, size, hexagon[0], hexagon[1], 0);
bond_create (bond_out, size, hexagon[2], hexagon[3], 0);
bond_create (bond_out, size, hexagon[4], hexagon[5], 0);

/*****
 * create 9 new bonds *
 *****/

bond_create (bond_out, size, hexagon[0], near_1, 4);
bond_create (bond_out, size, hexagon[1], near_1, 4);
bond_create (bond_out, size, hexagon[2], near_2, 4);
bond_create (bond_out, size, hexagon[3], near_2, 4);
bond_create (bond_out, size, hexagon[4], near_3, 4);
bond_create (bond_out, size, hexagon[5], near_3, 4);
bond_create (bond_out, size, center_1, near_1, 4);
bond_create (bond_out, size, center_1, near_2, 4);
bond_create (bond_out, size, center_1, near_3, 4);
}

```

Code Segment 20 - $C_{24} \rightarrow C_{28}$ transformation

3.3.2. Nanotube Fullerenes

According to the analysis of Hasheminezhad et al. [63], (5,0)-type nanotube fullerenes can be generated by F expansions. As can be seen in Figure 23, this transformation adds five hexagonal faces around an atomic configuration consisting of a central pentagon surrounded by five other pentagons. In terms of nanotubes, transformation F corresponds to increasing a (5,0)-type nanotube's height through the addition of a ring layer. Based on this approach, all that is necessary to generate (5,0)-type nanotube fullerenes is to cover both ends of (5,0)-type nanotubes with two caps with the configuration from Figure 23.

C_{20} corresponds to the case where both caps are simply joined (a nanotube with a height equal to 0). By adding one layer of five hexagonal rings (a nanotube with a height equal to 1) in between the caps results in C_{30} . The addition of another layer leads to the creation of C_{40} , so the general formula for (5,0)-type nanotube fullerenes is: $C_{20 + 10k}$, where k is the number of ring layers (or the height of the nanotube).

The process of adding caps to both ends of a carbon nanotube, in order to create nanotube fullerenes can be applied to any (m,n) nanotube, in particular to (5,5)- and (6,6)-type nanotubes. For (5,5)-type nanotube fullerenes, the cap is formed by one pentagon surrounded by five hexagons which are linked to five alternating pentagons and hexagons, and the general formula is: $C_{60 + 10k}$. For (6,6)-type nanotube fullerenes, the cap is formed by one hexagon surrounded by six hexagons which are linked to six alternating pentagons and hexagons, and the general formula is: $C_{72 + 12k}$. (5,5)-type nanotube fullerenes C_{60} and C_{70} are of particular interest, as they are the experimentally observed IPR isomers.

The corresponding caps for each type of nanotube fullerene can be seen in Figure 56.

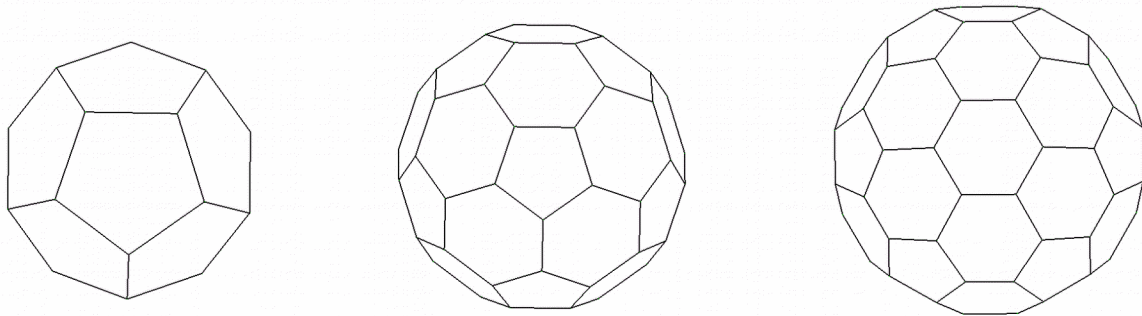


Figure 56 - From left to right, top down view of caps for (5,0)-, (5,5)- and (6,6)-type nanotubes

3.3.2.1. (5-0)-type Nanotube Fullerenes

The general procedure to create nanotube fullerenes is to create two caps at both ends of a nanotube and bond the external atoms of the caps (the ones more distanced to the cap center) with the top and bottom atoms of the tube. For a better understanding, let us examine the case of (5,0)-type nanotube fullerenes.

In order to form the correspondent cap, it is necessary to place two pentagons, one above the top end and another below the bottom end of the tube, respectively. These pentagons must be concentric with the tube, and its sides must be equal to the length of the bond between atoms. To generate the pentagons, note that it is possible to divide a pentagon into five equal triangles of length l (see Figure 57). The straight lines that unite the center of the pentagon and its vertices are separated by θ angles, equal to $2\pi / 5$.

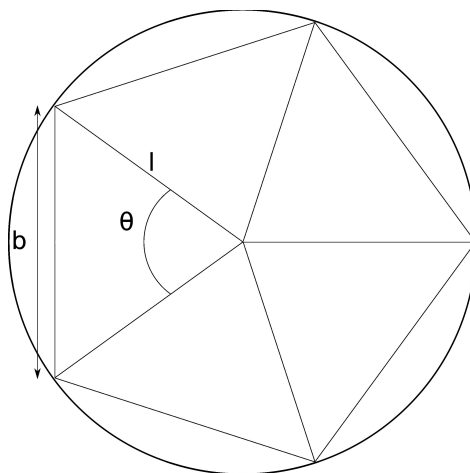


Figure 57 - Pentagon circumscribed in circumference. θ is the inner angle of the triangles, which is known to be $2\pi / 5$, b is the bond length

From this, it is clear that:

$$\frac{b}{2l} = \sin\left(\frac{\theta}{2}\right) \quad (31)$$

Rearranging the equation yields:

$$l = \frac{b}{2 \sin\left(\frac{\pi}{5}\right)} \quad (32)$$

The atoms must be generated at a distance l from the center, separated by $2\pi / 5$ degrees. The five bottom atoms of the nanotube make angles with the horizontal x axis equal to multiples of $2\pi / 5$, starting from 0, so the five atoms of the bottom pentagon are generated with the same angles in order to be aligned. In the case of the five top atoms of the nanotube, they will make the same angles with the horizontal x axis in case the number of ring layers is odd, otherwise they will be rotated by π . In the latter case, the top pentagon atoms also need to be rotated by π so they are aligned with the nanotube atoms. The angle at which the atoms should be generated is given by the following formula:

$$\alpha = \alpha_{start} + \frac{i \cdot 2\pi}{5} \quad (33)$$

where i is between 0 and 4. α_{start} is always 0 for the bottom pentagon, and is either 0 or π for the top pentagon whether the number of ring layers is odd or even, respectively.

```

/*****
* build atoms and bonds for a (5,0) nanotube cap, *
* formed by 1 pentagon linked to 5 pentagons *
*****/

for (i = 0; i < 5; i++)
{
  angle = angle_start + i * 2 * GAMGI_MATH_PI / 5;
  x = radius * cos (angle);
  y = radius * sin (angle);

  element = (i % 2 == 0) ? element1 : element2;
  atoms[i] = static_atom (molecule, element, x, y, z_top);
}

```

Code Segment 21 - Pentagon atom creation for (5,0)-type nanotube fullerenes

To calculate the height at which the pentagons should be placed such that the bonds they form with the tube will have the same length as the other bonds in the structure, consider Figure 58.

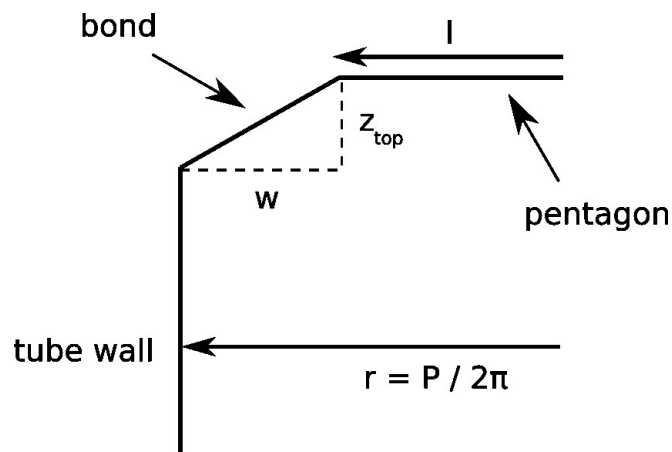


Figure 58 - The height at which the cap should be placed, z_{top} , should be such that the bonds between the cap and the tube have the expected bond length

The value of the width, w , is given by:

$$w = \frac{P}{2\pi} - l \quad (34)$$

where P is the perimeter of the nanotube. The height, z_{top} , is given by:

$$z_{top} = \frac{h}{2} + \sqrt{b^2 - w^2} \quad (35)$$

where h is the height of the nanotube. It is positive for the top cap and negative for the bottom cap. The x and y coordinates are simply given by:

$$x = l \cdot \cos(\alpha) \quad (36)$$

$$y = l \cdot \cos(\alpha) \quad (37)$$

```
void static_caps_5_0 (gamgi_molecule *molecule,
gamgi_atom **driver, int max, int rings, double perimeter,
double height, int element1, int element2, double bond)
{
double radius, z_top, width;

/*****
* build bottom and top caps for (5,0) nanotubes, *
* formed by a single pentagon linked to 5 pentagons *
*****/

radius = 0.5 * bond / sin (GAMGI_MATH_PI/5);
```

```

width = 0.5 * perimeter / GAMGI_MATH_PI - radius;
z_top = height / 2 + sqrt (bond * bond - width * width);

/*****
 * cap below (to build cap or cage) *
 *****/

static_lp5p (molecule, driver, max, 0, 0, 0,
FALSE, element1, element2, radius, -z_top, 0.0);

if (rings < 0) return;

/*****
 * cap above (to build cage) *
 *****/

if (rings % 2 == 0)
  static_lp5p (molecule, driver, max, 3, rings / 2, rings,
  TRUE, element1, element2, radius, z_top, GAMGI_MATH_PI);
else
  static_lp5p (molecule, driver, max, 0, rings / 2, rings,
  TRUE, element1, element2, radius, z_top, 0.0);
}

```

Code Segment 22 - Pentagon height calculation and cap creation for (5,0)-type nanotube fullerenes

To create the bonds, it is necessary to bond each of the five atoms of the pentagons to the five bottom or top atoms of the nanotube, depending on the cap. By “unfolding” the nanotube and caps, it is possible to verify that the atoms of the bottom cap are connected to A type atoms in the nanotube with index j equal to 0, and that the atoms of the top cap are connected to B type atoms of the nanotube with index j equal to the number of rings. The bond creation is illustrated in Figure 59.

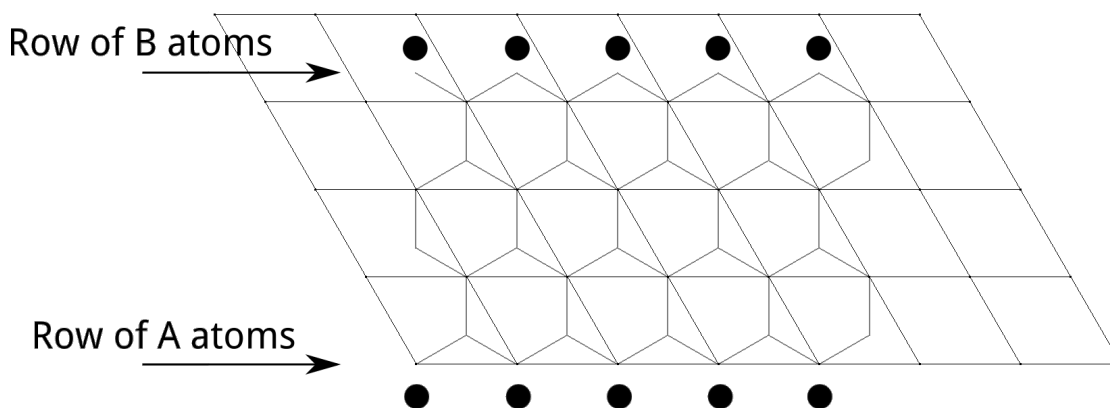


Figure 59 - Bonding scheme for the (5,5) cap. The black dots represent the cap atoms.

```

j = j_start;
for (i = 0; i < 5; i++)
{
  k = (j + 1) % 5;

  /*****
   * create cap bonds *
   *****/

  bond = gamgi_chem_bond_create (atoms[j], atoms[k]);
  gamgi_chem_bond_name (bond, bond->object.name);
}

```



```

gamgi_mesa_start_bond (bond, gamgi->bond);

/*****
 * create tube-cap bonds *
 *****/

number = gamgi_chem_graphene_index (2 * max, max, 0, a + i, b, type);
atom = driver[number];
if (atom != NULL)
{
    bond = gamgi_chem_bond_create (atoms[j], atom);
    gamgi_chem_bond_name (bond, bond->object.name);
    gamgi_mesa_start_bond (bond, gamgi->bond);
}

j = k;
}
}

```

Code Segment 23 - (5,0)-type nanotube fullerenes bond creation

3.3.3. Results

With the implementation of these algorithms, it is now possible to generate fullerenes using GAMGI. Figures 60 to 62 show two fullerenes for each type of cap. C_{24} generated from C_{20} through transformation L_o and C_{28} obtained from the application of the $C_{24} \rightarrow C_{28}$ transformation to C_{24} can also be seen in figures 63 and 64.

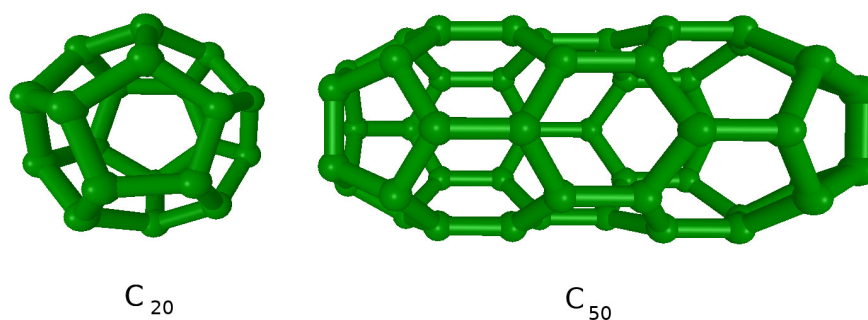


Figure 60 - C_{20} and (5,0)-type nanotube C_{50}

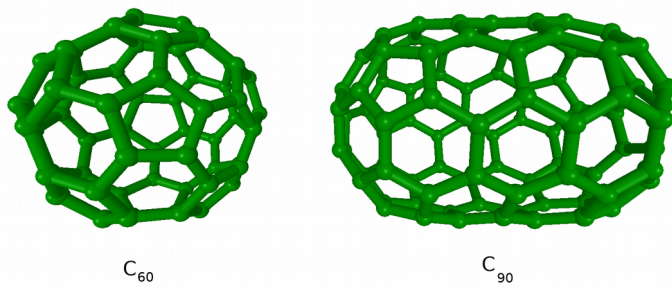


Figure 61 - C_{60} and (5,5)-type nanotube C_{90}

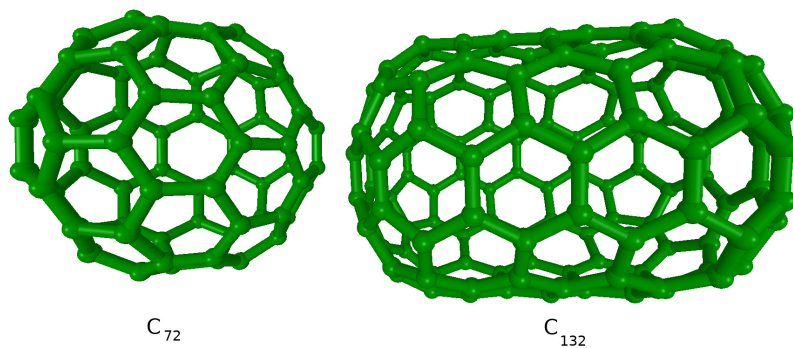


Figure 62 - C_{72} and (6,6)-type nanotube C_{132}

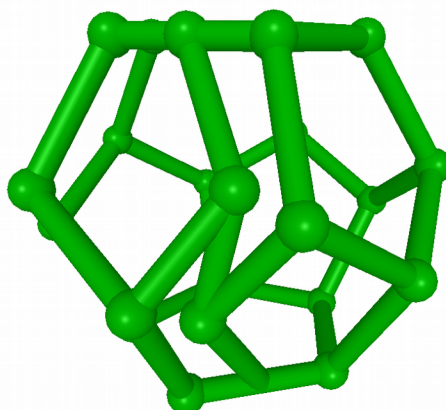


Figure 63 - C_{24} obtained from applying transformation L0

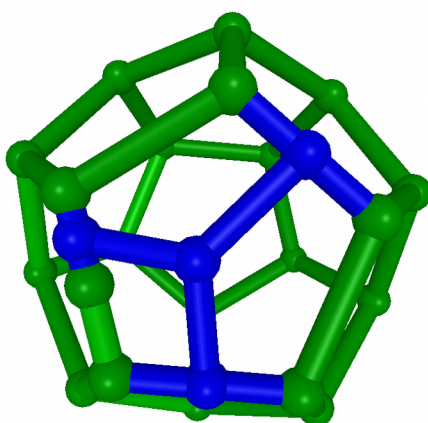


Figure 64 - Result of transformation C28 to C24

It is now possible to create three families of nanotube fullerenes through GAMGI, with the structures being geometrically optimized. Transformation L_0 and the transformation from C_{28} to C_{24} were successfully implemented in terms of topology, since a correct bonding between atoms has been achieved, but they are geometrically incorrect, so it was decided that the creation of fullerenes through these methods would not be incorporated into GAMGI.

4. Graphical interface

Several features have been added to GAMGI's graphical interface that allow for the introduction of data by a user in order to produce any of the molecules that have been described in this work. GAMGI's graphical interface was programmed in the C language, using the GTK graphical toolkit.

To create a molecule, one must choose the options **Molecule** → **Create** (Figures 65 and 66).



Figure 65 - The GAMGI graphical interface. The white arrow points to the "Molecule" option

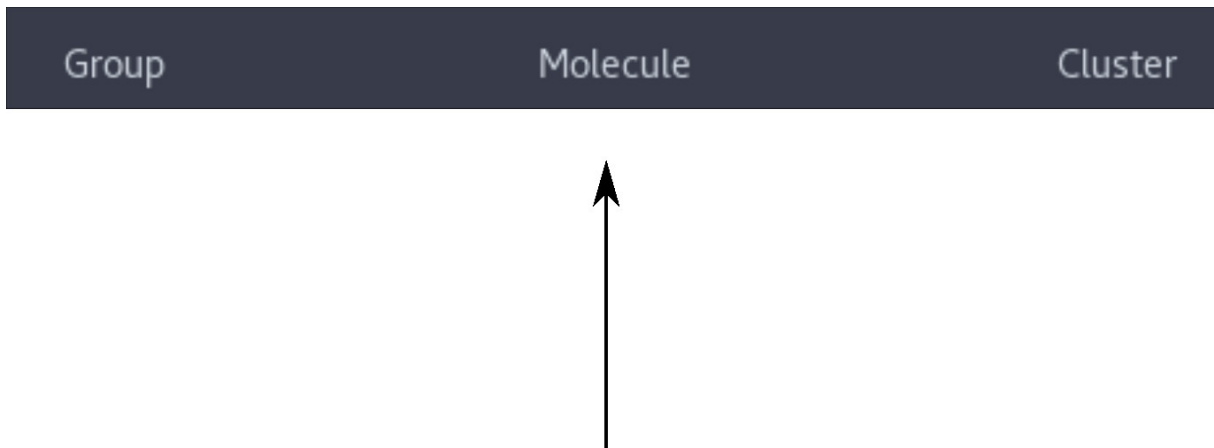


Figure 66 - Enlarged image of the "Molecule" option

By clicking on these options before, a simple toolbox would show up, but it would not allow for a functional input (Figure 67).

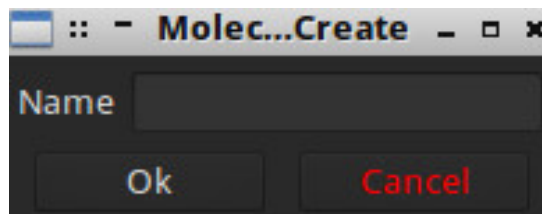


Figure 67 - Toolbox for the previous "Molecule" -> "Create" option

Now, in the "Method" scrollbox, the option "Graphene" is available. From it, the options to create graphene sheets, nanotubes, nanocones, fullerenes (cages) and their correspondent caps are available (Figure 68).

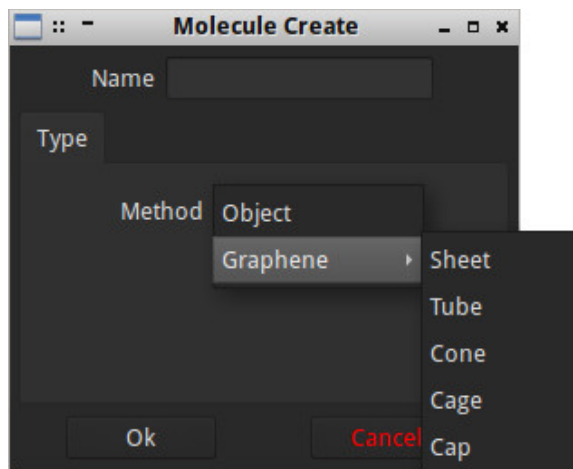


Figure 68 - Toolbox for the new "Molecule" -> "Create" option

Images 69 to 73 show the input options for each case:

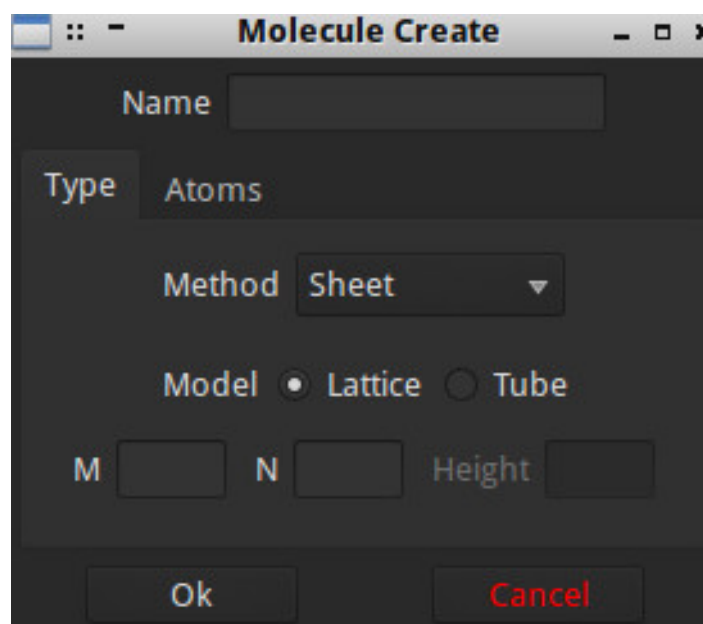


Figure 69 - Input options for a graphene sheet

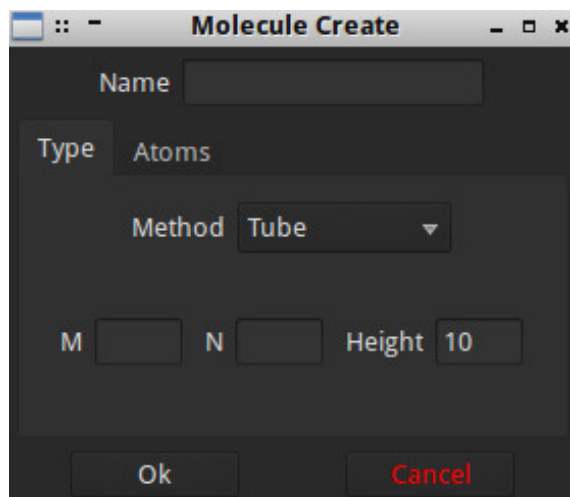


Figure 70 - Input options for nanotubes

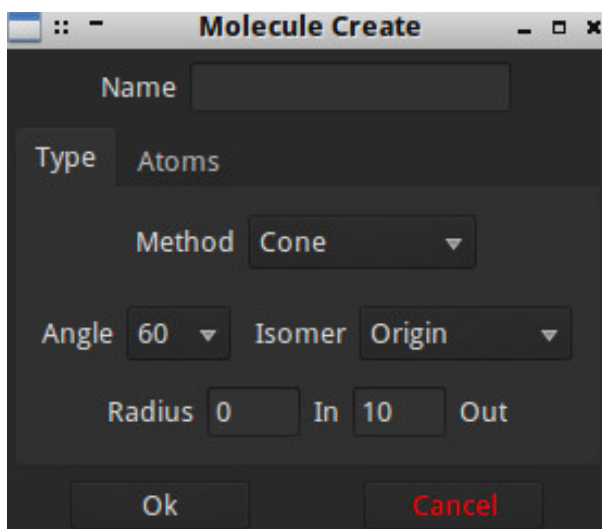


Figure 71 - Input options for nanocones

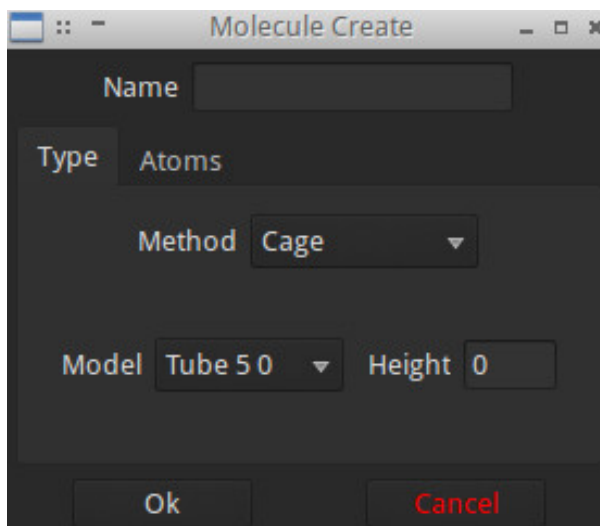


Figure 72 - Input options for fullerenes

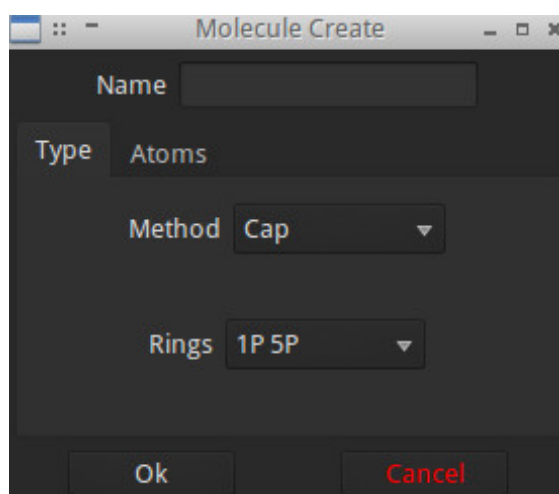


Figure 73 - Input options for caps

It is also possible, for any of the given molecules, to insert any combination of two elements, instead of simply carbon (Figure 74):

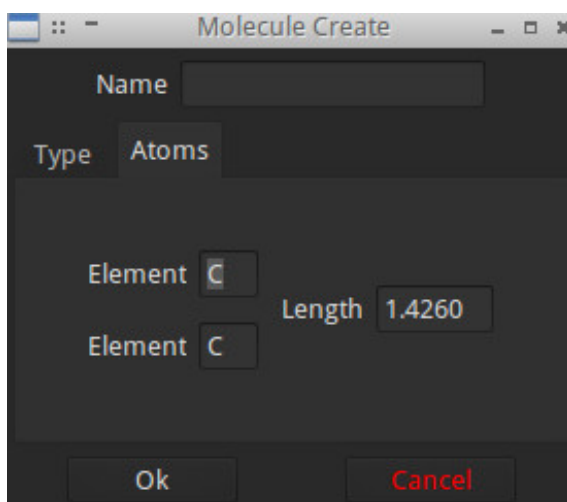


Figure 74 - Atom options

5. Conclusions

With this work, new features have been added to GAMGI which allow for the creation of three types of graphene based structures, based on simple user input: nanotubes, nanocones and fullerenes.

The implementation of the nanotube generation functionality was the most successful of all three. Among the harder aspects were: a) the definition of the height of the nanotubes and b) the creation of bonds.

An equation which returns the height of any kind of nanotube based on the number of ring layers was developed and found to produce very good results in the case of zig-zag and armchair nanotubes. In the case of chiral nanotubes, due to their more complex orientation, some loose bonds (atoms bonded to only one other atom) are created, making the overall look of the structure slightly worse.

The concept labelling atoms in the graphene lattice according to their position in the hexagonal lattice, which was discovered independently over the course of this work, proved to be extremely useful to generate bonds in the case of nanotubes, nanocones and nanotube fullerenes because of the six-fold symmetry of the honeycomb structure and the way these molecules can be modelled from a graphene sheet. For nanotubes in particular, it provided an easy way of identifying which atoms had to be discarded and what bonds to create due to periodic boundary conditions.

The result is that all three kinds of nanotubes, with any combination of (n,m) indices and with any height, can now be generated. The equations used to make the coordinate transformation from the 2D sheet to the 3D nanotube and the bonding scheme have proved to be correct, since the resulting structures present adequate geometries, with the bonds suffering very little distortion. The

only existing issue regards the creation of loose bonds in chiral nanotubes, but it can easily be corrected in the future, by either programming any such kinds of atoms from not being generated, or by bonding them with hydrogen atoms.

Although the implemented algorithms generate very satisfactory results, there are other algorithms which can be used for the generation of nanotubes [71] and it may be interesting to implement and compare the results of both cases in the future.

Parts of the algorithms for the generation of nanotubes served as a basis for the generation of nanocones, namely the graphene sheet generation and the atom labelling scheme. In what concerns cones with a single disclination, we were able to develop a satisfactory formula for the radii, which, just like in the case of nanotubes, returns a nanocone with a specified number of ring layers, although it presents the same problem as chiral nanotubes. The equations used for coordinate transformation are also adequate, producing cones with the correct apex angle for each disclination angle and geometrically ordered atoms. In particular, the formula used to calculate the new angle an atom has after it has been rotated works very well in the case of single disclinations, as it leads to a proportional rotation of every atom in the structure.

For nanocones with multiple disclinations, it was decided that the different models would be established based on: a) pentagon positions, b) disclination orientation and c) rotation of one or both edges. Most pentagon positions were derived from the existing scientific literature, along with the disclination orientations, but the decision on the way the gaps were closed was independent. As a result, it is now possible to generate sixteen different, open or closed, nanocone models, with at least one model for each number of pentagons. In the cases where the origin is changed to the center of a bond (and due to this), the formula works slightly worse, with more loose bonds being created, which means a new formula should be developed.

As was previously mentioned, the atom labelling provided a very useful tool for the creation of bonds. After a thorough analysis of each nanocone model, it was possible to invent a simple and coherent scheme to create the bonds independently of the disclination centers or the disclination orientation, and this is one of the strongest features that was developed and implemented throughout the course of this work.

Although all cones present correct apex angles, the structures are noticeably more distorted than in the case of single disclination cones. This is due to the fact that not all atoms of the structure are rotated to close the gaps, which results in an uneven distribution of atoms, as there are areas where atoms remain fixed and others where they are heavily dislocated from their original positions. The greater the number of disclinations, the more noticeable the effect is. As such, there is room for improvement of the structures of nanocones. This may be achieved, for example, by the incorporation of force-field optimization methods, which presents itself as a very interesting prospect for future work in GAMGI. It would also be of interest to implement in GAMGI the functionality to generate nanocones

through the cone-helix model, which although considerably more complex, would allow for nanocones with more than five discrete angles to be generated.

Among the three kinds of molecules of this work, fullerenes are arguably the hardest to generate, because unlike nanotubes and nanocones, they are not built directly from the graphene sheet. Instead, they are mostly generated starting from one of the most basic fullerenes, such as C_{20} , by first applying operations which involve solely topology to the fullerene graph, and then by generating correct 3D atomic positions. Although the previously suggested spiral algorithm has been proved to fail to generate all fullerenes above 380 atoms [52], new topological algorithms, based on L , B and F transformations, seem to generate all possible topological structures for fullerenes. However, no systematic algorithm has ever been proposed to generate the actual geometric coordinates.

We have managed to apply one of the simplest patch replacement operations (L_0) to generate C_{24} from C_{20} with correct bonds, but incorrect geometry. This means that there is a great room for improvement in this area. With the incorporation of fullerene graph generation and force-field optimization methods, it will be possible for GAMGI to produce at least a considerable amount of fullerene isomers based on simple user input.

In this work, through the “capping” of nanotubes, an operation which has proven to be fairly simple, it is now possible for GAMGI to generate three infinite families of nanotube fullerenes of the types (5,0), (5,5) and (6,6), with the molecules presenting a correct geometry. In particular, IPR C_{60} and C_{70} (particular cases of (5,5)-type nanotubes with 0 and 1 ring layer), which are two of the most experimentally observed fullerene isomers [52], can be generated in this way. The only setback to nanotube fullerenes is that all the pentagons of the structure are always located at the caps, and that fullerenes with a higher atom count will inevitably have an elongated shape. Also, just like fullerenes, there are cap isomers (there are 73 caps for (5,5)-type nanotube fullerenes alone), and this is also an aspect to be considered for future work.

6. References

- [1] "GAMGI - General Atomistic Modelling Graphic Interface." 30 Nov. 2016. <http://www.gamgi.org/>.
- [2] Wyckoff, Ralph Walter Graystone, and Ralph WG Wyckoff. *Crystal structures*. Vol. 2. New York: Interscience, 1960.
- [3] Cooper, Daniel R., et al. "Experimental review of graphene." *ISRN Condensed Matter Physics* 2012 (2012).
- [4] Balandin, Alexander A. "Thermal properties of graphene and nanostructured carbon materials." *Nature materials* 10.8 (2011): 569-581.
- [5] Apell, S. Peter, G. W. Hanson, and Carl Häggglund. "High optical absorption in graphene." *arXiv preprint arXiv:1201.3071* (2012).
- [6] Chen, Jian-Hao, et al. "Intrinsic and extrinsic performance limits of graphene devices on SiO₂." *Nature nanotechnology* 3.4 (2008): 206-209.
- [7] Akinwande, Deji, et al. "Large-Area Graphene Electrodes: Using CVD to facilitate applications in commercial touchscreens, flexible nanoelectronics, and neural interfaces." *IEEE Nanotechnology Magazine* 9.3 (2015): 6-14.
- [8] Lu, Xin, and Zhongfang Chen. "Curved pi-conjugation, aromaticity, and the related chemistry of small fullerenes." *Chemical Reviews* 105.10 (2005): 3643-3696.
- [9] Wang, Xueshen, et al. "Fabrication of ultralong and electrically uniform single-walled carbon nanotubes on clean substrates." *Nano letters* 9.9 (2009): 3137-3141.
- Eklund, Peter, et al. *International assessment of carbon nanotube manufacturing and applications*. World Technology Evaluation Center INC Baltimore MD, 2007.
- [10] <https://upload.wikimedia.org/wikipedia/commons/2/22/CntHAADF.jpg> (accessed on October 2016)
- [11] Giannopoulos, G. I., P. A. Kakavas, and N. K. Anifantis. "Evaluation of the effective mechanical properties of single walled carbon nanotubes using a spring based finite element approach." *Computational Materials Science* 41.4 (2008): 561-569.
- [12] Lee, J. H., and B. S. Lee. "Modal analysis of carbon nanotubes and nanocones using FEM." *Computational Materials Science* 51.1 (2012): 30-42.
- [13] Contreras, M. Leonor, et al. "Algorithm for nanotubes computer generation with different configurations." *Algorithms* 2.1 (2009): 108-120.
- Brinkmann, Gunnar, U. V. Nathusius, and A. H. R. Palser. "A constructive enumeration of nanotube caps." *Discrete Applied Mathematics* 116.1 (2002): 55-71.
- [14] Melchor, Santiago, and José A. Dobado. "CoNTub: an algorithm for connecting two arbitrary carbon nanotubes." *Journal of chemical information and computer sciences* 44.5 (2004): 1639-1646.
- Terrones, Humberto. "Curved graphite and its mathematical transformations." *Journal of mathematical chemistry* 15.1 (1994): 143-156.
- [15] Pujadó, Mercè Pacios. *Carbon nanotubes as platforms for biosensors with electrochemical and electronic transduction*. Springer Science & Business Media, 2012.

- [16] Oberlin, Agnes, M. Endo, and T. Koyama. "Filamentous growth of carbon through benzene decomposition." *Journal of crystal growth* 32.3 (1976): 335-349.
- [17] Endo, Morinobu. "Grow carbon fibers in the vapor phase." *Chemtech* (1988).
- [18] Abrahamson, John, Peter G. Wiles, and Brian L. Rhoades. "Structure of carbon fibres found on carbon arc anodes." *Carbon* 11.37 (1999): 1873-1874.
- [19] Kroto, Harold W., et al. "C 60: buckminsterfullerene." *Nature* 318.6042 (1985): 162-163.
- [20] Iijima, Sumio. "Helical microtubules of graphitic carbon." *nature* 354.6348 (1991): 56-58.
- [21] Mintmire, J. W., B. I. Dunlap, and C. T. White. "Are fullerene tubules metallic?." *Physical Review Letters* 68.5 (1992): 631.
- [22] Takeuchi, K., et al. "The State-of-the-art science and applications of carbon nanotubes." *Nanosyst. Phys. Chem. Math* 5.1 (2014): 15-24.
- Thostenson, Erik T., Zhifeng Ren, and Tsu-Wei Chou. "Advances in the science and technology of carbon nanotubes and their composites: a review." *Composites science and technology* 61.13 (2001): 1899-1912.
- [23] Sinha, Saion, et al. "Off-axis thermal properties of carbon nanotube films." *Journal of Nanoparticle Research* 7.6 (2005): 651-657.
- Pop, Eric, et al. "Thermal conductance of an individual single-wall carbon nanotube above room temperature." *Nano letters* 6.1 (2006): 96-100.
- Gore, Jay P., and Anup Sane. "Flame synthesis of carbon nanotubes." *Carbon Nanotubes-Synthesis, Characterization, Applications* 1 (2011): 16801.
- [24] Yu, Min-Feng, et al. "Strength and breaking mechanism of multiwalled carbon nanotubes under tensile load." *Science* 287.5453 (2000): 637-640.
- Coleman, Jonathan N., et al. "Small but strong: a review of the mechanical properties of carbon nanotube-polymer composites." *Carbon* 44.9 (2006): 1624-1652.
- [25] Jensen, K., et al. "Buckling and kinking force measurements on individual multiwalled carbon nanotubes." *Physical Review B* 76.19 (2007): 195436.
- [26] Wang, Zuankai, et al. "Polarity-dependent electrochemically controlled transport of water through carbon nanotube membranes." *Nano letters* 7.3 (2007): 697-702.
- [27] Hong, Seunghun, and Sung Myung. "A flexible approach to mobility." *Nature Nanotech* 2 (2007): 207-208.
- [28] "Pirahna USV built using nano-enhanced carbon prepreg". ReinforcedPlastics.com. 19 February 2009. Archived from the original on 3 March 2012.
- [29] He, Hua, et al. "Carbon nanotubes: applications in pharmacy and medicine." *BioMed research international* 2013 (2013).
- [30] Ge, Maohui, and Klaus Sattler. "Observation of fullerene cones." *Chemical physics letters* 220.3-5 (1994): 192-196.
- [31] Krishnan, A., et al. "Graphitic cones and the nucleation of curved carbon surfaces." *Nature* 388.6641 (1997): 451-454.

- [32] Iijima, S., et al. "Nano-aggregates of single-walled graphitic carbon nano-horns." *Chemical Physics Letters* 309.3 (1999): 165-170.
- [33] Wei, Jingxi. "Simulation of carbon nanocones using molecular dynamics." (2007).
- [34] Lu, X., et al. "Field electron emission of carbon-based nanocone films." *Applied Physics A* 82.2 (2006): 293-296.
- [35] Qu, Chao Qun, et al. "Electronic and field emission properties of carbon nanocones: a density functional theory investigation." *IEEE Transactions on Nanotechnology* 8.2 (2009): 153-158.
- [36] Jaszczak, John A., et al. "Naturally occurring graphite cones." *Carbon* 41.11 (2003): 2085-2092. Jaszczak, John A., and George W. Robinson. "Spherical and triskelial graphite from Gooderham, Ontario, Canada." (2000): 172.
- van der Pluijm, Ben A., and Katherine A. Carlson. "Extension in the Central Metasedimentary Belt of the Ontario Grenville: timing and tectonic significance." *Geology* 17.2 (1989): 161-164.
- Carlson, Katherine A., Ben A. van der Pluijm, and Simon Hanmer. "Marble mylonites of the Bancroft shear zone: Evidence for extension in the Canadian Grenville." *Geological Society of America Bulletin* 102.2 (1990): 174-181.
- Weis, Paul L. "Graphite skeleton crystals—A newly recognized morphology of crystalline carbon in metasedimentary rocks." *Geology* 8.6 (1980): 296-297.
- Gogotsi, Y., S. Dimovski, and J. A. Libera. "Conical crystals of graphite." *Carbon* 40.12 (2002): 2263-2267.
- [39] Ekşioğ̃lu, Burcu, and Arunan Nadarajah. "Structural analysis of conical carbon nanofibers." *Carbon* 44.2 (2006): 360-373.
- [40] Lee, J. H., and B. S. Lee. "Modal analysis of carbon nanotubes and nanocones using FEM." *Computational Materials Science* 51.1 (2012): 30-42.
- [41] Ansari, R., et al. "On the vibration of single-walled carbon nanocones: molecular mechanics approach versus molecular dynamics simulations." *Shock and Vibration* 2014 (2014).
- Kürkçü, Ömür Kivanç, and Ersin Aslan. "Atom Bond Connectivity Index of Carbon Nanocones and An Algorithm." *Applied Mathematics and Physics* 3.1 (2015): 6-9.
- [42] Lin, Cheng-Te, et al. "Graphene structure in carbon nanocones and nanodiscs." *Langmuir* 23.26 (2007): 12806-12810.
- [43] Naess, Stine Nalum, et al. "Carbon nanocones: wall structure and morphology." *Science and Technology of advanced materials* (2016).
- Endo, M., et al. "Structural characterization of cup-stacked-type nanofibers with an entirely hollow core." *Applied Physics Letters* 80.7 (2002): 1267-1269.
- [44] Iijima, S., et al. "Nano-aggregates of single-walled graphitic carbon nano-horns." *Chemical Physics Letters* 309.3 (1999): 165-170.
- [45] Double, D. D., and A. Hellawell. "Cone-helix growth forms of graphite." *Acta Metallurgica* 22.4 (1974): 481-487.
- [46] Buldum, Alper, and Jian Ping Lu. "Electron field emission properties of closed carbon nanotubes." *Physical review letters* 91.23 (2003): 236801.

- [47] Wang, Qing Hua, et al. "A nanotube-based field-emission flat panel display." *Applied Physics Letters* 72.22 (1998): 2912-2913.
- [48] Bonard, J-M., et al. "Field emission properties of carbon nanohorn films." *Journal of applied physics* 91.12 (2002): 10107-10109.
- [49] Shenderova, O. A., et al. "Predicted structure and electronic properties of individual carbon nanocones and nanostructures assembled from nanocones." *Nanotechnology* 12.3 (2001): 191.
- [50] Osawa, Eiji. "Superaromaticity." *Kagaku* 25.854 (1970): 101.
- [51] Thrower, P.A. (1999). "Editorial". *Carbon*. **37** (11): 1677–1678.
- [52] Fowler, Patrick W., and D. E. Manolopoulos. An atlas of fullerenes. Courier Corporation, 2006.
Fowler, Patrick W., et al. *A generalized ring spiral algorithm for coding fullerenes and other cubic polyhedra*. University of Ljubljana. Department of Mathematics, 1998.
- Fowler, Patrick W., and Kevin M. Rogers. "Spiral codes and Goldberg representations of icosahedral fullerenes and octahedral analogues." *Journal of chemical information and computer sciences* 41.1 (2001): 108-111.
- [53] Brinkmann, Gunnar, and Andreas WM Dress. "A constructive enumeration of fullerenes." *Journal of Algorithms* 23.2 (1997): 345-358.
- [54] Schwerdtfeger, Peter, Lukas N. Wirz, and James Avery. "The topology of fullerenes." *Wiley Interdisciplinary Reviews: Computational Molecular Science* 5.1 (2015): 96-145.
- Goedgebeur, Jan, and Brendan D. McKay. "Fullerenes with distant pentagons." *arXiv preprint arXiv:1508.02878* (2015).
- [55] Schwerdtfeger, Peter, Lukas Wirz, and James Avery. "Program fullerene: a software package for constructing and analyzing structures of regular fullerenes." *Journal of computational chemistry* 34.17 (2013): 1508-1526.
- Senn, Peter. "Computation of the cartesian coordinates of buckminsterfullerene." *J. Chem. Educ* 72.4 (1995): 302.
- [56] Kroto, H. W. "The stability of the fullerenes C_n , with $n= 24, 28, 32, 36, 50, 60$ and 70 ." *Nature* 329.6139 (1987): 529-531.
- [57] Raghavachari, Krishnan. "Ground state of C_{84} : two almost isoenergetic isomers." *Chemical physics letters* 190.5 (1992): 397-400.
- [58] G. Brinkmann, O. Delgado Friedrichs, S. Lisker, A. Peeters, N. Van Cleemput, **CaGe - a Virtual Environment for Studying Some Special Classes of Plane Graphs - an Update**, *MATCH Commun. Math. Comput. Chem.*, 63(3), pp. 533-552, 2010, <http://www.math.uni-bielefeld.de/~CaGe>
- [59] https://en.wikipedia.org/wiki/Graph_theory#/media/File:6n-graf.svg (accessed on September 2016)
- [60] https://upload.wikimedia.org/wikipedia/commons/7/7e/Graph_of_20-fullerene_w-nodes.svg (accessed on September 2016)
- [61] Liu, X., et al. "Generation of carbon-cage polyhedra." *Journal of computational chemistry* 12.10 (1991): 1252-1259.
- [62] Brinkmann, Gunnar, et al. "Growing fullerenes from seed: Growth transformations of fullerene polyhedra." *Chemical physics letters* 428.4 (2006): 386-393.

[63] Hasheminezhad, Mahdiah, Herbert Fleischner, and Brendan D. McKay. "A universal set of growth operations for fullerenes." *Chemical Physics Letters* 464.1 (2008): 118-121.

[64] Brinkmann, Gunnar, Jan Goedgebeur, and Brendan D. McKay. "The generation of fullerenes." *Journal of chemical information and modeling* 52.11 (2012): 2910-2918.

Goedgebeur, Jan, and Brendan D. McKay. "Recursive generation of IPR fullerenes." *Journal of Mathematical Chemistry* 53.8 (2015): 1702-1724.

[65] G. Brinkmann, K. Coolsaet, J. Goedgebeur, H. Mélot, House of Graphs: a database of interesting graphs, *Discrete Applied Mathematics*, 161:311-314, 2013 (DOI). Available at <http://hog.grinvin.org>

[66] Fulcheri, Laurent, et al. "Fullerene production in a 3-phase AC plasma process." *Carbon* 38.6 (2000): 797-803.

[67] Lalwani, Gaurav, and Balaji Sitharaman. "Multifunctional fullerene-and metallofullerene-based nanobiomaterials." *Nano Life* 3.03 (2013): 1342003.

[68] Bakry, Rania, et al. "Medicinal applications of fullerenes." *International journal of nanomedicine* 2.4 (2007): 639.

[69] Bosi, Susanna, et al. "Fullerene derivatives: an attractive tool for biological applications." *European journal of medicinal chemistry* 38.11 (2003): 913-923.

[70] Kernighan, Brian W., and Dennis M. Ritchie. "The C programming language." (2006).

[71] Contreras, M. Leonor, et al. "Computational algorithms for fast-building 3D carbon nanotube models with defects." *Journal of Molecular Graphics and Modelling* 38 (2012): 389-395.

Appendix A – Creation schemes for the non-origin nanocone isomers

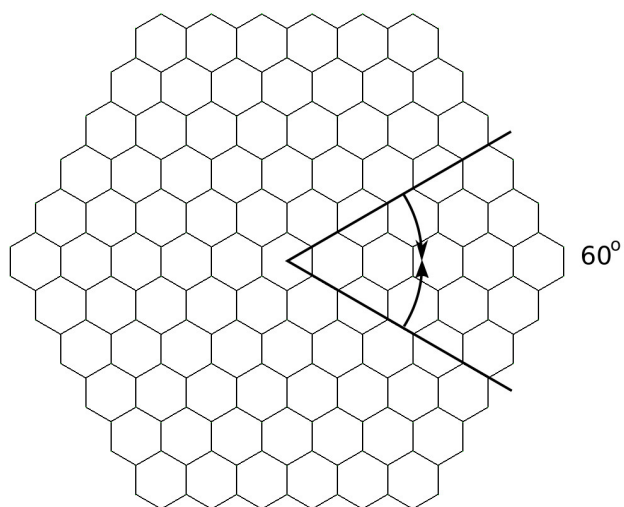


Fig.1 – 60 Origin

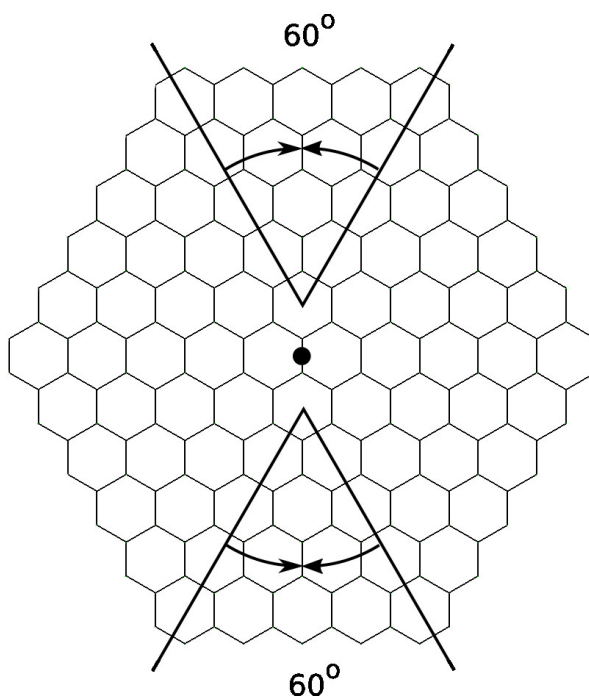


Fig.2 – 120 Edge

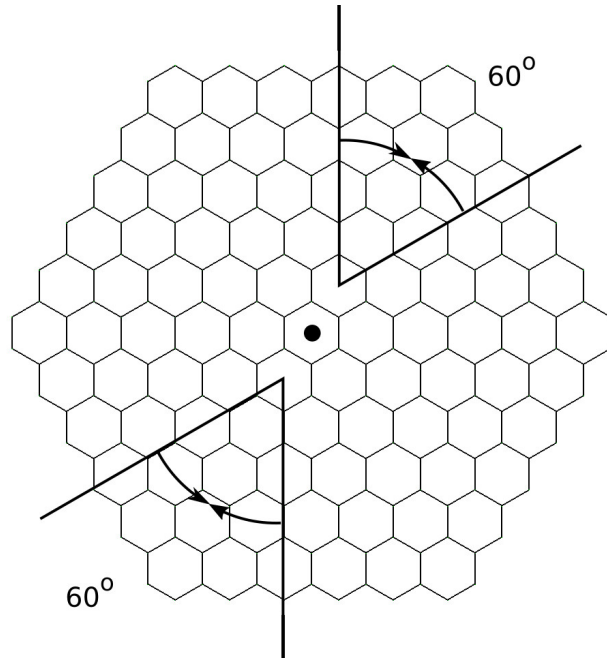


Fig.3 - 120 Hexagon

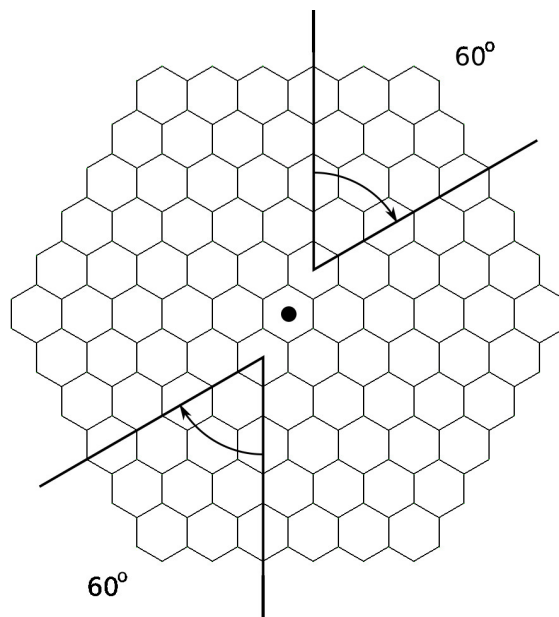


Fig.4 - 120 Single

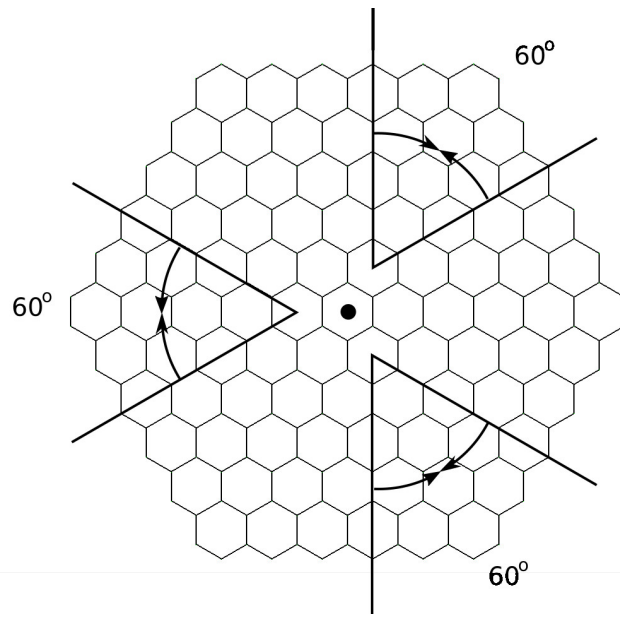


Fig.5 - 180 Double

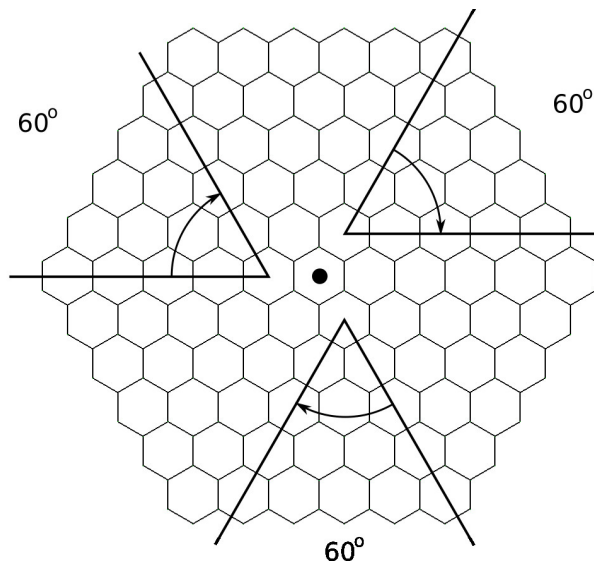


Fig.6 - 180 Single

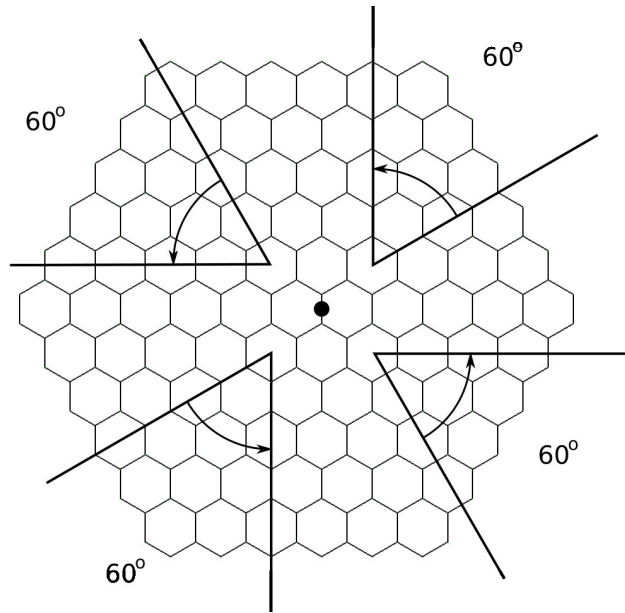


Fig.7 - 240 Axis

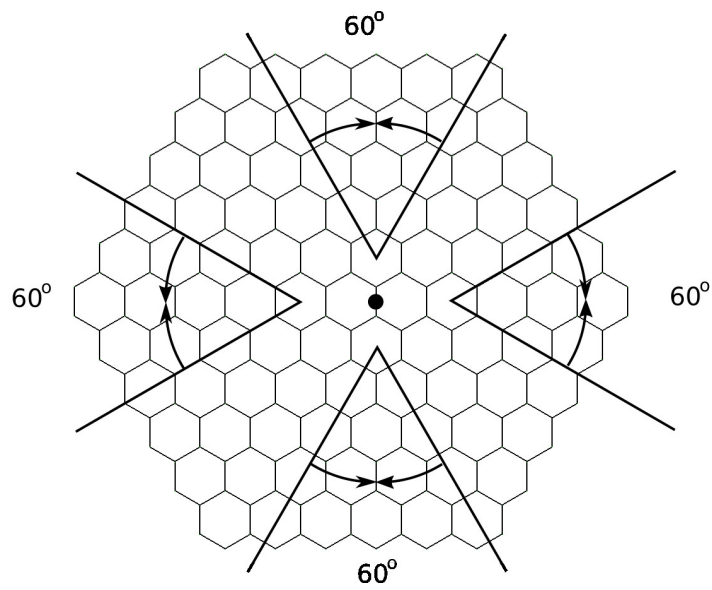


Fig.8 - 240 Edge

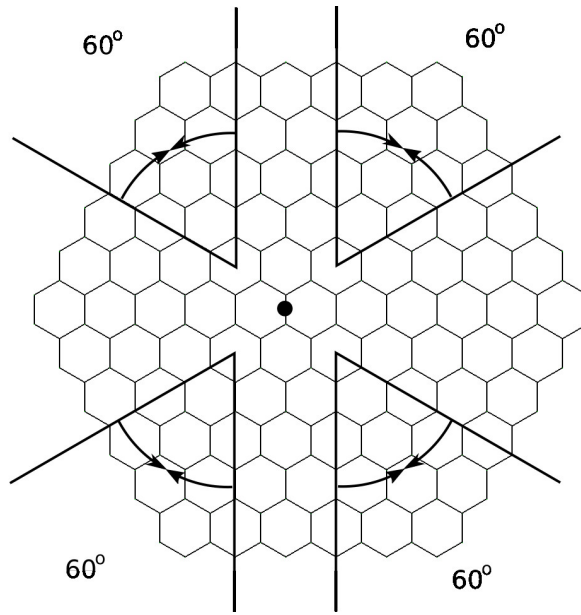


Fig.9 - 240 Hexagon

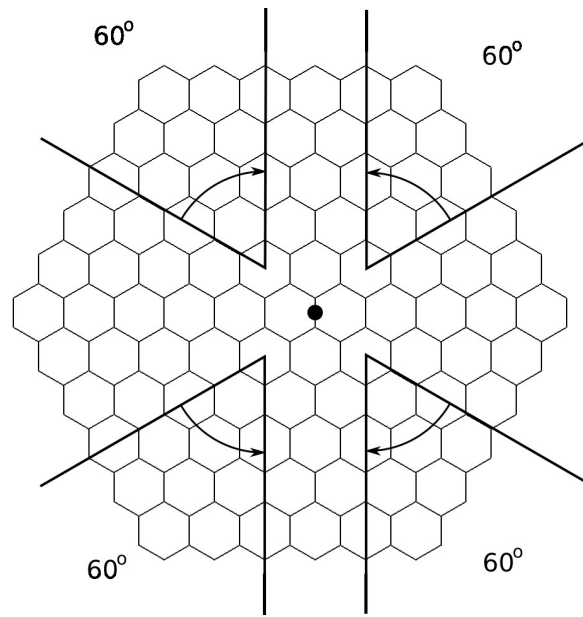


Fig.10 - 240 Plane

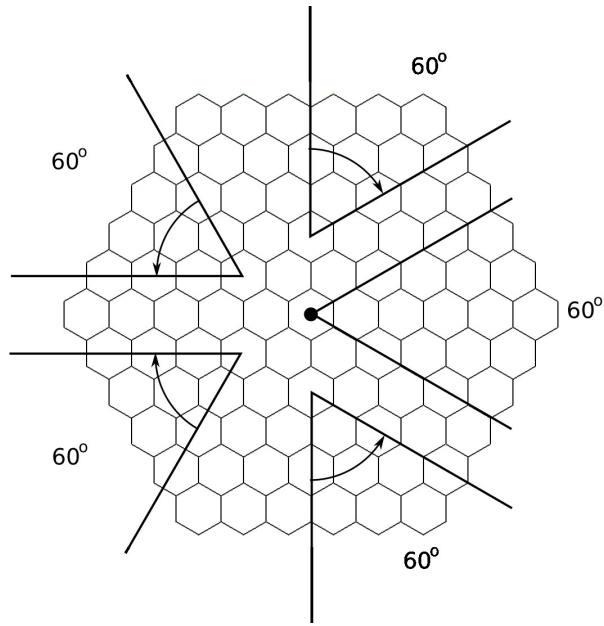


Fig.11 – 300 Plane

Appendix B – Table of nanocone bond indices

Isomer	Gap	i_0	j_0	k_0	i_1	j_1	k_1	ii_0	jj_0	kk_0	ii_1	jj_1	kk_1
60 origin	1	-1	-1	1	-1	-1	1	1	0	0	1	1	0
60 double	1	-1	0	0	-1	0	0	2	-2	1	1	3	1
120 origin	1	-1	-1	1	-1	-1	0	1	0	0	0	1	0
120 edge	1	-1	0	0	-1	0	0	1	1	0	0	1	0
	2	-1	-1	1	-1	-1	1	-1	-1	0	0	-1	0
120 hexagon	1	0	0	1	0	0	1	1	3	1	2	1	1
	2	-1	-1	0	-1	-1	0	-1	-2	1	-3	-1	1
120 single	1	0	0	1	0	0	1	1	0	0	1	1	0
	2	-1	-1	0	-1	-1	0	-1	0	0	-1	-1	0
180 origin	1	-1	-1	1	0	-1	1	1	0	0	-1	0	0
180 double	1	0	0	1	0	0	1	1	3	1	2	1	1
	2	-1	-1	1	-1	-1	1	-3	3	1	-1	-2	1
	3	0	-1	1	0	-1	1	-3	-1	1	2	-2	1
180 single	1	-1	-1	1	-1	-1	1	0	1	0	-1	0	0
	2	0	-1	1	0	-1	1	-1	-1	0	0	-1	0
	3	0	0	1	0	0	1	1	0	0	1	1	0
240 origin	1	-1	-1	1	0	0	0	1	0	0	-1	-1	0
240 edge	1	-1	0	0	-1	0	0	1	1	0	0	1	0
	2	-1	-1	1	-1	-1	1	-1	-1	0	0	-1	0

	3	0	0	0	0	0	0	2	-2	1	1	3	1
	4	-2	-1	1	-2	-1	1	-3	3	1	-1	-2	-1
240 hexagon	1	0	0	1	0	0	1	1	3	1	2	1	1
	2	-1	0	1	-1	0	1	0	1	0	-1	0	0
	3	-2	-1	0	-2	-1	0	-1	-2	1	-3	-1	1
	4	-1	-1	0	-1	-1	0	1	0	0	0	-1	0
240 axis	1	0	0	1	0	0	1	1	3	1	2	1	1
	2	-1	0	1	-1	0	1	0	1	0	-1	0	0
	3	-2	-1	0	-2	-1	0	-1	-2	1	-3	-1	1
	4	-1	-1	0	-1	-1	0	0	-1	0	1	0	0
240 plane	1	0	0	1	0	0	1	1	3	1	2	1	1
	2	-1	0	1	-1	0	1	2	1	1	-3	3	1
	3	-2	-1	0	-2	-1	0	-1	-2	1	-3	-1	1
	4	-1	-1	0	-1	-1	0	-3	-1	1	2	-2	1
300 origin	1	0	-1	1	0	-1	1	1	0	0	0	-1	0
300 edge	1	0	1	0	0	1	0	1	3	1	2	1	1
	2	-1	0	1	-1	0	1	0	1	0	-1	0	0
	3	-2	-1	0	-2	-1	0	-1	0	0	-1	-1	0
	4	-1	-2	1	-1	-2	1	-3	-1	1	2	-2	1
	5	-1	0	0	-1	0	0	2	-2	1	1	3	1