GENERATION AND SIMULATIONS OF NANOSTRUCTURES OF CAGE STRUCTURES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

EMRE TAŞCI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF DOCTOR OF PHILOSOPHY
IN
PHYSICS

JULY 2007

Approval of the thesis:

GENERATION AND SIMULATIONS OF NANOSTRUCTURES OF CAGE STRUCTURES

submitted by **EMRE TAŞCI** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy in Physics Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen
Dean, Graduate School of **Natural and Applied Sciences** ____________________

Prof. Dr. Sinan Bilikmen
Head of Department, **Physics** ____________________

Prof. Dr. Şakir Erkoç
Supervisor, **Physics Dept., METU** ____________________

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu
Electrical and Electronics Engineering Dept., METU ____________________

Prof. Dr. Şakir Erkoç
Physics Dept., METU ____________________

Prof. Dr. Demet Gülen
Physics Dept., METU ____________________

Prof. Dr. Ümit Kızıloğlu
Physics Dept., METU ____________________

Assoc. Prof. Dr. Oğuz Gülseren
Physics Dept., Bilkent University ____________________

Date: 18.07.2007

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Emre Taşcı

Signature :

ABSTRACT

GENERATION AND SIMULATIONS OF NANOSTRUCTURES OF CAGE STRUCTURES

Taşcı, Emre
Ph.D., Department of Physics
Supervisor : Prof. Dr. Şakir Erkoç

July 2007, 97 pages

This thesis proposes algorithms to construct various nanosystems such as nanotori, nanogear and nanojunctions based on graphite type structures, exploiting the observed pentagonal and heptagonal defects. These produced systems are then simulated to test for their thermal stability and for their electronic properties. A brief review of the methods used is also included.

Keywords: Graphite based nanosystems, construction algorithm, molecular dynamics, simulation.

ÖZ

KAFES YAPIDAKİ NANOYAPILARIN ÜRETİLMESİ VE BENZETİMLERİ

Taşçı, Emre
Doktora, Fizik Bölümü
Tez Yöneticisi : Prof. Dr. Şakir Erkoç

Temmuz 2007, 97 sayfa

Bu çalışma nanotoruslar, nanoçarklar, nanobirleşimler ve benzeri grafit tabanlı nanoyapıların gözlemlenmiş olan beşgen ve yedigen bozukluklardan faydalanılarak üretilmesine dair algoritmalar sunmaktadır. Üretilen bu yapılar ayrıca ısıya karşı dayanıklılıklarını incelemek üzere benzetimlere tabi tutulmuş ve elektronik özelliklerini belirlemek üzere birtakım hesaplamalarda bulunulmuştur. Çalışma, kullanılan yöntemlere dair kısa bir açıklama da içermektedir.

Anahtar Kelimeler: Grafit tabanlı nanosistemler, üretim algoritmaları, moleküler dinamik, benzetim.

To Ece, for the hope of a beautiful future,
and to Michael ‘Mike’ Slackenerny for the mirror image.

ACKNOWLEDGEMENTS

I would like to express my gratitude to Prof. Dr. Şakir Erkoç with whom during the seven years I studied in METU, I learned every day a new aspect of scientific discipline and collaboration. It is through his advisorship that I became accustomed to dense usage and adaptation of computational skills with respect to the encountered physical issues. I will always cherish the memories of the colleagues I have met in our workgroup making me feel like I was among the family members. Barış Malcıoğlu, more like a brother to me; ex-group partner Efe Yazgan currently on route to the CERN and who is the 'curly' of our distorted version of the three stogees with Barış and me; Hüseyin Oymak now happily moved to Seul, always the elder brother; Deniz Çalışır Tekin and her ever smiling face; Rengin Peköz the little sister with her wit and *da coolness*; Nazım Dugan, whom we had made to go astray from the illuminated way of dedicated studying into the forked paths of 'party today and maybe study tomorrow!'; Aytun Koyuncular who will –I hope– trace our steps and will go farther than we managed to go – thank you all my fellow nanoscience group members, it was a pleasure to be working among you! *Cheedem* Özkan, Emel Kilit, Özge Amutkan – my Chai Tea Latte fellows, we'll meet again on some sunny day, in another Starbucks, this I'm sure of. Ruzin Ağanoğlu, Bayram Tekin, Sezen Sekmen, Yasemin Gürcan – thank you for the physics, the laughs and the friendship. A big bucket of thanks goes to Dr. Hande Üstünel for the unofficial DFT classes and the update on the scientific world, to Dr. Ayşe Küçükarslan for being the ideal roommate one can hope for and to Selma Şenozan for always being there in times of need. Serkan Polad, Özlem İpek, Turan Birol, Cesim Dumlu, Sefa Aslan, Emre Aziz, Ziya Kalay, Egemen Özcan, Volkan Fırat – my bright 'students' who became my colleagues, thank you for your kind friendship and for our continued correspondence even after your graduation and 'flight from the nest'. A great deal of thanks goes to Prof. Dr. Demet Gülen and Prof. Dr. Mehmet Zeyrek for their faith and confidence in me when I applied for the post doc positions. To our beautiful department where best of my days passed: all my gratitude to Prof. Dr.

Sinan Bilikmen, the chairperson, to the vice chairpersons Prof. Dr. İbrahim Günal, Prof. Dr. Meltem Serin Zeyrek. I am indebted to the kindness and concern I received from Prof. Dr. Çiğdem Erçelebi and Prof. Dr. Gürsevil Turan. I would not most probably be here at this time, writing these acknowledgements if, in the first hand it was not for dear Mrs. Sevim Aygar, the departmental secretary accepting my late application and having it evaluated. Mrs. Zeynep Eke and Mrs. Gülşen Özdemir Parlak with now retired Mrs. Sultan Köksal and recently joined Mr. Süleyman Taş, it was a pleasure knowing you and thank you for everything you have done concerning the otherwise time-consuming, depressing departmental and bureaucratic tasks.

Last, but not least, I would like to thank my wife Bengü, my daughter Ece, my mother Dilek Kurtar and the rest of my dear family for their ever growing support that enabled me to go on even at the worst days.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Starting with Iijima's discovery of carbon nanotubes in 1991 [1], nanostructures begin to play a very important role in the construction of the state-of-the-art devices [2-6], applicable everyday technology [7], and they are one of the most promising medium in the long sought alternative energy solutions [8-13].

Although they are still being examined and experimented upon, it is accepted that the day they will be items of manufacture are not very far [14,15]. But, until that date, more heavy research keeps on being conducted worldwide. Because of the high costs and limitation of the current measuring devices, a dominant percent of the research includes computer simulations. Even for hard-wired experiments, there is almost always an initial simulation process which gives idea to formulate the outcome expectations.

In this work, we present a new and efficient method to model some specific structures such as tori, helix, n-toothed gears and toothed canals as well as a way to merge any number of zigzag or armchair type nanotubes. Using the outcome structures of this work, one will be able to construct model systems to use in any molecular simulation processes. We also included the results of some of our simulations containing of these systems.

We have focused on systems that are of carbon graphite derivatives but the method can easily be implemented upon other hcp packed atoms. The reason for us to choose carbon is the widely tested, verified and reported carbon system experiments. Recently other elements doped carbon systems [16-18] and even gold nanotubes [19] have been reported and as long as they obey similar bonding rules, they can be also employed by the present method.

The thesis is divided in four main sections. In the second section, we discuss the theoretical methods to generate various structures along with suggestions for implementation of these methods into programming code with the help of algorithms. These methods are divided into three subsections being the force fields and potential functions specific to the systems investigated being the MM+ and the Tersoff potential, then a brief introduction is given on optimization techniques beginning with the steepest descent algorithm for tutorial purposes and continuing with the more complex conjugate gradient methods such as the Fletcher-Reeves and the Polak-Ribiere optimization procedures, third subsection of the ‘methods’ section is devoted to the molecular dynamics where the ‘motor’ of the molecular dynamics simulations being the Verlet algorithm is derived along with the principals in the simulation process. The second section concludes with a simplified introduction to the electronic orbitals and their semi-empirical calculations via the Extended Hückel method. Third section is devoted to the construction methods, and algorithms if applicable, of various nanocage structures. Due to their forming differences, ‘brute-force’ cross junctions, nanotori and nanogears were dealt in separate subsections whereas similar forming structure groups such as nanohelix in similarity with nanotori or multi-junctions and toothed canals in similarity with nanotori were grouped together. In the fourth section, the stability, dynamics and electronic properties of these systems constructed via the procedures described in the previous section are investigated through simulation and the simulation results with the details of the related parameters are presented and discussed. The fifth section titled ‘Conclusion’ aims to summarize what has been done and to point out a possible future scenario.

CHAPTER 2

SIMULATION AND COMPUTATIONAL TOOLS, METHODS AND ALGORITHMS

2.1 Force Fields and Potential Functions

The primary aim of molecular mechanics (MM) is to predict the possible formation of an unknown molecule by finding the positions that yield the minimum energy arising from the interactions between its atoms, thus obtaining the most stable form which generally corresponds to its natural geometry. Molecular dynamics (MD) also need a method to calculate the impulses generated by the interactions between the atoms so that it can proceed and locate the components in advanced times. In order to calculate the energies and the forces, we must know the formulated interaction equations and the different parameter sets for different types of atoms including different hybridizations and configurations of the same atoms. These potentials are given by the so called force fields and potential functions. In general, bond information considering force fields are used for molecular mechanics calculations, whereas for molecular dynamics, those potentials which enable bond formations and bond breaking are employed - as MM try to find the most stable structure whereas MD seek a realistic simulation of a natural process.

A modern force field consists of various interaction components such as bond-stretching, angle-bending, out-of-plane bending, dihedral motions, non-bonded interactions, Coulomb interactions, and such. Different types of force fields have been developed for different atom types and different environments. Among these force fields, to summarize the most popular ones, Dreiding [20] is parameterized for

H, C, N, O, P, S, F, Cl, Br and I; MM3 developed by Allinger [21,22] and MM+ by Hypercube [23] deals mainly with hydrocarbons with MM+ having a wildcard guessing model, Amber [24] and OPLS [25] (Optimized Potentials for Liquid Simulations) specializes on amino acids and proteins and the Johnson [26] force field is specific to pure elements Fe, W and V.

2.1.1 MM+ Force Field Components

This subsection is mainly based on the manual of Hypercube's HyperChem software manual [23].

2.1.1.1 Bond Stretching

A bond in a molecule can be modeled as a spring connecting two masses, oscillating towards the equilibrium distance. MM+ formulates this behavior as:

$$E_{bond} = 143.88 \sum_{bonds} \frac{1}{2} K_r \left(r - r_0\right)^2 \left[1 + \text{switch}\left(r - r_0, -\frac{1}{3}CS, -\frac{4}{3}CS\right) CS\left(r - r_0\right)\right] \quad (2.1)$$

with *CS* set to -2.0 and the cut-off functions *switch* is defined as

$$\begin{aligned} &\text{switch}\left(x,a,b\right) = 1 && x \le a \\ &\text{switch}\left(x,a,b\right) = 0 && x \ge b \\ &\text{switch}\left(x,a,b\right) = \frac{\left(b-x\right)^2\left(b+2x-3a\right)}{\left(b-a\right)^3} && a < x < b. \end{aligned} \quad (2.2)$$

This cut-off function maintains a swift transition between the omitted and active regions. It also fixes the original MM2 potential's repulsive behavior when $r - r_0 > \frac{2}{3}CS$. This potential is very different than the classical ones because of the cubic stretch term. Although this difference separates the model from the standard spring model, nonetheless, it behaves more accurate when experimental results are considered. The unit of the energy is kcal/mol where K_r is the empirical 'spring' coefficient for the atoms in questions.

2.1.1.2 Bond Dipoles

In MM+, the electrostatic contributions come from bond dipole moments associated with polar bonds. The center of the dipole is defined to be the midpoint of the bond and for two dipoles μ_i and μ_j separated by R_{ij} as shown in Figure 1, their dipole interaction energy is given as:

$$E_{dipole} = 14.39418\varepsilon \sum_{i,j \in polar\,bonds} \mu_i \mu_j \left(\frac{\cos\chi - 3\cos\alpha_i \cos\alpha_j}{R_{ij}^3} \right) \quad (2.3)$$

where ε is the dielectric constant of the medium, χ is the angle between the two dipole vectors and α_i and α_j are the angles that the two dipole vectors make with the R_{ij} vector. The constant 14.39418 is used to convert the ergs/mol energy unit to kcal/mol.

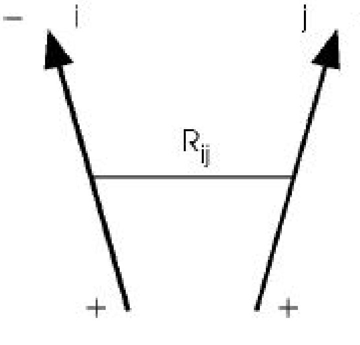


Figure 1: Bond dipoles

2.1.1.3 Angle Bending

The restoring energy for the angles disturbed from their equilibrium values is calculated by means of the even function

$$E_{bond\,angle} = 0.043828 \sum_i \frac{1}{2} K_\theta \left(\theta_i - \theta_0\right)^2 \left[1 + SF\left(\theta_i - \theta_0\right)^4\right] \quad (2.4)$$

The SF term multiplying the sextic bending term is equal to 7.0×10^{-8} and the constraint 0.043828 converts the unit from millidyne-Å per radian^2 to kcal/mol per degree^2. K_θ is the force constant, θ_0 is the equilibrium bond angle and θ_i is the bond angle.

2.1.1.4 Bond Stretch and the Angle Bending Cross Term

Allinger's concept of coupling the components [21,22] was a revolutionary idea of his time. By means of this correction factor, much more accurate results could be achieved. Between three bonded atoms labeled *i,j* and *k* (*k* being the center atom), the MM+ couples the stretching of the ik and jk bonds with the angle as:

$$E_{stretch-bend} = 2.51118 \sum_{angles} K_{sb} \left(\theta - \theta_0\right)_{ikj} \left[\left(r - r_0\right)_{ik} + \left(r - r_0\right)_{jk}\right] \quad (2.5)$$

The constant 2.51118 converts the unit from millidynes per radian to kcal per degree.

2.1.1.5 Out-of-Plane Bending

An sp^2 hybridized atom tends to be coplanar with the 3 atoms it is bonded. To model this effect, the angle bending interactions are calculated with respect to the image of the atom onto the plane its bonded atoms define, rather than its actual position.

This fact is illustrated in Figure 2: Rather calculating the angle bending energies for the angles AXB, AXC and BXC, we use the angles AYB, AYC and BYC.

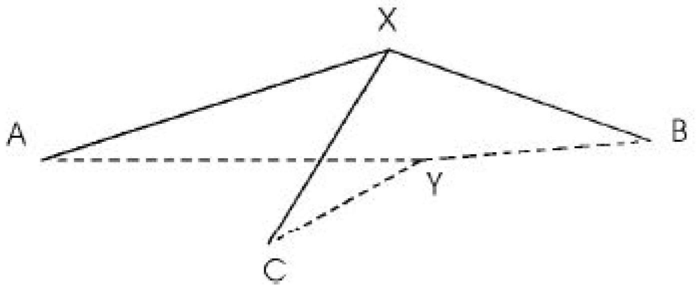


Figure 2: Out-of-Plane Bending

2.1.1.6 Dihedral Motions

The torsion that arises from the bondings of 4 atoms with 2 planes is shown in Fig. 3. The interaction energy for the dihedral motions is given by the formula:

$$E_{dihedrals} = \sum_i \frac{V_1}{2}\left(1+\cos\phi_i\right) + \frac{V_2}{2}\left(1-\cos 2\phi_i\right) + \frac{V_3}{2}\left(1+\cos 3\phi_i\right) \quad (2.6)$$

where V_1, V_2 and V_3 are given in kcal/mol units.

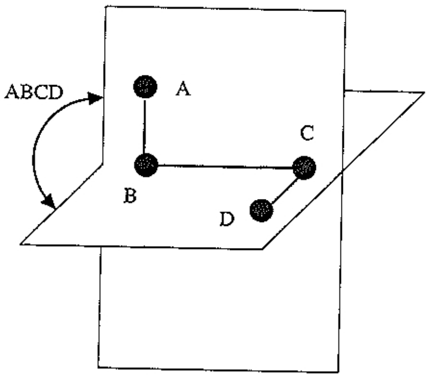


Figure 3: Dihedral motions

2.1.1.7 van der Waals

The MM+ interactions do not use a Lennard Jones potential as is the custom, but combine an exponential repulsion with an attractive R^{-6} dispersion interaction. The basic parameters are the van der Waals radii r_i^* for the atoms and a hardness parameter ε_i that determines the depth of the attractive well and the relative difficulty to push the atoms close together. The parameters for a nonbonded pair *i,j* is then calculated as:

$$r_{ij}^* = r_i^* + r_j^* \tag{2.7}$$

$$\varepsilon_{ij} = \sqrt{\varepsilon_i \varepsilon_j} \tag{2.8}$$

and the van der Waals interaction is given by:

$$E_{vdW} = \sum_{ij \in vdW} \varepsilon_{ij} \left[2.9 \times 10^5 \exp\left(-12.5\rho_{ij}\right) - 2.25\rho_{ij}{}^{-6} \right] \tag{2.9}$$

where

$$\rho_{ij} = \frac{R_{ij}}{r_{ij}^{*}}. \tag{2.10}$$

At short distances, $\left(R_{ij} \leq 3.311\text{Å}\right)$ the above expression is replaced by:

$$E_{vdW} = 336.176 \sum_{ij \in vdW} \varepsilon_{ij} \rho_{ij}^{-2}. \tag{2.11}$$

In this section, the unit of the calculated energies are given in kcal/mol units, distances are in Å and angles are in radian.

2.1.2 The Tersoff Potential

The Tersoff potential energy function [27] developed for carbon [28] employs two and three body interactions and the total interaction energy is taken to be the sum of the total two-body and the three-body interactions:

$$\Phi = \phi_2 + \phi_3 \tag{2.12}$$

Total two-body and three-body energies are given as:

$$\phi_2 = A \sum_{i<j}^{N} U_{ij}^{(1)} \tag{2.13}$$

$$\phi_3 = -B \sum_{i<j}^{N} U_{ij}^{(2)} \left[1 + \beta^n \left(\sum_{k \neq i,j}^{N} W_{ijk} \right)^n \right]^{-1/2n} \tag{2.14}$$

where U_{ij} and W_{ijk} represent the two-body and three-body interactions, respectively and defined by:

$$U_{ij}^{(1)} = f_c\left(r_{ij}\right) \exp\left(-\lambda_1 r_{ij}\right) \tag{2.15}$$

$$U_{ij}^{(2)} = f_c\left(r_{ij}\right) \exp\left(-\lambda_2 r_{ij}\right) \tag{2.16}$$

$$W_{ijk} = f_c\left(r_{ik}\right) g\left(\theta_{ijk}\right) \tag{2.17}$$

where

$$g\left(\theta_{ijk}\right)=1+\frac{c^2}{d^2}-\frac{c^2}{d^2+\left(h-\cos\theta_{ijk}\right)^2} \tag{2.18}$$

and

$$f_c\left(r\right)=\begin{cases} 1, & r<R-D \\ \dfrac{1}{2}-\dfrac{1}{2}\sin\left[\dfrac{\pi}{2}\left(r-R\right)/D\right], & R-D<r<R+d \\ 0, & r>R+D \end{cases} \tag{2.19}$$

with the parameters set for the carbon atom are given as follows:

$$\begin{array}{ll} A=1393.6eV & B=346.74eV \\ \lambda_1=3.4879\text{Å}^{-1} & \lambda_2=2.2119\text{Å}^{-1} \\ \beta=1.5724\times10^{-7} & n=0.72751 \\ c=38049 & d=4.3484 \\ h=-0.57058 & R=1.95\text{Å} \\ \text{D=0.15Å} & \end{array} \tag{2.20}$$

2.2 Molecular Mechanics and Energy Optimization

Molecular mechanics try to find the optimum configurations of the atoms of a system that minimizes the interaction energies defined via a force field. In minimizing the system energy, various methods can be employed. The two optimization methods used in this thesis study are the steepest descent and the Polak-Ribiere optimization. While Polak-Ribiere optimization is faster when the system is near to the optimum configuration, the steepest descent method is favored in the first optimization steps, converging the system to a configuration that is near to the optimized one. It is advisable to start with the steepest descent method and then switching to the Polak-Ribiere method. While both methods can stuck in local minima instead of the global minima, this problem can be checked and improved using various other methods to increase the probability that the minima achieved is a global one.

2.2.1 The Steepest Descent Algorithm

The steepest descent method uses the gradient of the potential in approaching the minima. In a system of *N* atoms, we have *3N* components for the positions. If we are to calculate the gradient of the potential with respect to these *3N* positions, and relocate the positions towards this gradient, we arrive at a better configuration. Modeling the force field derived from the gradient of the potential as a landscape with hills where the potential is high and valleys where it is low, we can think of the configuration as a marble released. It will move towards where the field is weaker (another way of saying, it will be affected more effectively with the strong field areas causing it to move, while strolling more swiftly and gently in the weaker force fields).

Having calculated the gradient, one can say that the optimum lies in the direction of the strongest (i.e. highest) gradient component. To formulate, we can denote this direction **g** ($f(\mathbf{r})$ being the potential function) by:

$$\mathbf{g}_i = \frac{-\nabla f(\mathbf{r}_i)}{|\nabla f(\mathbf{r}_i)|} \tag{2.21}$$

and the improved position vector is derived as follows:

$$\mathbf{r}_{i+1} = \mathbf{r}_i + \lambda_i \mathbf{g}_i \tag{2.22}$$

where λ is defined as the step size, as the name implies, the parameter to control how much relocation in this direction is going to be realized. One can use small but efficient algorithms like the Golden Section Search [29] or the Armijo [29] algorithm or utilize some basic algorithm [23] as simple as:

1. Choose an initial step size.
2. Take a step in the direction defined by (2.21) and calculate the configuration.
3. If the arrived configuration generates a higher interaction energy, return to the configuration at the beginning of the 2nd step, take the new step size to be the half of the current step size and proceed to the 2nd step. If the arrived configuration generates a lower interaction energy then take the new step size as 1.2 times the current step size and proceed to the 2nd step.

Although the steepest descent algorithm is fast and easy to employ, the main drawback is, as mentioned earlier in this section, its convergence towards a minimum is very slow. This can be seen in Figure 4 [23].

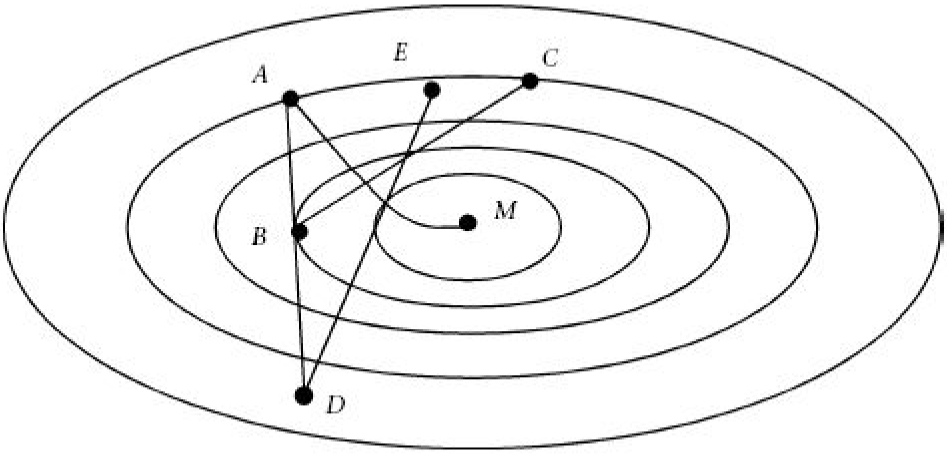


Figure 4: The Steepest Descent Method

Let's suppose the system has a minimum at point *M* and our system is initially at point *A*. For a chosen step size, the algorithm finds point *B*, and then takes on from there to some point along the *B-C* path. With a larger initial step size, we can reach point *D* from *A*, but the next step arrives us to a point on the *D-E* path which diverges from the minimum. It will take many steps to arrive to point *M*, the algorithm jumping around this point. One can also use infinitesimal step sizes and can trace the path shown in the figure from point *A* to *M* but this will cost in calculation time. So, steepest descent method should be used at the beginning of the optimization and then must be switched to a more efficient method.

2.2.2 Conjugate Gradient Methods with the Fletcher-Reeves and Polak-Ribiere procedures

This subsection summarizes the concepts as expressed in References [23,29-31].

The main difference of the conjugate gradient methods from the steepest descent method is that, it 'remembers' the previous direction and searches for a better alternative based on this one. It first begins a rough line search in the given direction $\mathbf{h}_i$ for the optimum step size. Then, calculates the gradient at this point and derive the new conjugate direction $\mathbf{h}_{i+1}$ using the formula

$$\mathbf{h}_{i+1} = \mathbf{g}_{i+1} + \gamma_{i+1}\mathbf{h}_i. \tag{2.23}$$

The coefficient γ is described by various procedures. Of these procedures, two frequently used in energy optimizations are the Fletcher-Reeves and the Polak-Ribiere procedures. Fletcher-Reeves procedure calculates the coefficient γ as

$$\gamma_{i+1} = \frac{\mathbf{g}_{i+1} \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i} \tag{2.24}$$

while the Polak-Ribiere procedure calculates the coefficient γ as

$$\gamma_{i+1} = \frac{\left(\mathbf{g}_{i+1} - \mathbf{g}_i\right) \cdot \mathbf{g}_{i+1}}{\mathbf{g}_i \cdot \mathbf{g}_i}. \tag{2.25}$$

For quadratic functions, Equations (2.24) and (2.25) are identical [31] but for non-quadratic functions, Polak-Ribiere has been found superior [23,29]. Also while for Fletcher-Reeves, the conjugate direction must be reset to that of the steepest descent every *3N* iterations or whenever the energy between the cycles increase, Polak-Ribiere is reset only when the energy between the cycles increase. Also according to Polak himself [30], the reason for his procedure is superior may be the fact that Polak-Ribiere method satisfies the assumptions of the Polak-Sargent-Sebastian theorem [32], while the Fletcher-Reeves method does not.

The main advantage of the conjugant gradient methods is their ability to almost directly advance to the minima. This fact is illustrated in Figure 5 [23] based on the steepest descent's search in Figure 4.

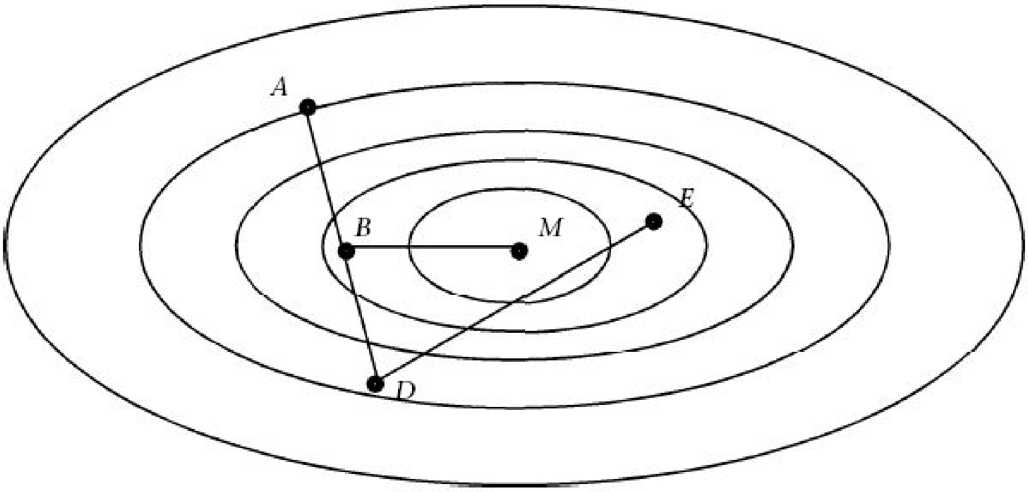


Figure 5: The Conjugate Gradient Method

In Figure 5, the search for the minimum point *M* is initiated at point *A*. After deducing the *A-D* search path, we obtain the optimal point *B* by means of the line search procedure. Even if point *B* can not be achieved and point *D* is used for further search, the algorithm comes up with point *E*, nearer to the minimum point *M*, using the previous step's information in contrast to that of the steepest descent method. Also, for quadratic functions, this method guarantees that the convergence to the minimum will be achieved in no more steps than the total number of position components [31]. This fact is also the main disadvantage of this method. The number of cycles will be linearly proportional to the number of atoms (*N*) and the time per cycle will be proportional to (N^2).

2.3 Molecular Dynamics

2.3.1 The Verlet Algorithm

In elementary classical physics, Euler's equations [33] are thought as basic equations relating the velocity, acceleration and the trajectories as:

$$a(t) = \frac{dv(t)}{dt} \tag{2.26}$$

$$v(t) = \frac{dx(t)}{dt} \tag{2.27}$$

integrating these two equations yield:

$$v(t_0 + \Delta t) = v(t_0) + \Delta t.a(t_0) \tag{2.28}$$

$$x(t_0 + \Delta t) = x(t_0) + \Delta t.v(t_0) \tag{2.29}$$

but these simple equations are handicapped when the magnitude of the error arises. Considering the Taylor expansion of the trajectory:

$$x(t_0 + \Delta t) = x(t_0) + \Delta t.v(t_0) + \tfrac{1}{2}\Delta t^2 a(t_0) + O(\Delta t^3) \tag{2.30}$$

and subtracting this equation from (2.29) yields an error magnitude of Δt^3

$$-\tfrac{1}{2}\Delta t^2 a(t_0) + O(\Delta t^3). \tag{2.31}$$

For small Δt, the dominant error per step is proportional to the second power of Δt which can be quite large. For this reason, Verlet algorithm is preferred for which, the error part in position is of the order 4.

Verlet algorithm [34,35] employs the previous and the present time steps' trajectory information to calculate the next time step's. Using Taylor expansion:

$$x(t_0+\Delta t) = x(t_0) + \Delta t.v(t_0) + \frac{1}{2}\Delta t^2 a(t_0) + \frac{1}{6}\Delta t^3 \frac{d^3x}{dt^3} + O(\Delta t^4) \tag{2.32}$$

$$x(t_0-\Delta t) = x(t_0) - \Delta t.v(t_0) + \frac{1}{2}\Delta t^2 a(t_0) - \frac{1}{6}\Delta t^3 \frac{d^3x}{dt^3} + O(\Delta t^4) \tag{2.33}$$

adding these two equations and arranging the result yields:

$$x(t_0+\Delta t) = 2x(t_0) - x(t_0-\Delta t) + \Delta t^2 a(t_0) + O(\Delta t^4) \tag{2.34}$$

thus eliminating the third-order term from the equation. As can be seen, the velocity information is not needed when computing the position but in order to control the stability of the ongoing simulation, energy conservation must be checked and for this we need the velocity component of the atoms. Subtracting equation (2.33) from equation (2.32) and dividing the result by the time step Δt gives us the velocity:

$$v(t_0) = \frac{x(t_0+\Delta t) - x(t_0-\Delta t)}{2\Delta t} + O(\Delta t^2) \tag{2.35}$$

Although the velocity equation's error magnitude is of the second order, the systematical calculations of the positions employed in deducing the velocities keeps this error from propagating, that is, resetting it in every step.

The problem with Eq. (2.35) lies in the generation order. We can calculate the velocity of time t_0 only after we have calculated the position at $t_0 + \Delta t$. Also, the velocity is obtained from position information thus increasing the error propagation by using already derived magnitudes. To overcome the second problem, a procedure called the "leap-frog algorithm" is implemented in which, the velocity is calculated at half time steps using the acceleration information at t_0 and the previous velocity at $\left(t_0 - \frac{1}{2}\Delta t\right)$:

$$v\left(t_0 + \frac{1}{2}\Delta t\right) = v\left(t_0 - \frac{1}{2}\Delta t\right) + a\left(t_0\right)\Delta t \tag{2.36}$$

and since we have calculated the velocity explicitly, we can use that information whilst calculating the position:

$$x\left(t_0 + \Delta t\right) = x\left(t_0\right) + v\left(t_0 + \frac{1}{2}\Delta t\right)\Delta t. \tag{2.37}$$

As can be seen, in order to calculate the position, one needs the velocity at a time $\frac{\Delta t}{2}$, and for the velocity, the acceleration obtained from the force acting for the position information at time t_0. This need/feed relation can be diagrammed as in Figure 6 [36].

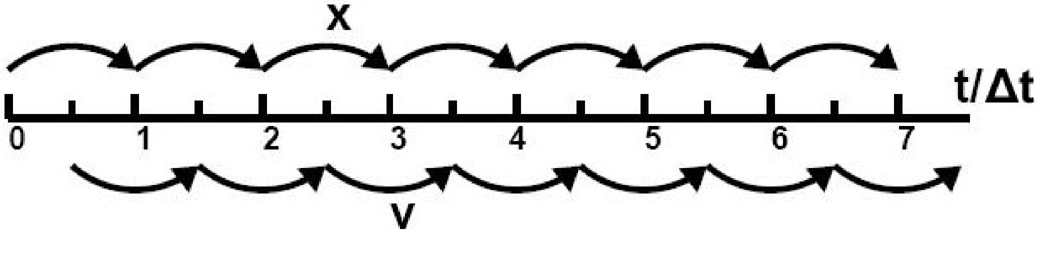


Figure 6: Need/feed relation of the leap-frog algorithm.

The instantaneous velocity at a time *t* can be approximated by the formula:

$$v(t) = \frac{v\left(t+\frac{\Delta t}{2}\right)+v\left(t-\frac{\Delta t}{2}\right)}{2}. \tag{2.38}$$

To achieve the position and the velocity information at the same time-step, an implementation of the Verlet algorithm called the "Velocity Verlet Algorithm" is devised. In this scheme, the position, velocity and the accelerations at time $t_0 + \Delta t$ are obtained from the same quantities at time *t* using the formulas:

$$x(t_0+\Delta t) = x(t_0)+v(t_0)\Delta t+\frac{1}{2}a(t_0)\Delta t^2 \tag{2.39}$$

$$v\left(t_0+\tfrac{\Delta t}{2}\right) = v(t_0)+\frac{1}{2}a(t_0)\Delta t \tag{2.40}$$

$$a(t_0+\Delta t) = -\frac{1}{m}\nabla V\Big|_{x(t_0+\Delta t)} \tag{2.41}$$

$$v(t_0+\Delta t) = v\left(t_0+\tfrac{\Delta t}{2}\right)+\frac{1}{2}a(t_0+\Delta t)\Delta t \tag{2.42}$$

This algorithm uses *9N* memory locations to save the *3N* position, velocity and acceleration information but now we do not need to simultaneously store quantity's value at two different times. Two good reviews on the subject and other methods can be found in the works of Ercolessi [37] and Young [36].

2.3.2 Simulation Process

In the simulation, canonical ensemble molecular dynamics was used. Within this ensemble the number of atoms *N*, the volume *V* and the temperature *T* is taken to be constants and velocities are scaled with respect to *T*, to ensure that the total kinetic energy and hence the temperature is constant (isokinetic MD [38]). Also, the initial velocities are distributed using the Maxwell distribution [39]:

$$f\left(v_{x,y,z}\right)dv_{x,y,z}=\frac{N\left(v_{x,y,z}\right)dv_{x,y,z}}{N}=\sqrt{\frac{m}{2\pi kT}}\exp\left(-\frac{mv_{x,y,z}^{2}}{2kT}\right)dv_{x,y,z} \qquad (2.43)$$

where $f\left(v_{x,y,z}\right)$ is the fraction of atoms that have the velocity v for their *x, y* or *z* components, accordingly, *k* is the Boltzmann constant and $N\left(v_{x,y,z}\right)$ is the number of atoms with velocity v for their *x, y* or *z* components accordingly. The trajectories are calculated using the Velocity Verlet Algorithm and from these trajectories, average system observables are derived. All these procedures were calculated using Erkoç's *Molecular Dynamics Program for Cluster Simulations (md-tpc-pbc.f)* [40]. Necessary modifications for the systems considered were implemented into the original code with the given consent of Erkoç himself.

2.4 Electronic Structure Calculations

The content of this section is summarized from Reference [41].

In order to solve the electronic Schrödinger equation

$$\hat{H}_e\Psi_e=\left(\hat{H}_{k,e}+\hat{H}_{p,e-e}+\hat{H}_{p,n-e}\right)\Psi_e=E_e\Psi_e \qquad (2.44)$$

where *k* implies the kinetic energy, *p* the potential energy, *n* the nucleus and *e* the electrons. Explicitly written form of the electronic Schrödinger equation is given by

$$\left[-\sum_{i=1}^{N}\frac{\hbar^2}{2m_e}\nabla_{\vec{r}_i}^2+\frac{1}{2}\sum_{i_1\neq i_2=1}^{N}\frac{1}{4\pi\varepsilon_0}\frac{e^2}{\left|\vec{r}_{i_1}-\vec{r}_{i_2}\right|}-\sum_{k=1}^{M}\sum_{i=1}^{N}\frac{1}{4\pi\varepsilon_0}\frac{Z_k e^2}{\left|\vec{R}_k-\vec{r}_i\right|}\right]\Psi_e=E_e\Psi_e. \qquad (2.45)$$

Some approximations and generalizations are usually applied. Among these generalizations, the modification of pure hypothetical wave functions is mostly done as writing them in combinations of some base functions, usually defined by orthonormal sets.

2.4.1 Electronic orbitals

Depending on the system’s complexity and symmetry conditions, the basis sets that will be used as the components of the wave function (or initial guess for the self consistent field type iterations) are chosen. Among these are the Slater type orbitals which were devised to include anti-symmetric properties and the Gaussian type orbitals that were developed as a means of simplifying the Slater type orbitals but for which the matrix elements can be calculated relatively easy are the most commonly used orbitals. Slater type orbitals are generally given in the form

$$\chi\left(\vec{r}\right)=\chi_{\vec{R},\zeta,n,l,m}\left(\vec{r}\right)=\frac{\left(2\zeta\right)^{n+1/2}}{\left(2n!\right)^{1/2}}r^{n-1}e^{-\zeta r}Y_{lm}\left(\theta,\phi\right) \qquad (2.46)$$

whereas the Gaussian type orbitals are defined as

$$\chi\left(\vec{r}\right)=\chi_{\vec{R},\alpha,n,l,m}\left(\vec{r}\right)=2^{n+1}\frac{\alpha^{(2n+1)/4}}{\left[(2n-1)!!\right]^{1/2}\left(2\pi\right)^{4}}r^{n-1}e^{-\alpha r^{2}}Y_{lm}\left(\theta,\phi\right). \qquad (2.47)$$

The main difference of Eqn. (2.47) from Eqn. (2.46) is the argument of the exponential function containing the square (even) form of *r* instead of a linear (odd) form of *r*. Due to this fact, all Gaussians have a flat tangent at the side of nucleus. Other than these sets, plane waves are also used (mainly in Density Functional Theory calculations).

2.4.2 The Extended Hückel semi-empirical method

The extended Hückel method developed by Hoffman [42,43]’ takes into account not only the π electrons of organic molecules as the original Hückel method [44-46] but all orbitals, with the electronic Hamiltonian operator replaced by a single effective operator of form

$$\hat{H}_e = \sum_{i=1}^{N} \hat{h}_{eff}(i). \qquad (2.48)$$

The single particle equations now reduce to

$$\hat{h}_{eff}\phi_i = \varepsilon_i \phi_i. \qquad (2.49)$$

Assuming we have selected a set of basis functions χ for representing wave functions, the Hartree-Fock-Roothaan [47] equations are defined by

$$\sum_{n=1}^{N_b} \left\langle \chi_p \middle| \hat{h}_{eff} \middle| \chi_n \right\rangle c_{nl} = \varepsilon_l \sum_{n=1}^{N_b} \left\langle \chi_p \middle| \chi_n \right\rangle c_{nl}. \qquad (2.50)$$

Normally, one needs to calculate the matrix elements $\left\langle \chi_p \middle| \hat{h}_{eff} \middle| \chi_n \right\rangle$ and the overlap matrix elements $\left\langle \chi_p \middle| \chi_n \right\rangle$ explicitly. Within the Extended Hückel approach, the approximations

$$\left\langle \chi_p \middle| \chi_n \right\rangle = \delta_{p,n} \qquad (2.51)$$

$$\left\langle \chi_p \middle| \hat{h}_{eff} \middle| \chi_p \right\rangle = -\tfrac{1}{2}\left(I_p + A_p\right) \qquad (2.52)$$

and

$$\left\langle \chi_p \middle| \hat{h}_{eff} \middle| \chi_n \right\rangle = \tfrac{1}{2}K\left[\left\langle \chi_p \middle| \hat{h}_{eff} \middle| \chi_p \right\rangle + \left\langle \chi_n \middle| \hat{h}_{eff} \middle| \chi_n \right\rangle\right] O\left(\chi_p, \chi_n\right) \qquad (2.53)$$

are evaluated with the I_p being the ionization potential of the isolated atom of the orbital χ_p, A_p the electron affinity, K an empirical constant (1.75 being a preferred value) and $O\left(\chi_p, \chi_n\right)$ being the overlap between the two functions. Once the matrix elements have been specified, the orbitals ϕ_i and their energies ε_i can be calculated.

CHAPTER 3

GENERATION OF NANOSTRUCTURES

In a geometrical system, we can talk about two kinds of curvatures: being the negative and positive curvatures. To generate curvatures in honeycombed graphite planes, we employ pentagonal and heptagonal defects in the hexagonal cells. While the pentagons deform the area towards a positive curvature, the heptagons generate negative curvature shapes as depicted in Figure 7.

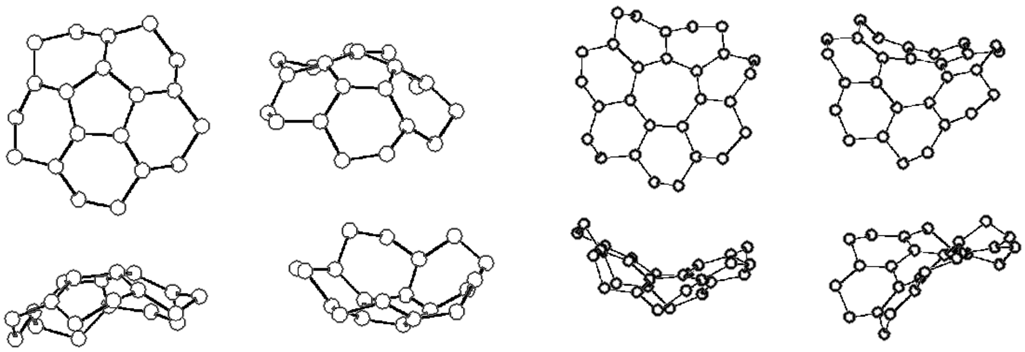


Figure 7: The curvatures generated by pentagon (left side) and heptagon (right side) defects.

3.1 Cross junction

A cross junction is formed by melding two carbon nanotubes by means of applying pressure. Using two non-interacting slabs, the tubes are pushed towards each other and the junction occurs.

3.1.1 The system

The system consists of two identical C(10,0) single walled carbon nanotubes, rotated 90º with respect to each other such that their azimuthal axes reside in the x and y directions. The two tubes are located between two carbon graphene layers. The overall system snapshot can be seen in Figure 8.

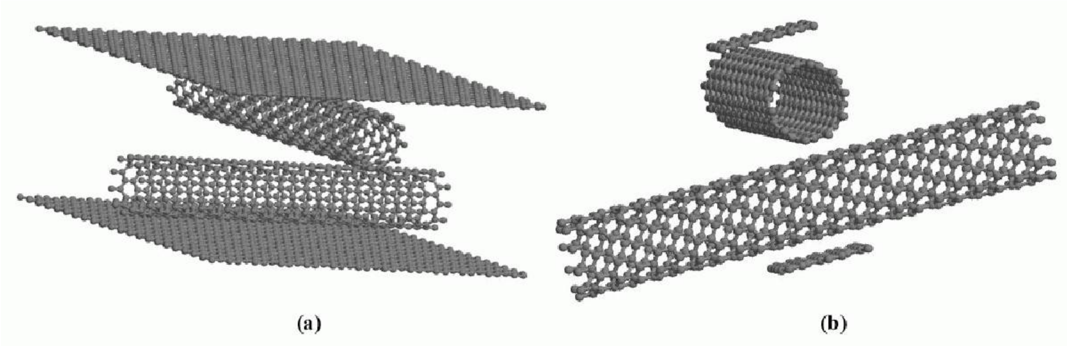


Figure 8: The initial system snapshots. Graphene layers covering (a) all the space, (b) only the junction region.

3.1.2 Simulation Method

In the simulations, the empirical many-body Tersoff Potential Energy Function developed for carbon was used. Unlike Molecular Mechanics method, Tersoff

Potential Energy Function does not employ bond information and thus, at every stage of the simulation, with the absence of a bond conservation, realistic approach was made possible in the forming and breaking of bonds.

The equations of motion of the particles were solved using the Verlet algorithm with NVT conditions. The temperature scaling is taken into account at every MD step and the temperature of the system is kept constant at 1K. The graphene layers were systematically moved closer to each other by 0.50Å in the first 8 phases where the tubes were not too close and by 0.25Å for the latter phases. In each of these phases, the system was relaxed for 30000 time steps and 1 time step corresponded to 10^{-16} seconds.

Two simulations were run, using two different sized graphene layers. In the first simulation, the graphene layers' size were taken to cover the nanotubes and with the periodic boundary conditions imposed upon the system, they acted as 2-dimensional infinite slabs.

The atoms on the graphene layers were not allowed to be affected within themselves or by the tube atoms. Their relative positions with respect to each other remained fixed so that they can be seen as rigid bodies. This can be visualized as a concrete catalyst in actual applications made up of non- or minimal interacting material such as the noble elements.

The main advantage of this method lies in the fact that the junction occurs in a very short time and it remains stable especially for mechanical strain and stress. It can be used efficiently in the situations where a net-like system is needed and the mechanical properties are more important than the electronic properties.

The handicap of this method arises from the heterogeneous and random scattering of the atoms involved in the formation of the junction. These distortions act as disruptors when dealing with currents and other conductions and render the system unpredictable in its electronic properties.

3.2 Tori, helix

3.2.1 Introduction

After the discovery of fullerenes [48] and carbon nanotubes [1], new attempts were made to obtain novel carbon based systems among which are the toroidal and helical carbon nanostructures [49-58]. Various methods have been suggested theoretically to generate toroidal structures [59,61]. A nanocage structure in the shape of a torus may be expected to have some interesting chemical and physical properties due to its multi-connectivity [62,63] and its negative curve regions [64].

Nanotori type structures have many potential prospects in view of electronic and structural applications. Addition of magnetic material inside the tori may prove to be useful in nano engineering applications whereas, presence of knee-regions may be utilized for hydrogen storage.

In our study, we have chosen the Fonseca-type nanotori [61] as our base model, which is joined by knee regions as shown in Figure 9. The curvatures of the knee regions are given by the pentagon-heptagon pairs. Furthermore, at the knee regions, the strain is reduced. The tori constructed using this method possess 5-fold symmetry and are composed of armchair and zigzag type nanotubes. The curvature is given by pentagon-heptagon pairs facing each other lying on inner and outer rims of the torus. Construction of tori directly in three dimensions is a relatively difficult task when topologically a torus can be obtained by folding a flat sheet into a cylinder and then connecting the ends of the cylinder. Consequently any tiling or covering can be drawn on a flat sheet of paper and folded to obtain a torus surface covering. Fonseca have defined a unit cell on flat sheet which could be used to construct toroidal and helical structures.

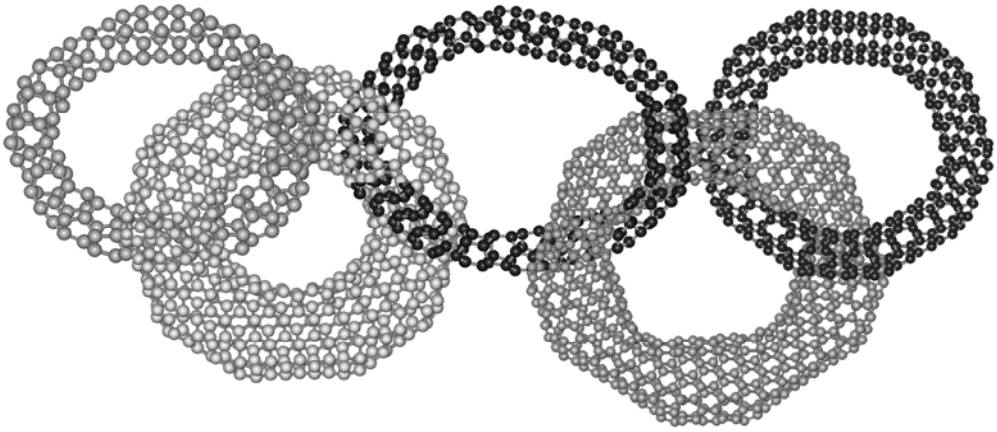


Figure 9: Fonseca-type nanotori distinguished by the knee regions.

3.2.2 Generation Algorithm for the Fundamental Cell

The algorithm for generating Fonseca typed tori and helices consists of 4 stages. In the first stage, the fundamental unit cell is constructed with its symmetrical counterpart and then the relevant bonds are defined. In the second stage, this structure is optimized using molecular mechanics methods. As a result of the optimization, one fifth of a torus segment is obtained which is used in the third stage to build up the helix or the torus depending on the atoms chosen to bond with each other. In the fourth stage, the joined segments are optimized to form the final structure.

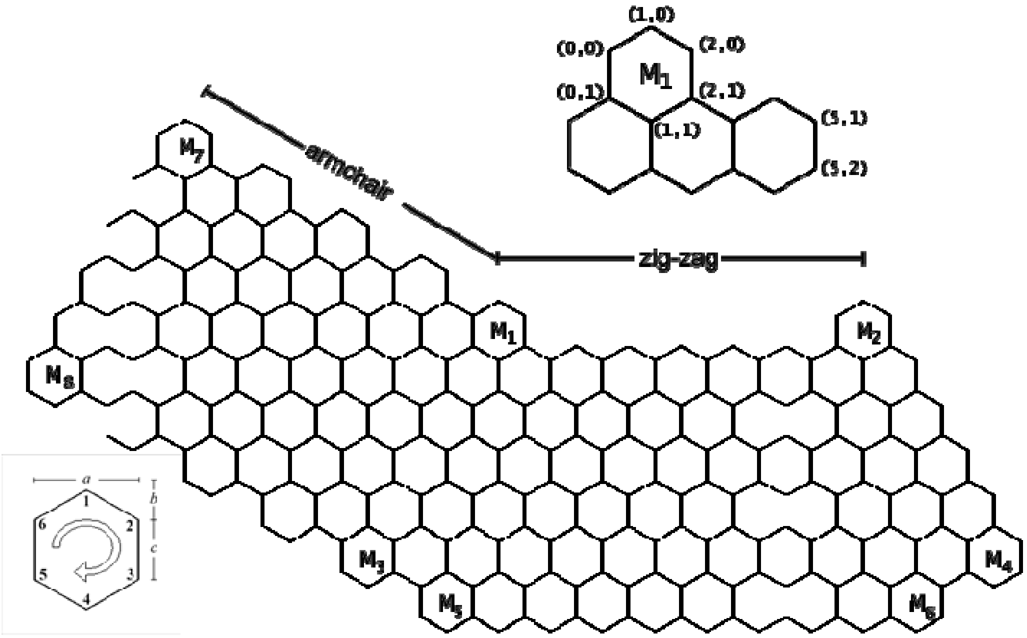


Figure 10: The fundamental unit cell and the definitions. For this particular cell A=6, B=5 and C=4. The labels of a hexagon unit are given on the bottom left

The fundamental unit cell is displayed in Figure 10. In the fundamental cell, there are no pentagons and heptagons. We represent atom locations as an entry in matrix coordinate (x', y'). These matrix coordinates are converted into real Cartesian coordinates with the help of a mapping function, $\mathbf{map}(x', y')$ prior to the output stages. At the same time, this function also generates the symmetrical counterpart which is to be connected with the original cell. The symmetry is taken with respect to $y' = B + 5$ where B determines the tube radius r. The **map** function assigns an index number to each atom it has been called for along with calculating its Cartesian coordinates. An atom with an index n will have its symmetrical counterpart labeled as $n+1$. If we designate the Cartesian coordinates by (x, y), then the primed (matrix) coordinates are converted to real coordinates by the transformations:

$$x = \frac{a}{2}x' \tag{2.54}$$

$$\begin{aligned} y = {} & \mathrm{mod}_2\left(|x'|\right)\left[cy' + \left(y' - \mathrm{mod}_2\left(|y'|\right)\right)b - (-1)^{y'}\right] \\ & + \left(-\mathrm{mod}_2\left(|x'|\right) + 1\right)\left[cy' + \left(y' - \mathrm{mod}_2\left(|y'|\right)\right)b\right] \end{aligned} \tag{2.55}$$

where a, b and c are the lengths described in Figure 10, and the $\mathrm{mod}_2(\ldots)$ is the remainder of the parameter when divided by 2. We call the single hexagon a hexagon unit and label the atoms in any hexagon unit from 1 to 6 as shown in the bottom left of Figure 10. Practical use showed that, a 2-dimensional description of one fifth of a torus segment, with the proper bonds defined, does not always successfully optimize to a proper 3-dimensional segment, because the optimization method tries to fix the system in a plane. In order to prevent this, we tilt the unit cells (the original and its symmetric counterpart) with respect to each other by assigning the z coordinate of the atoms in the original cell as

$$z = \frac{y}{1.5}. \tag{2.56}$$

We start building the fundamental unit cell by constructing the hexagon units designated with $M_1, M_2, M_3, M_4, M_5, M_6$ and M_7 as shown in Figure 10. The matrix coordinates of M_1 are displayed in the upper right corner of Figure 10. In this first part, the algorithm needs three parameters to be input. The first one, *A*, is the number of hexagons between M_1 and M_2; the second one, *B*, the number of hexagons between M_1 and M_3, including M_3; and the last one, *C*, the number of hexagons between M_1 and M_7, including M_7. These three parameters are the only parameters needed to generate a torus, although as will be explained later, an additional parameter is needed to obtain the pitch of the helix. After these parameters are obtained, M_1 and M_6 are located in matrix coordinates which determine the end points of the main (zigzag) part of the fundamental cell. To clarify the labeling system in the present algorithm, some examples can be given as the matrix coordinates of some of the atoms in the key cells M_2 through M_6 in Table 1.

Table 1: Examples for matrix coordination.

Cell Label	Atom number w.r.t. the cell	Matrix Coordinate (x', y')
M_2	1	$(2A+3,0)$
M_3	5	$(-B,B+1)$
M_4	3	$(2A+B+4,B+1)$
M_5	5	$(-B+3,B+2)$
M_6	6	$(2A+B+1,B+2)$

We continue building the fundamental cell by mapping 1st, 2nd and 6th atoms of M_1 and M_2 in real space by calling the previously defined map function with the following parameters:

$$\text{map}(x'=0\ldots 2, y'=0) \tag{2.57}$$

$$\text{map}(x'=2A+2\ldots 2A+4, y'=0) \tag{2.58}$$

where the $(\ldots)$ operator is the loop operator that iterates the value defined, starting from the number on the left of the operator to the number on the right with both of these boundaries included. The increments are evaluated by a step-size of 1. The mapped atoms by procedures (2.57) and (2.58) are shown in Figure 11 as the circled atoms. To obtain the rest of the atoms displayed in Figure 11, we use the map function in three loops where the first call (2.59) maps the atoms from top to the (a) labeled region (including this region), the second call (2.60) maps the atoms in the (b) labeled region and the third call (2.61) maps the atoms in the (c) labeled region.

```
for $y' = 1$ to $B$
    for $x' = -1$ to $2A+5$
        $\mathrm{map}(x', y')$          (2.59)
    end for
end for
```

To add the 3rd, 4th and the 5th atoms of M_3 and M_4 cells and the atoms in between we call the map function as:

$$\mathrm{map}\left(x' = -1 \ldots 2A+5, B+1\right) \tag{2.60}$$

To add the 3rd, 4th and the 5th atoms of M_5 and M_6 cells and the atoms in between we call the map function as:

$$\mathrm{map}\left(x' = 2 \ldots 2A+2, B+2\right) \tag{2.61}$$

The zigzag part of the fundamental cell is now complete as shown in Figure 11. The rest of the cell (the armchair part) is constructed as follows:

```
for $i = 0$ to $C$
        for $j = 1$ to $B + 1$
                for $x' = -(j+2)$ to $-j$
                        $y' = j - 2 - i$
                        $x_2' = x' - 3i$
                        map$(x_2', y')$
                end for
                $y' = y' + 1$        }
                map$(x', y')$        } Generation of the * atoms
        end for
        $x' = x' - 1$        }
        map$(x', y')$        }
        $x' = x' - 1$        } Generation of the # atoms                    (2.62)
        map$(x', y')$        }
end for
```

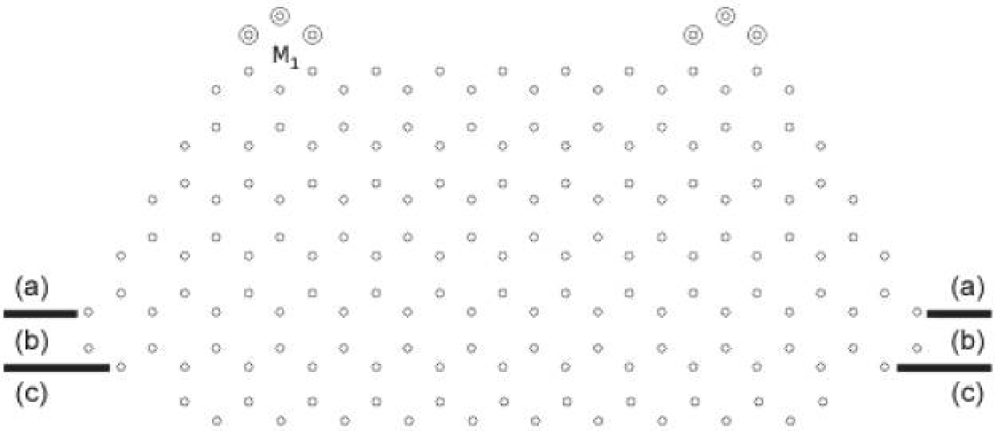


Figure 11: The evolution steps of the zigzag part of the fundamental unit cell.

Figure 12 shows the construction steps of the armchair part. In the generation of this part, first the three atoms from left to right (represented by 1 in Figure 12) is located, then the atom labeled by (*) next to the three atoms is mapped. This procedure is repeated by *B* times. When the bottom is reached, the two atoms represented by (#) is mapped. This completes one column of hexagons in the armchair part. This whole procedure is repeated by *C* times to complete all of the armchair part.

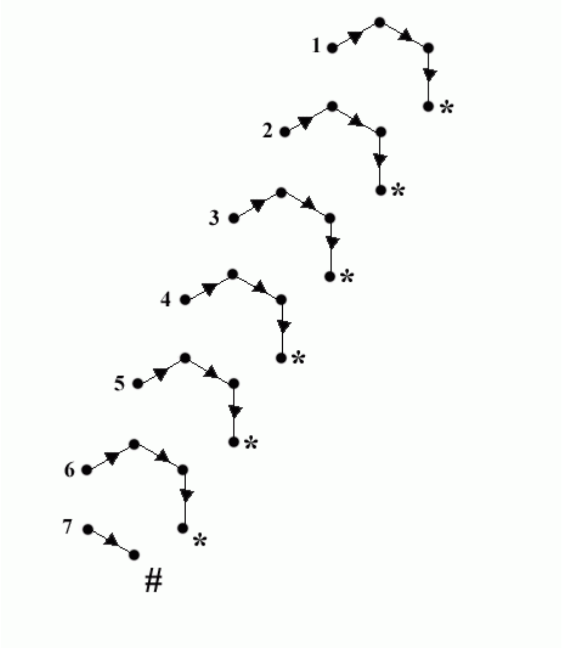


Figure 12: Steps to generate the armchair part.

At the end of this process, the atoms on both ends of the segment are recorded into two arrays atom$_1$ and atom$_2$ in specific order starting from the 2nd atom of M_2 and finishing at the 2nd atom of M_4 for atom$_1$ array and starting from the 6th atom of M_7 and finishing at the 6th atom of M_8 for the atom$_2$ array. This information will be used later when connecting the segments.

3.2.3 Bond Generation

Up to this point, the map function places the atoms such that the distance between two neighboring atoms is less than *a*, the optimum minimal second neighbor distance, so that using a simple analyzer algorithm, bonds take place between the neighboring atoms. At this point, since all of the atoms for the torus segment are placed, bonds between the unit cell and its symmetrical counterpart will be investigated.

The bond information is a crucial component in the optimization of the tori. Since we are arranging our atom formation in a semi 2-dimensional plane, this information will be used in order to obtain a 3-dimensional version. Only the atoms currently having two bonds on the upper and lower edges of the fundamental unit cell will be connected with their symmetrical counterparts to ensure the sp^2 hybridizations of the carbon atoms [65]. Some of the atoms will be connected directly and some of them will be connected to their counterparts with a new atom placed between them at the symmetry axis to conserve the graphite tiling of hexagons with the observed defects of pentagons and heptagons to generate the curvatures. The distribution of these types of bonds is made as follows:

The atoms located between the 4th atom of M_5 and the 4th atom of M_6, including the boundary atoms, will be directly connected to their counterparts, that is, with no additional atom between the connected atoms.

The atoms located between M_1 and M_2, having matrix coordinates in the range $(6,1)$ and $(2A-2,1)$ with boundaries included, will be also connected directly. This rule imposes a lower limit on *A* allowing only values greater than or equal to 4. There are no upper limits for *A*, *B* or *C* and B and C can take the minimum values of 0.

The bond construction for the atoms covered in the previous two paragraphs can be symbolically written as:

(For the atoms between and including the 4th atoms of M_5 and M_6)

```
for $i = -B + 4$ to $2A + B$ (with step size taken as 2)
    connect$\left(\text{atom\_id}(i, B+2), \text{atom\_id}(i, B+2)+1\right)$    (2.63)
end for
```

(For the atoms in the range $\left[(6,1),(2A-2,1)\right]$)

```
for $i = 6$ to $2A - 2$ (with step size taken as 2)
    connect$\left(\text{atom\_id}(i,1), \text{atom\_id}(i,1)+1\right)$    (2.64)
end for
```

where the function $\text{atom\_id}(x', y')$ returns the id label of the atom located at the matrix coordinate (x', y') and the function $\text{connect}(atom1, atom2)$ generates a bond between *atom₁* and *atom₂* and reports this bond to the Protein Data Bank [66] formatted output file via the "CONECT" directive [67].

Other than these, all the bonds must include a middle atom which will be placed on the symmetry axis and in between the atoms they are connecting. Except the four atoms in the edges of the unit cell given by matrix coordinates $(-B-3C, B-C+1)$, $(2A+3,0)$, $(2A+4+B, B+1)$, $(1-3C,-C)$, all the in between atoms will also be connected with their neighbor in between atoms (one neighbor connection for each). This process is maintained by the help of two functions, the onebond and twobond. Onebond function accepts the matrix coordinates of an atom as parameter and generates a bond with the given atom and its symmetrical counterpart with an in between atom. Twobond function, which consists of two onebond functions, accepts the matrix coordinates of two atoms and passes these atoms to onebond function but in addition also generates a bond between the two in between atoms of these two onebond generated bonds. The algorithm for this bonding process is given as:

```
onebond(-B-3C,B-C+1,1)
onebond(2A+3,0,-1)
onebond(2A+4+B,B+1,1)
onebond(1-3C,-C,-1)
twobond(2A+1+B,B+2,2A+3+B,B+1,1)
twobond(-B+3,B+2,-B+1,B+1,1)
for i = 0 to C-1
$x_1'$ = -B - 3(i+1) + 1
      $y_1'$ =  B – i
      $y_2'$ =  B – i + 1
      $x_2'$ = -B – 3i
      twobond($x_1'$,$y_1'$,$x_2'$,$y_2'$,1)
end for
twobond(2A+2,0,2A,1,-1)
twobond(2,0,4,1,-1)
for i = 0 to C-1
$x_1'$ =  1 – 3i
      $y_1'$ = -i
      $y_2'$ = -i - 1
      $x_2'$ = -1 – 3i
      twobond($x_1'$,$y_1'$,$x_2'$,$y_2'$,-1)
end for
```

After this bonding process, the fundamental cell is ready to for optimization. A particular fundamental unit cell is presented in Figure 13, where the various types of bonds can be seen.

After the fundamental unit cell is defined with its bonds, the system is optimized using the Polak-Ribiere conjugate gradient algorithm [29] with MM+ force field [23]. The force field includes the bond, angle, torsion, van der Waals and electrostatic forces. The optimization is performed in vacuo. The top-view of the fundamental unit cell after the optimization process is shown in Figure 14.

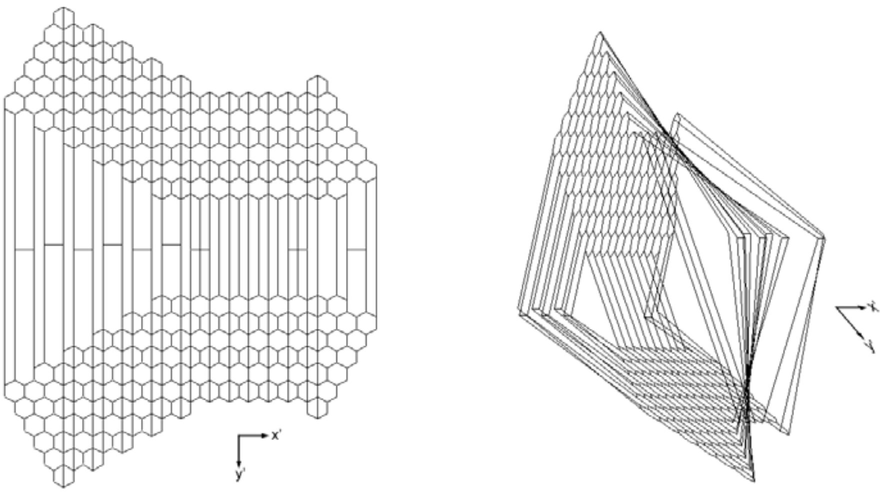


Figure 13: Completed fundamental unit cell bonded with its symmetrical counterpart: (a) top view, (b) tilted view.

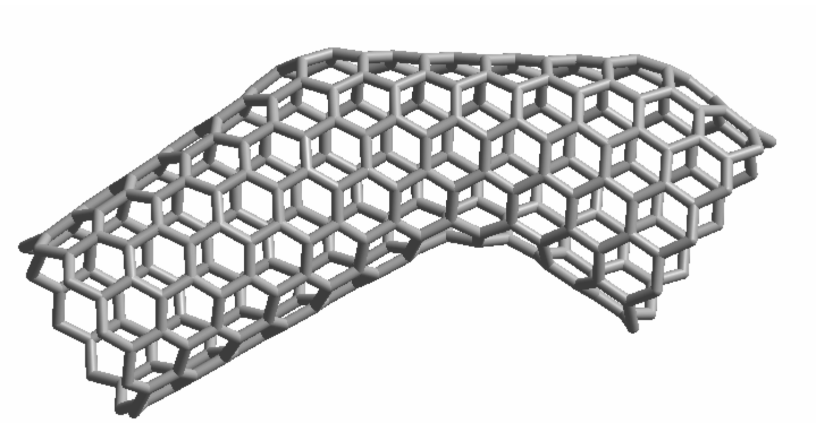


Figure 14: Optimized fundamental unit cell.

3.2.4 Building up the torus (or helix) from the segments

Now that we have a 1/5 torus segment, we are going to copy and locate it in a circular fashion for the remaining 4/5 part. For the rotation information, the approximate center of the torus should be known in order to rotate with respect to it. The torus segment is thought as its symmetry axis being the *z*-axis so, the rotation will take place in the *xy*-plane. If we take the *xy*-plane located such that it splits the segment in two with respect to the *z*-axis, we can calculate the center by finding the intersection point of the two lines joining the two atoms that lie on the *xy*-plane on both ends of the segment. This is achieved by using the function intersect(x_1,z_1,x_2,z_2,x_3,z_3,x_4,z_4) where the parameters represent the Cartesian coordinates of the 4 atoms (the first two belonging to the atom$_1$ array and the latter two belonging to the atom$_2$ array). This function calculates the intersection point (intersect$_x$, intersect$_z$) with the formulas:

$$intersect_x = \frac{\alpha x_1 + \beta x_3 + z_1 - z_3}{\alpha + \beta} \tag{2.65}$$

$$intersect_z = \alpha\left(intersect_x - x_1\right) + z_1 \tag{2.66}$$

where α and β are defined as:

$$\alpha = \frac{\left(z_1 - z_2\right)}{\left(x_2 - x_1\right)};\ \beta = \frac{\left(z_4 - z_3\right)}{\left(x_4 - x_3\right)}. \tag{2.67}$$

With the intersection point known, the segment copies can now be rotated using the rotation formulas:

$$x = \left(x_1 - intersect_x\right)\cos\left(\frac{2\pi n}{5}\right) + \left(z_1 - intersect_z\right)\sin\left(\frac{2\pi n}{5}\right) + intersect_x \tag{2.68}$$

$$z = \left(x_1 - intersect_x\right)\sin\left(\frac{2\pi n}{5}\right) + \left(z_1 - intersect_z\right)\cos\left(\frac{2\pi n}{5}\right) + intersect_z \tag{2.69}$$

where *n* denotes the number of the copied segment starting from 1 with a total of 5. After this, we have the segments placed to make up the torus, ready to be bonded with each other. Depending on the type of the bonding, we will have a torus or a helix. Prior to the bonding process, we shall define a new array allatoms[] which stores all the atoms positions. the number of rows of this array is taken to be *N* where the total number of atoms in a segment equals to *N*/5. Also we will denote the number of rows in $atom_1$ (or $atom_2$) array by *t*, that is the total number of atoms that will bond with the corresponding atoms in the subsequent segment.

3.2.4.1 Bonding algorithm for tori

As defined previously, $atom_1$ holds the list for the atoms located on one end (i.e., zigzag end) that will bond to the other segment's atoms located on the other end (i.e., armchair end). Since we have 5 segments and *t* atom pairs that will be connected, we iterate the connect function over these values as:

```
for i = 0 to t-1
  for j = 0 to 4
          connect ($atom_1$[t]+j*N/5, $atom_2$[t] + (j+1)%5 * N/5)
  end for
end for
```

This way, the jth segment's $atom_1$ atoms are bonded with the (j+1)th segment's $atom_2$ atoms while, its $atom_2$ atoms are bonded with the (j-1)th segment's $atom_1$ atoms. The remainder operator '%' is used for the continous inclusion of bonding between the original (0th) segment's $atom_2$ atoms and the last copied (4th) segment's $atom_1$ atoms.

3.2.4.2 Bonding algorithm for helix

The bonding algorithm for helix is very similar to that of the torus but, this time, the order of the bonding atoms are shifted by a value *h* which we will call the helicity. While bonding, the *i*th atom of the *j*th segment's $atom_1$ array is bonded not with the *i*th atom of the ((*j*+1)%5)th segment's $atom_2$ array but with the (*i*+*h*)th atom of the

((j+1)%5)th segment's atom$_2$. It can be immediately observed that taking *h* as 0 or any integer times *t* will result in a torus. A helix generated using this algorithm is presented in Figure 15.

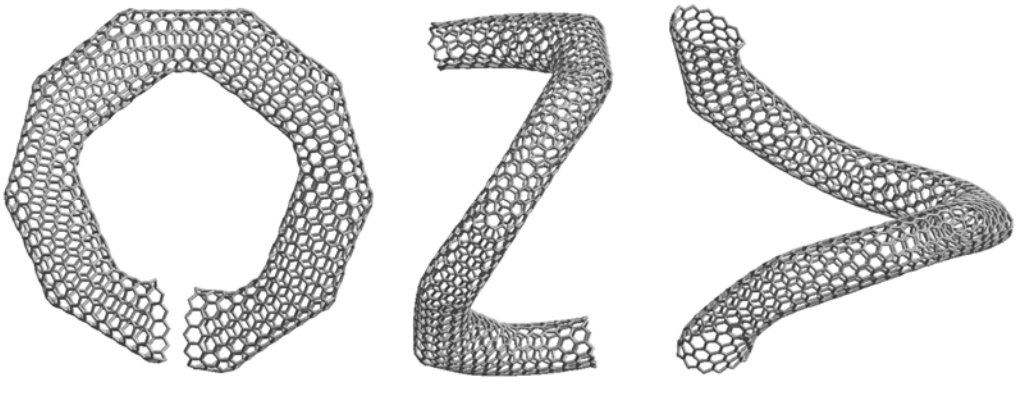


Figure 15: (a) Top, (b) and (c) side views of a completed helix. The parameters for this particular helix are: A=5, B=4, C=1 and h=1.

3.2.5 Final Optimization and the results

Now that the fundamental unit cell is generated, the bonds between its atoms are defined, it is optimized, reproduced 4 times and the copy segments are located, bonds generated between the original one and the copies themselves, the system (whether a torus or a helix) is ready to a final optimization procedure.

This final optimization takes relatively less computation time because of the alignment of the segments and the segments' already optimized configurations. It is once again done using MM+ force field with the Polak-Ribiere conjugation gradient algorithm. Because of the non-uniform shape of the typical Fonseca type tori, the inner radius *r* and outer radius R can only be defined arbitrarily. Also, any definite formula relating the input parameters and the two radii will inevitably fail because of the optimization procedures following. A choice for the definition of the radii is presented in Figure 16. The input parameters *A*, *B* and *C* are approximately related with *r* and *R* as follows:

$$R \approx c\left(B+1\right)+\sqrt{\left(intersect_x - x'\right)^2+\left(intersect_z - z'\right)^2} \qquad (2.70)$$

$$r \approx c\frac{\left(2B+4\right)}{\pi}. \qquad (2.71)$$

where *c* is defined as the distance between the 2nd and 3rd atoms in the hexagonal unit cell as shown in Figure 10, x' and z' being the x and z coordinates of the 3rd atom of M_2.

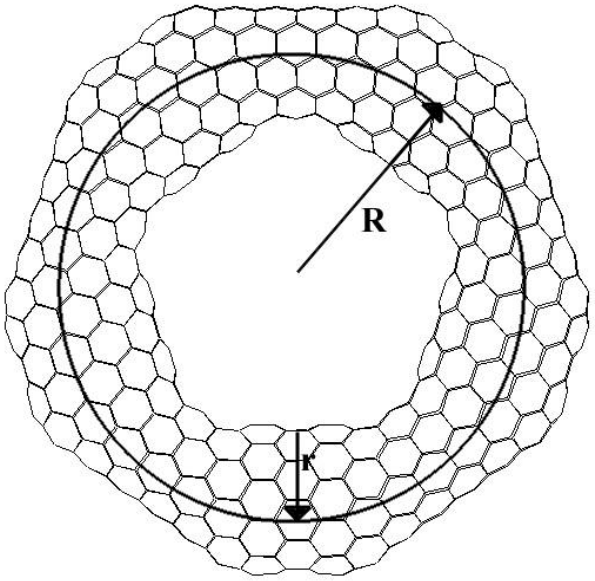


Figure 16: Geometrical definitions of r and R.

In the present algorithm to generate any toroidal or helical structure, one needs to define only the parameters *A*, *B* and *C* which determine *R* and *r*. For a fixed *R*, when the radius of the zigzag part is defined, the corresponding armchair part is automatically generated. This makes it possible to generate a torus with any inner radius *r*, as long as *r* is a regular zigzag tubule radius $\left(r \propto aB/2\pi\right)$.

3.3 Nanogears and arbitrary nanojunctions

3.3.1 Introduction

Shortly after their discovery, carbon nanotubes have found many application fields from quantum computers [68-70] to nanomotors [71,72]. Although most of these applications are, at the time, hypothetical and at most simulations, the new technologies and application techniques are rapidly being developed. Nanojunctions play a fundamental role in the construction of these systems. A method for generating nanojunctions has been developed by Zsoldos et al. [73,74]. This article, being a trailblazer in many respects, yet limited itself to two base nanotubes C(3,3) and C(6,0). In section 3.7 of their article, they give an example of a tetrahedral junction having a larger diameter than the previously mentioned ones and report that:

> (…) in order to construct tubes having larger diameters it is necessary to increase the number of hexagons in the hexagonal rings of the tubes. At the same time hexagons have to be placed among the heptagons in the heptagonal ring and among the pentagons in the pentagonal hemispherical formations too. [73]

In this work, an algorithm that can be used to develop systematical and general methods in order to produce any such junctions and gear like systems is devised. Although the generated tiles will be, in general, different than those of Zsoldos et al.'s, they can be employed in the same way as the ones developed in Ref. [73] and thus can be used to form a nanojunction made up of any desired zigzag and armchair nanotubes.

3.3.2 Algorithm, Definitions and Rules

Consider a nanojunction consisting of three nanotubes that are C(9,0), C(6,6) and C(17,0) as shown in various angles in Figure 17. This system can be thought of a fold graphite including two protuberances that will form the C(9,0) and the C(6,6) nanotubes in capped form when geometry optimized with respect to the energy as shown in Figure 18. The forming of the graphite layer such that A with A’ and B with B’ intercept each other will make up the C(17,0) nanotube.

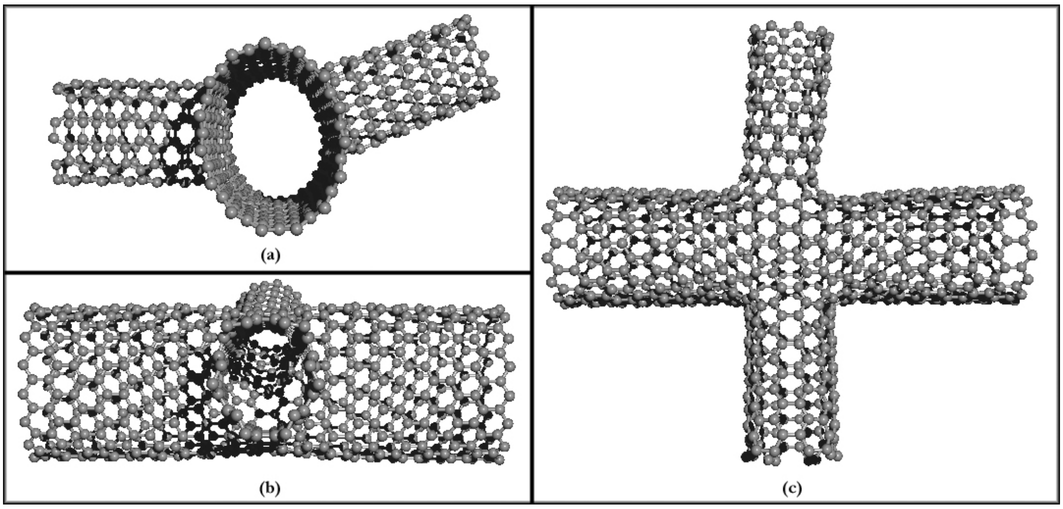


Figure 17: X- junction made of C(17,0), C(9,0) and C(6,6).

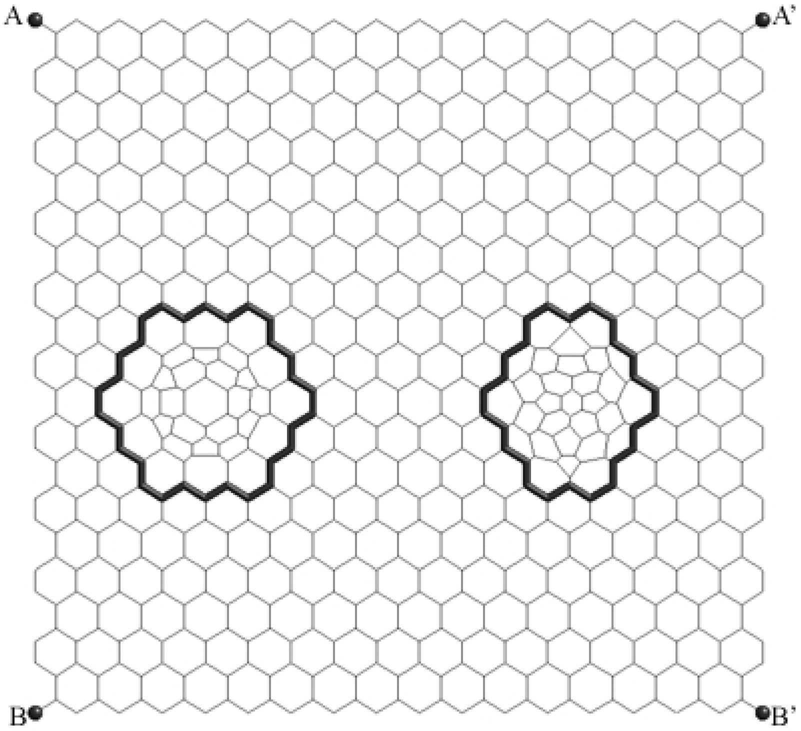


Figure 18: Unfolded C(17,0) and protuberances for C(6,6) (left) and C(9,0) (right).

The main challenge in this phase is the determination of the area that the protuberance of the nanotube to be constructed. To start with, we may assume these three rules to find the optimum dimensions of the area:

i. The 'perimeter', defined as the number of the hexagons that forms the outermost ring of the protuberance, must be equal to $n+m$ for an (n,m) nanotube (it must be kept in mind that, we are only dealing with zigzag $(n,0)$ and armchair (n,n) nanotubes) so that an (n,m) nanotube can be connected to this ring without outruling the sp^2 bonding of the carbon atoms.

ii. When going down one row in graphite, the number of hexagons the preceding row contains can be only one more or one less than the next row for the consistency of the shape.

iii. The shape of the protuberance's covered space in the graphite must be symmetrical, actually being equal to a circle in the limit so that, when a nanotube is formed from the protuberance, it will be as smooth as possible. If we make an analogy with a mapping of the hexagonal graphite to Cartesian space, the hexagonal space will become a rectangle, its diameter being proportional to *n+m*. As it is known from differential geometry, the maximal symmetry of a rectangle is achieved when the sides are equal to each other, i.e., when we have a square. This is the answer to the differential problem in which we are given the sum of the two sides *a* and *b* of a rectangle and asked for the exact values of the sides if the area of the rectangle is to be maximized. Alas, in our plane of hexagons with $\hat{e}_1$ and $\hat{e}_2$ being the base vectors of the system with an angle of 30º between, we do not posses the option to take fractional factors of these base vectors, so we have to approach the maximum symmetry as we are allowed.

The algorithm for finding such a geometry can be proposed as follows:

Let G(n,m) be the geometry for a protuberance that yields an (n,m) carbon nanotube. Examples for various G(n,m) values are presented in Figure 19 (Although they are shapes in 2-dimension before the optimization process, they can also be viewed as Schegel diagrams for the cause). We then define the notation [o_0- o_1- o_2- …- o_k- o_{k+1}] with o_i $\left(i \in \mathbb{Z}, i \geq 0\right)$ representing the number of hexagons in the i^{th} row before any modifications, i.e., addition of heptagons and pentagons, are made. The notation {o_i} will imply the series in general. Additional approaches to the protuberances can be found in An et al.'s [75] and Beuneu's [76] articles in which the protuberances are named as 'embryos'.

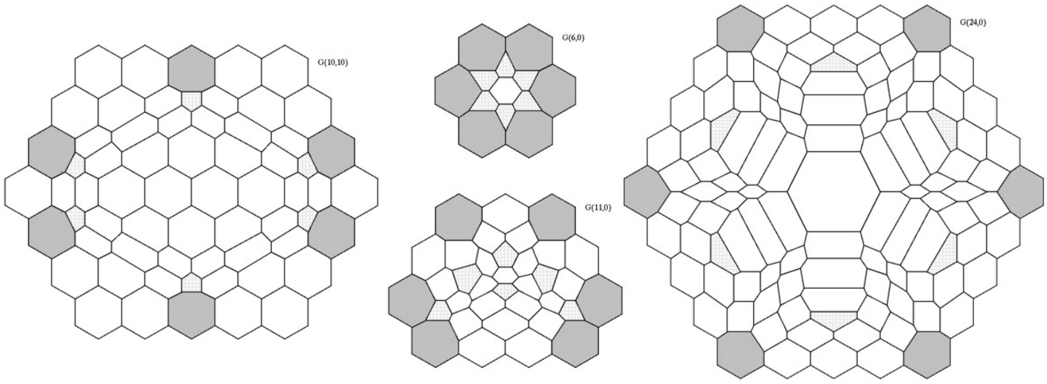


Figure 19: Various protuberances.

From basic geometric properties and using the ith rule, one can derive Table 2. k_L is the limit value after which, the symmetric *k* values begin to occur. So, it will be sufficient to search up to k_L. An optimized diagram given in Figure 20 equivalent to Table 2 can be constructed.

Table 2: k values.

n	k	k_L=$(n$-$o_0)$ / 2	k =
odd	even	integer, even	k_L - 2
		fractional, ip(k_L) even	integer part of k_L
		integer, odd	k_L - 1
		fractional, ip(k_L) odd	integer part of k_L - 1
even	odd	integer, even	k_L - 1
		fractional, ip(k_L) even	integer part of k_L - 1
		integer, odd	k_L - 2
		fractional, ip(k_L) odd	integer part of k_L

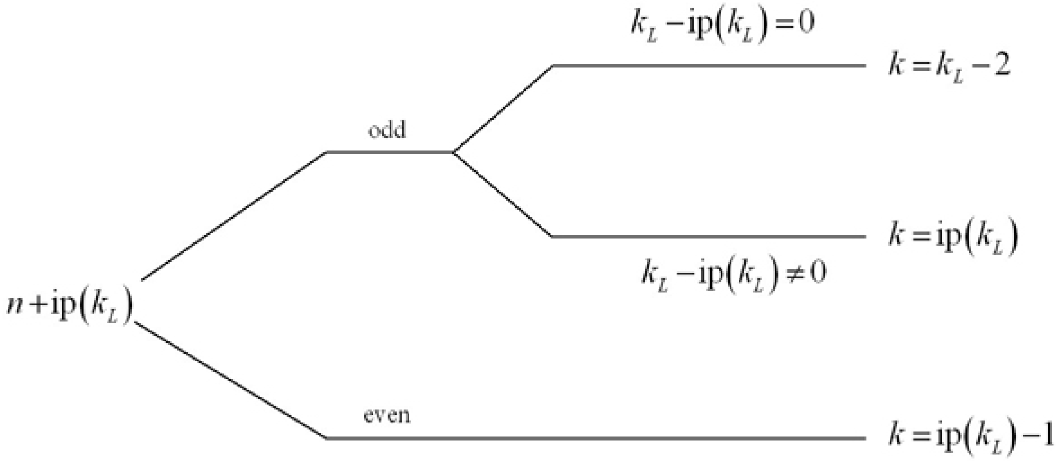


Figure 20: Table 2’s optimized diagram.

Knowing k, o_{k+1} is obtained by

$$o_{k+1} = n - \left(2k + o_0\right) \tag{2.72}$$

Now that we have k and o_{k+1}, we can construct the series. Let k_+ and k_- be the number of steps that increment and decrement o_0 respectively. Using the formulas

$$k + 1 = k_+ + k_- \tag{2.73}$$

$$o_{k+1} - o_0 = k_+ - k_- \tag{2.74}$$

we can now derive k_+ and k_- as follows:

$$\begin{gathered} k_+ = \frac{k + o_{k+1} - o_0 + 1}{2} \\ k_- = k - k_+ + 1 \end{gathered} \tag{2.75}$$

To acquire the maximum area, successive increments and decrements should be applied in order.

Now that we have an algorithm that calculates the series, we can iterate over the o_0 parameters for a given (fixed) n and by comparing the 'areas' of these series, maximum symmetrical series can be chosen. The simple flowchart for the selection of the maximum area series is given in Figure 21. Also, it must be kept in mind that the algorithm presented in Table 2 and Figure 21 gives the highest k for which the ith and iind statements are satisfied. If we take the o_i series for protuberance (17,0) with o_0 = 4 for example, the algorithm will yield 6 for k and 1 for o_{k+1} from which the constructed o_i series will be [4-5-6-5-4-3-2-1] whose perimeter is equal to 17 (n+m) and which increases/decreases by 1 for each row. So, the area and series calculation operation box in Figure 21 also checks for consecutive allowed k valued series which follows the rules

$$\begin{gathered} k = k - 2 \\ o_{k+1} = o_{k+1} + 4 \end{gathered} \tag{2.76}$$

so that the series that occupies more area ([4-5-6-7-6-5] in this case) is selected. Generally the first candidate has the largest area. Between G(6,0) and G(26,0) exceptions to this generalization were: n = 17, 20, 21, 23, 24, 25 and 26. Two alternative and ad-hoc methods deduced solely from Table 3 and Table 4 are presented in Appendix A.

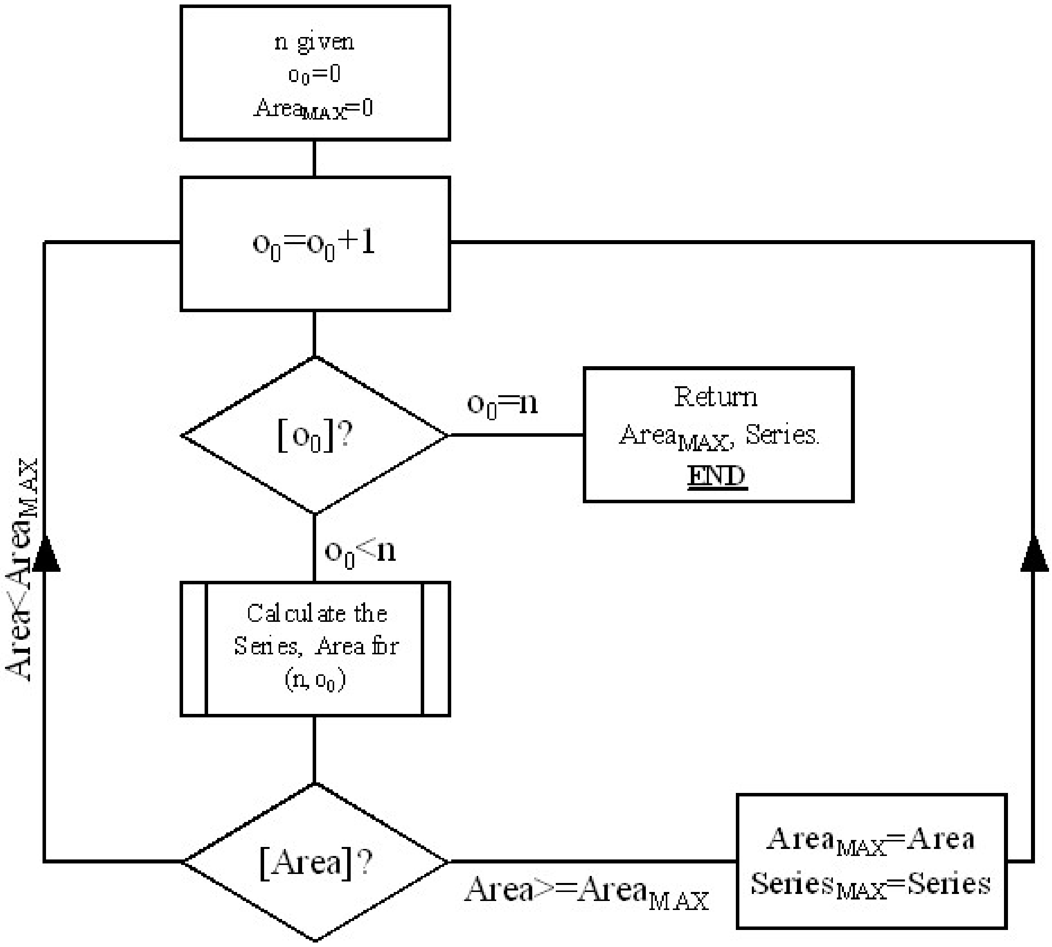


Figure 21: Flowchart for obtaining the maximum area yielding series for a given *n*.

Having acquired the proper, as symmetric as possible space, we can start filling up the protuberance by pentagons, hexagons and heptagons to form the geometry we need. We start by adding “half-lines” to the inner vertices of the hexagons in the outermost rim of the protuberance. The direction of these half-lines must be chosen such that they form an angle of approximately 120º with respect to the two sides forming the vertex. In the second step, we connect these lines with the nearest two with a bent line: if we are dealing with zigzag protuberances, then the corner hexagons, else if we are dealing with armchair protuberances, the hexagons in the centers of the sides will be transformed to heptagons. Two examples for the evolution of two protuberances, one zigzag and one armchair, are given in Figure 22, the second column covering the process described above.

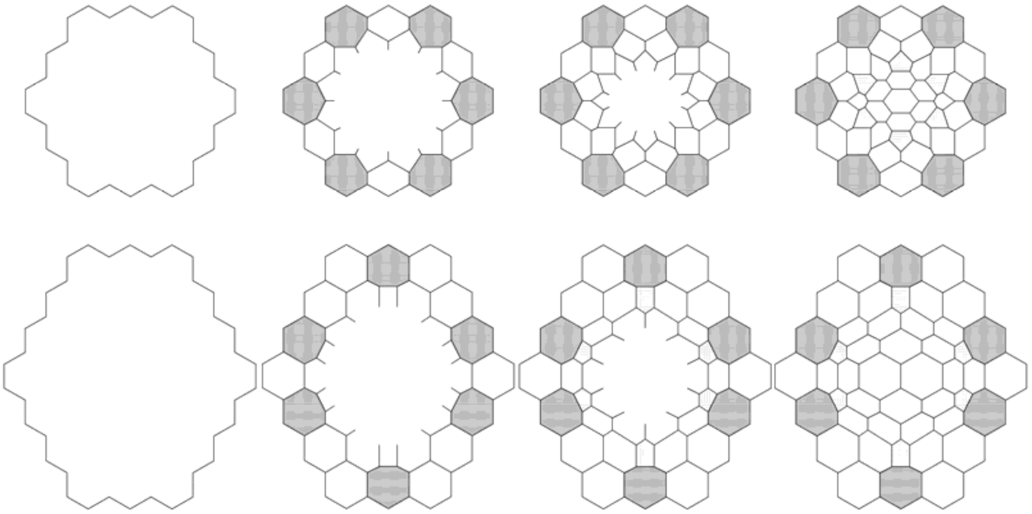


Figure 22: Evolution of protuberances G(12,0) and G(8,8).

At this stage, we have enough information to define the p_i series {p0:p1-p2:p3-p4:p5}as the number of hexagons between the corners going from top to bottom, left to right as shown in Figure 23. The relation between o_i and p_i is given as:

$$\begin{aligned} p_0 &= o_0 - 2 \\ p_1 = p_2 &= \max(o_i) - o_0 - 1 \\ p_3 = p_4 &= \max(o_i) - o_{\max(i)} - 1 \\ p_5 &= o_{\max(i)} - 2 \end{aligned} \tag{2.77}$$

Notice that, as a result of incrementing/decrementing the number of hexagons allowed in the subsequent rows by 1 hexagon at a time, $p_1 = p_2$ and $p_3 = p_4$. If we have two adjacent heptagons, then the corresponding *p* value would be 0.

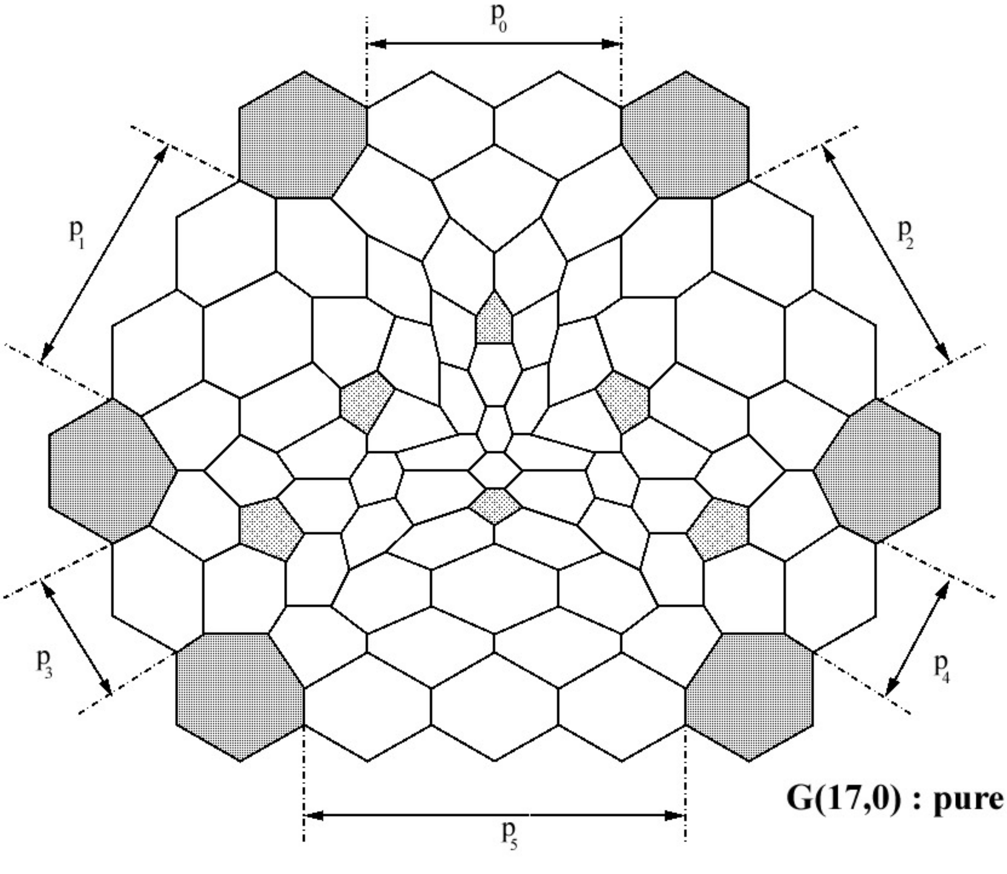


Figure 23: G(17,0) with {2:2-2:1-1:3}.

To hold up the Euler's rule [77] for regular shapes, the number of pentagons used must be equal to the number of heptagons.

3.3.2.1 Method for constructing zigzag tube yielding protuberances G(n,0)

If (p_0, p_1, p_2) set is formed by all even (excluding 0) numbers and (p_3, p_4, p_5) set is formed by all odd numbers, and vice versa, the zig-zag protuberance will be mentioned as 'pure'.

i. If there are adjacent heptagons, a pentagon must be placed between them in the next rim.

ii. If p_i is odd and the protuberance is not a 'pure' one, then a pentagon must be placed next to the hexagons in the middle (located on the first rim) in the second rim. If it is a 'pure' one, then place two pentagons corresponding to p_1 and p_2, next to the hexagons in the middle (located on the first rim) in the second rim, but place the pentagon for p_0 next to the hexagons in the middle (located on the third rim) in the fourth rim.

iii. If p_i is even, then a pentagon must be placed next to the hexagons in the middle (located on the second rim) in the third rim.

iv. After the placement of the heptagons and octagons, the remaining space is filled by hexagons.

While investigating the structures for a general formula, we have come up with a chart as given in Table 3.

In the table, rim occupation lists the total number of polygons, i.e. hexagons, pentagons and heptagons – with the number of pentagons occurring in that particular rim is given in parentheses. As can be verified from the p_i column, G(7,0), G(17,0), G(19,0), G(23,0), G(25,0) are pure protuberances in this manner.

Using basic geometric principles, we can derive the number of polygons in a rim from prior rims' line-up information. Also, we have the condition that the protuberance contains 6 heptagons in its first rim. Summarizing the information so far, a method for deriving the number and the positions of all the polygons in a rim is given below:

The total number of polygons in the first rim equals to n with 6 of these being heptagons placed in the corners. The total number of polygons in any rim other than the first one is equal to the number of polygons in the previous rim minus the total number of pentagons so far placed. This goes on until the calculated polygon total reaches 0 in which case it is treated as one; or the total number of polygons calculated for the rim is negative in which case it is omitted and the rims are concluded.

Table 3: Geometric properties of zigzag yielding protuberances G(n,0).

G(n,0)	***$\{o_i\}$***	***$\{p_i\}$***	**rim occupation**	**# of rims**
G(6,0)	2-3-2	0:0-0:0-0:0	6-6(6)-1	3
G(7,0)	1-2-3-2	-1:1-1:0-0:0	7-8(6)-2(1)	3
G(8,0)	3-4-3	1:0-0:0-0:1	8-8(4)-4(2)	3
G(9,0)	2-3-4-3	0:1-1:0-0:1	9-9(3)-6(3)-1	4
G(10,0)	2-3-4-3-2	0:1-1:1-1:0	10-10(2)-8(4)-2	4
G(11,0)	3-4-5-4	1:1-1:0-0:2	11-11(2)-9(3)-4(1)	4
G(12,0)	3-4-5-4-3	1:1-1:1-1:1	12-12-12(6)-6-1	5
G(13,0)	2-3-4-5-4-3	0:2-2:1-1:1	13-13(1)-12(3)-8(2)-2	5
G(14,0)	4-5-6-5-4	2:1-1:1-1:2	14-14-14(4)-10(2)-4	5
G(15,0)	3-4-5-6-5-4	1:2-2:1-1:2	15-15-15(3)-12(3)-6-1	6
G(16,0)	3-4-5-6-5-4-3	1:2-2:2-2:1	16-16-16(2)-14(4)-8-2	6
G(17,0)	4-5-6-7-6-5	2:2-2:1-1:3	17-17-17(2)-15(3)-10(1)-4	6
G(18,0)	4-5-6-7-6-5-4	2:2-2:2-2:2	18-18-18-18(6)-12-6-1	7
G(19,0)	3-4-5-6-7-6-5-4	3:3-3:2-2:2	19-19-19(2)-17(3)-12(1)-6-1	7
G(20,0)	5-6-7-8-7-6-5	3:2-2:2-2:3	20-20-20(2)-18(4)-12-6-1	7
G(21,0)	4-5-6-7-8-7-6-5	2:3-3:2-2:3	21-21-21(3)-18(3)-12-6-1	7
G(22,0)	4-5-6-7-8-7-6-5-4	2:3-3:3:3-2	22-22-22(4)-18(2)-12-6-1	7
G(23,0)	5-6-7-8-9-8-7-6	3:3-3:2-2:4	23-23-23(2)-21(3)-16(1)-10-4	7
G(24,0)	5-6-7-8-9-8-7-6-5	3:3-3:3-3:3	24-24-24(6)-18-12-6-1	7
G(25,0)	4-5-6-7-8-9-8-7-6-5	2:4-4:3-3:3	25-25-25(2)-23(3)-18(1)-12-6-1	8
G(26,0)	6-7-8-9-10-9-8-7-6	4:3-3:3-3:4	26-26-26(4)-22(2)-16-10-4	7

For every p_i that is equal to 0, a pentagon is placed in the second rim, between the adjacent heptagons corresponding to that p_i side. If the protuberance is not a pure one, then the total number of the pentagons in the third rim is equal to the number of odd p_i , else if the protuberance is a pure one, then the total number of pentagons in the third rim is equal to the number of odd p_i minus 1. A number of pentagons that is equal to the number of even p_i are placed in the fourth rim and if the protuberance is pure, then another pentagon is placed in the fifth rim. All these pentagons are positioned in the centers of the sides that are parallel to p_i lines such as the ones shown in Figure 23 for the G(17,0) protuberance.

3.3.2.2 Method for constructing armchair yielding protuberances G(n,n)

For G(n,n) protuberances, similar techniques are employed but the method differs itself in the positions of the heptagons. In contrast to G(n,0), the heptagons are places not in the corners but, as close as possible to the centers of the sides. The two corners occurring in $\max\left(\{o_i\}\right)$ are always hexagons but in some cases, when there is nowhere else, the corners of o_0 and $o_{\max(i)}$ can also be filled with heptagons. Two examples for this can be found in Figure 24, being the G(4,4) and G(5,5). The $\{o_i\}$ series for a G(n,n) protuberance is equal to that of a G(2n,0) protuberance's series. Also, note that, any armchair protuberance will always be symmetric along the horizontal axis which is taken as the imaginary line connecting the two hexagons' centers placed on the edges of $\max\left(\{o_i\}\right)$ because of the equity to the $\{o_i\}$ series of a 2*n* zigzag protuberance.

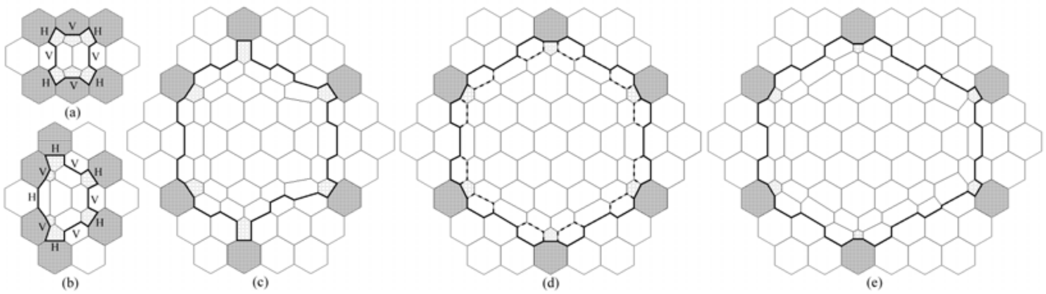


Figure 24: Examples of G(n,n) protuberances. (a) G(4,4); (b) G(5,5); (c) G(11,11); (d) G(12,12); (e) G(13,13).

The ‘pureness’ situation encountered in the zigzag protuberances G(n,0) has a similar counterpart for G(n,n): if *n* is even, then o_1 will be odd and the right side will be a mirror image of the left side with respect to the vertical axis which is taken as the imaginary line connecting the centers of o_1 and $o_{\max(i)}$. But in the case of an odd *n*, o_0 will be even and the protuberance space will be that of *n*+1 case with $\{o_i\}$ series of the G(*n*+1, *n*+1) decreased by 1 each. Let us call the first case with *n* even ‘pure’ and the other ‘impure’. For an impure protuberance, the heptagons placed for p_1 and p_2 should be on the same o_i row and they must be placed into the centers of the p_1 and p_2 for the odd p_1 and p_2 and for even p_1 and p_2, the hexagon closer to $\max(\{o_i\})$ must be transformed into a heptagon. Similar procedure should be employed for p_3 and p_4. Because of the vertical symmetry, the heptagons for p_0 and p_5 can be placed in either of the hexagons in the middle f the top and bottom row as long as p_0 and p_5 ‘s heptagons are in the same column, i.e. mirror imaging each other by means of horizontal symmetry. In our work the one to the left has been preferred for the purpose but this is not mandatory. The positions of the heptagons are as follows:

i. p_0 heptagon will be placed into the $\left[p_0+2+(p_0\%2)\right]/2.$ hexagon to the left, numbering the upper-left hexagon as the 1st.

ii. p_1 heptagon will be placed into the $\left[p_1+2+(p_0\%2)\right]/2.$ hexagon to the left, numbering the upper-left hexagon as the 1st.

iii. p_2 heptagon will be placed in the same o_i row with the p_1 heptagon.

iv. p_3, p_4 and p_5 heptagons will be the mirror images of p_1, p_2 and p_0 heptagons, repectively, with respect to $\max(\{o_i\})$ row.

In the equations above, the ‘%’ operator is the remainder operator, i.e. (odd number)%2 = 1, (even number)%2 = 0.

Generation of the G(n,n) protuberances is fairly easy when compared to G(n,0) protuberances. The pentagons are always placed in the second rim and they are positioned adjacent to the heptagons. As a result of alternating series in valley-hill sequence, a pentagon-heptagon intersecting side will be designated as the one before it if there are odd number of hexagons between the two related heptagons in the first rim, and will be designated as the opposite designation of the one before it if there are even number of hexagons in between.

For a G(n,n) protuberance, the total number of the polygons in the first rim is equal to 2*n* with 6 of these being heptagons. The second rim also has the same number of polygons as the first rim, being 2*n* with the 6 heptagons replaced by 6 pentagons. The inner rims follows exactly the same way as G(n,0) protuberances and terminates just like them and they may only contain hexagons. A table for the properties of some of these protuberances is given in Table 4.

Table 4: Geometric properties of armchair yielding protuberances G(*n*,*n*).

G(n,n)	***{o_i}***	***{p_i}***	**rim occupation**	**# of rims**
G(4,4)	3-4-3	1:0-0:0-0:1	8-8(6)-2	3
G(5,5)	2-3-4-3-2	0:1-1:1-1:0	10-10(6)-4	3
G(6,6)	3-4-5-4-3	1:1-1:1-1:1	12-12(6)-6-1	4
G(7,7)	4-5-6-5-4	2:1-1:1-1:2	14-14(6)-8-2	4
G(8,8)	3-4-5-6-5-4-3	1:2-2:2-2:1	16-16(6)-10-4	4
G(9,9)	4-5-6-7-6-5-4	2:2-2:2-2:2	18-18(6)-12-6-1	5
G(10,10)	5-6-7-8-7-6-5	3:2-2:2-2:3	20-20(6)-14-8-2	5
G(11,11)	4-5-6-7-8-7-6-5-4	2:3-3:3:3-2	22-22(6)-16-10-4	5
G(12,12)	5-6-7-8-9-8-7-6-5	3:3-3:3-3:3	24-24(6)-18-12-6-1	6
G(13,13)	6-7-8-9-10-9-8-7-6	4:3-3:3-3:4	26-26(6)-20-14-8-2	6
G(14,14)	5-6-7-8-9-10-9-8-7-6-5	3:4-4:4-4:3	28-28(6)-22-16-10-4	6

After filling up the protuberance space as defined above, we have to designate the ‘path’ of the nanotube contained within the protuberance. The practical way to do this is to begin from the uppermost pentagon-heptagon intersection. Label this line as ‘V’ for ‘valley’ or ‘H’ for ‘hill’. Although it may come out later that, it is –for example– a hill instead of a valley, the designation of the type is important only in distinguishing one type from the other. So, you can as well designate the two types as ‘A’ and ‘B’ instead of ‘V’ and ‘H’ as we have done. Once you have signed the uppermost line, try to reach the nearest pentagon-heptagon pair using steps such as a ╱‾╲ or ╲_╱, that increases-goes flat-decreases (or vice versa) in a direction so that they always have a minima or a maxima but never a ╱‾╱, which is like a saddle point. If you can not reach the next junction line, then treat the starting line as the opposite type. Some protuberances (like G(5,5), G(8,8), G(11,11), G(14,14)) are found to comply with the uppermost junction line being a ╱‾╲ step, some protuberances (like G(4,4), G(7,7), G(9,9), G(13,13)) found to comply with the uppermost junction line being a ╲_╱ step, while some (like G(6,6), G(10,10), G(12,12)) complied with both ways. Although there may be a relation between this behavior and the *n*, it could not be derived. For this reason, at this point, the systematic method works as a trial and error method with two options. Figure 24 shows the possible trailing –thus cut-off– lines for nanotube joints.

3.3.3 Flowcharts

In order to aid in computer programming, the processes for the generation of protuberances are summarized in the following flowcharts in Figure 25 through Figure 29:

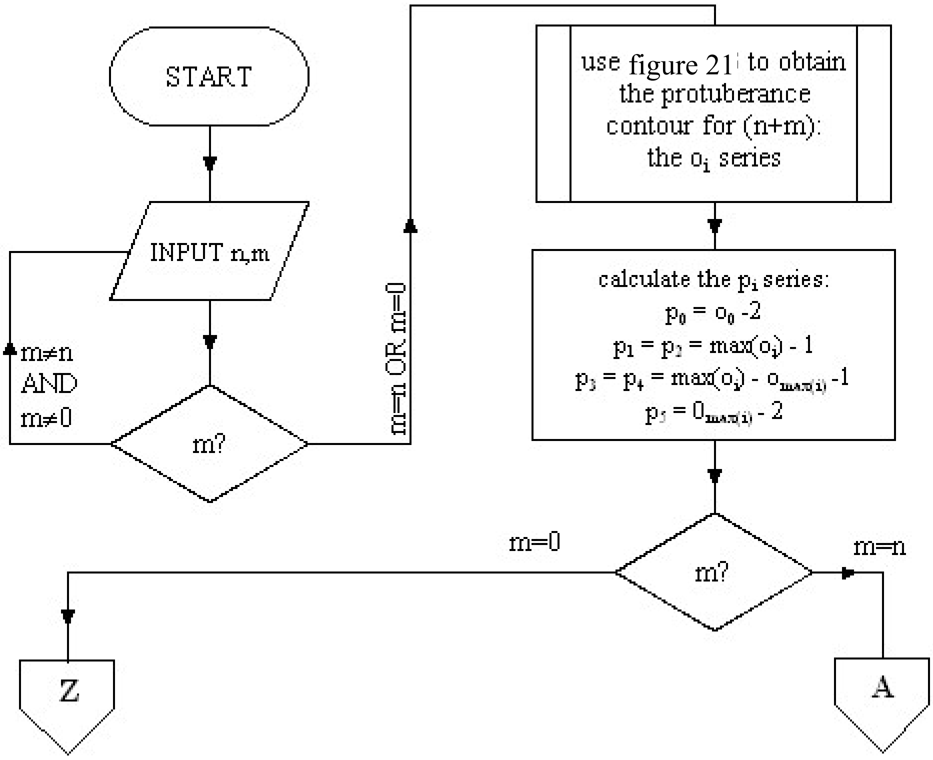


Figure 25: Flowchart to generate protuberances (part I).

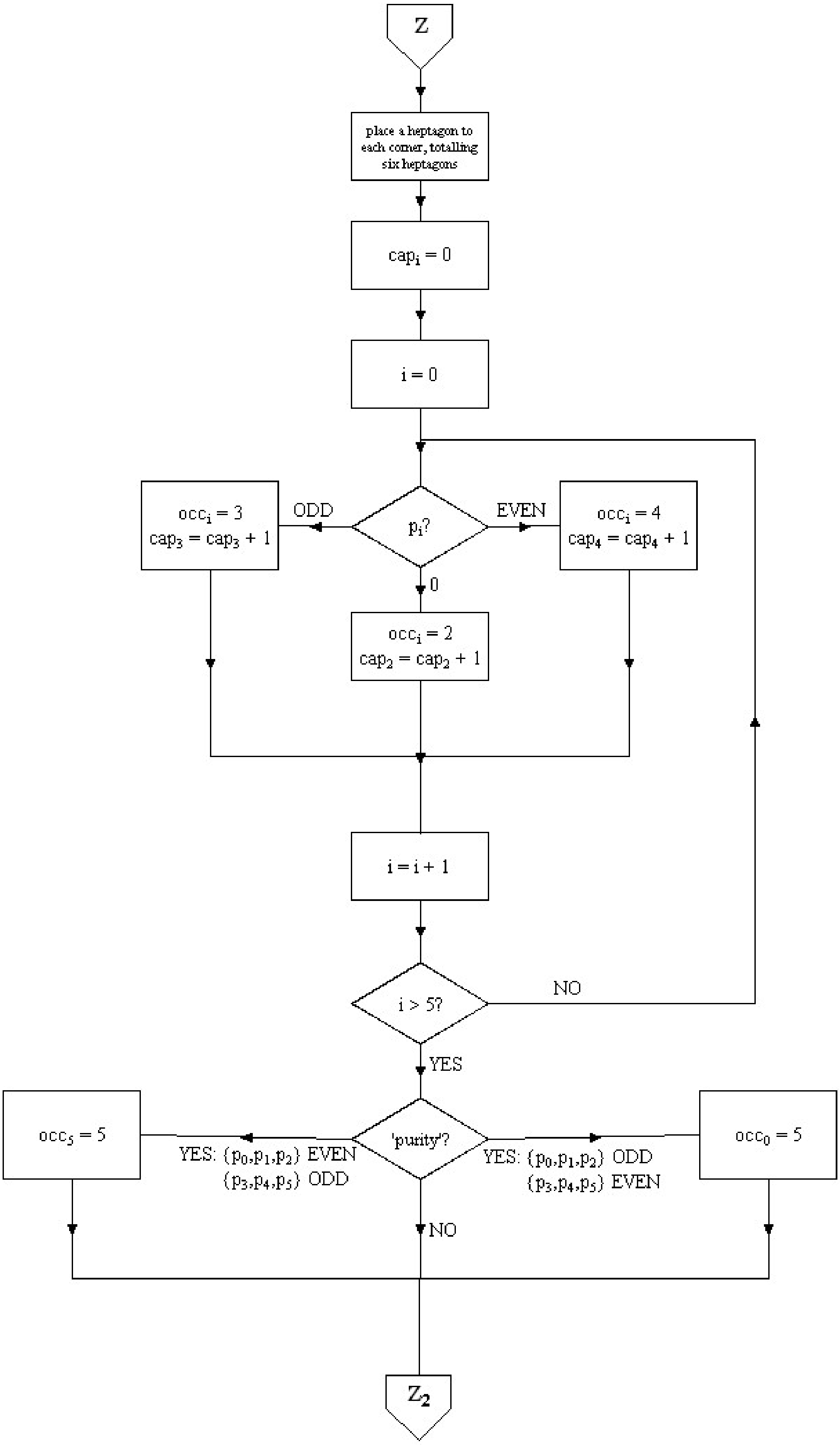
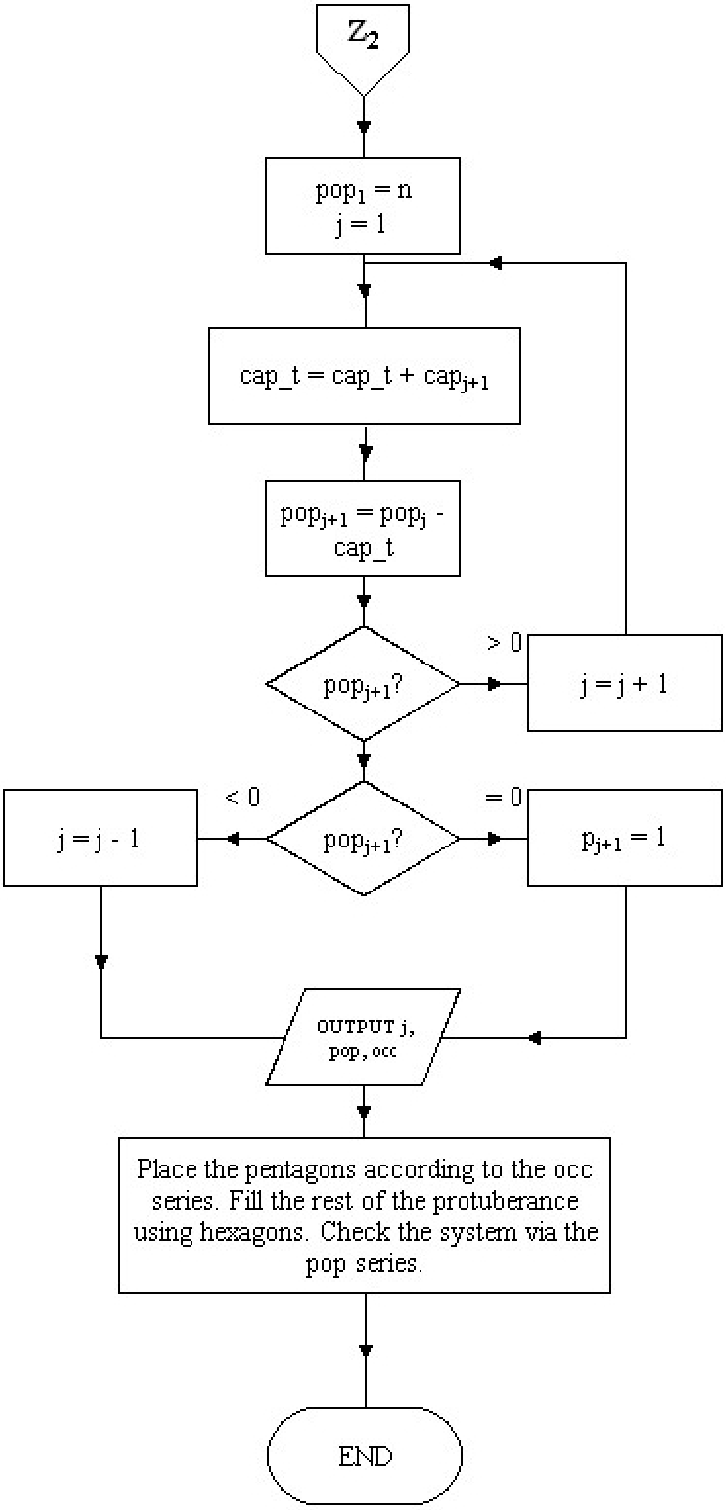


Figure 26: Flowchart to generate protuberances (part II). The zigzag protuberance generator (1/2).



Z_2
$pop_1 = n$
$j = 1$
$cap\_t = cap\_t + cap_{j+1}$
$pop_{j+1} = pop_j - cap\_t$
$pop_{j+1}?$
> 0
$j = j + 1$
< 0
= 0
$j = j - 1$
$p_{j+1} = 1$
OUTPUT j, pop, occ
Place the pentagons according to the occ series. Fill the rest of the protuberance using hexagons. Check the system via the pop series.
END

Figure 27: Flowchart to generate protuberances (part II). The zigzag protuberance generator (2/2).

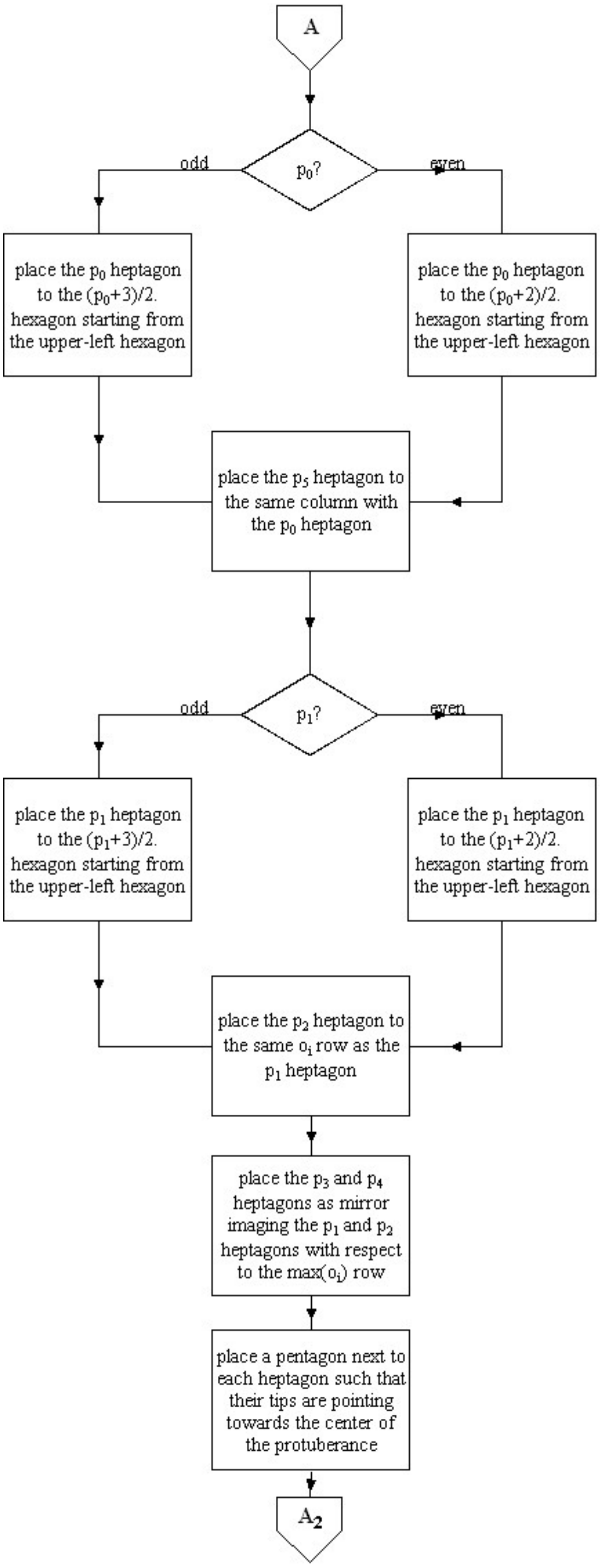

A
p_0?
odd
even
place the p_0 heptagon to the $(p_0+3)/2$. hexagon starting from the upper-left hexagon
place the p_0 heptagon to the $(p_0+2)/2$. hexagon starting from the upper-left hexagon
place the p_5 heptagon to the same column with the p_0 heptagon
p_1?
odd
even
place the p_1 heptagon to the $(p_1+3)/2$. hexagon starting from the upper-left hexagon
place the p_1 heptagon to the $(p_1+2)/2$. hexagon starting from the upper-left hexagon
place the p_2 heptagon to the same o_i row as the p_1 heptagon
place the p_3 and p_4 heptagons as mirror imaging the p_1 and p_2 heptagons with respect to the max(o_i) row
place a pentagon next to each heptagon such that their tips are pointing towards the center of the protuberance
A_2

Figure 28: Flowchart to generate protuberances (part II). The armchair protuberance generator (1/2).

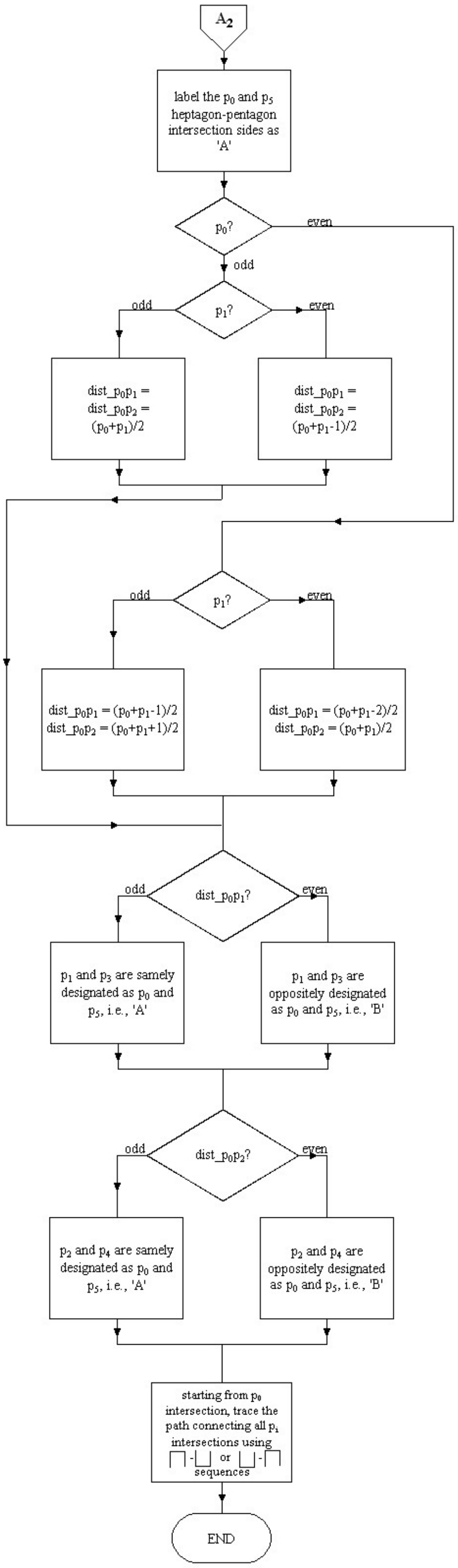

A₂
label the p₀ and p₅ heptagon-pentagon intersection sides as 'A'
p₀?
even
odd
p₁?
odd
even
dist_p₀p₁ = dist_p₀p₂ = (p₀+p₁)/2
dist_p₀p₁ = dist_p₀p₂ = (p₀+p₁-1)/2
p₁?
odd
even
dist_p₀p₁ = (p₀+p₁-1)/2 dist_p₀p₂ = (p₀+p₁+1)/2
dist_p₀p₁ = (p₀+p₁-2)/2 dist_p₀p₂ = (p₀+p₁)/2
dist_p₀p₁?
odd
even
p₁ and p₃ are samely designated as p₀ and p₅, i.e., 'A'
p₁ and p₃ are oppositely designated as p₀ and p₅, i.e., 'B'
dist_p₀p₂?
odd
even
p₂ and p₄ are samely designated as p₀ and p₅, i.e., 'A'
p₂ and p₄ are oppositely designated as p₀ and p₅, i.e., 'B'
starting from p₀ intersection, trace the path connecting all pᵢ intersections using ⊓-⊔ or ⊔-⊓ sequences
END

Figure 29: Flowchart to generate protuberances (part II). The armchair protuberance generator (2/2).

3.3.4 Various Applications of Protuberances

With protuberances at hand, one can form various systems including nanogears, nanojunctions, toothed canals as given in Figure 30 and Figure 31. It is also possible to join any nanotube to another by replacing appropriate (corresponding to the other tube's $\{o_i\}$ series) hexagons from the tube with the protuberance corresponding to the tube that will be joined, and then cutting up the 'cape section' of the protuberance and growing it into a tube with desired length. Also nanojunctions involving many different diameters and different chirality-valued tubes can be formed easily like the example given in Figure 17.

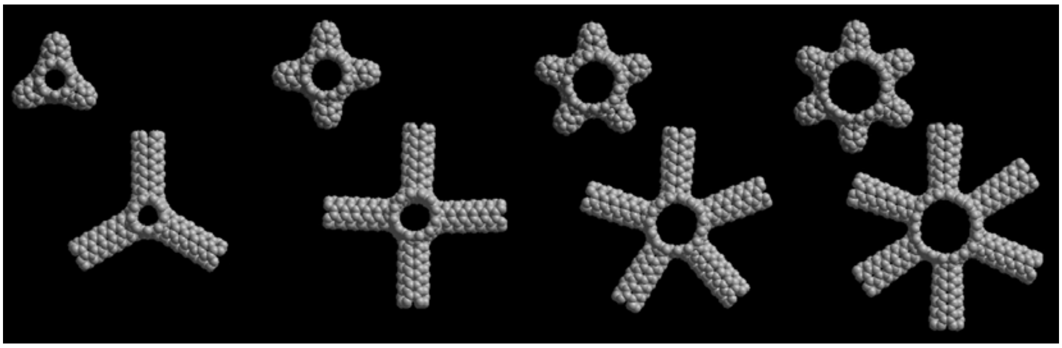


Figure 30: 3, 4, 5 and 6 toothed nanogears formed using G(4,4) with their junction counterparts with the top of the protuberances cut-off and joined to C(4,4) nanotubes.

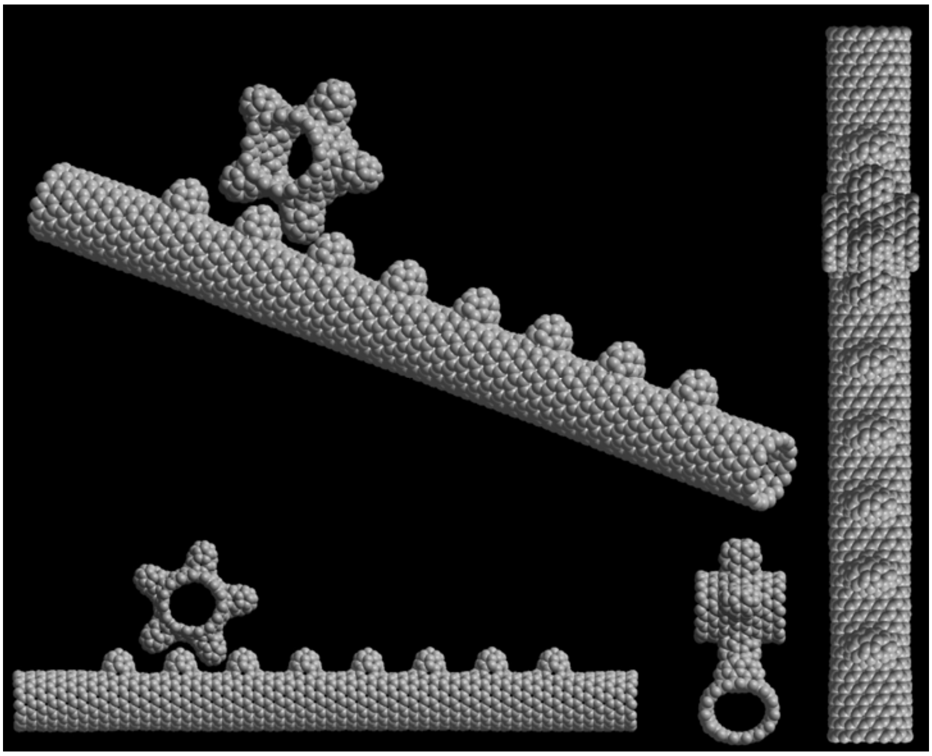


Figure 31: Various views of a toothed canal system.

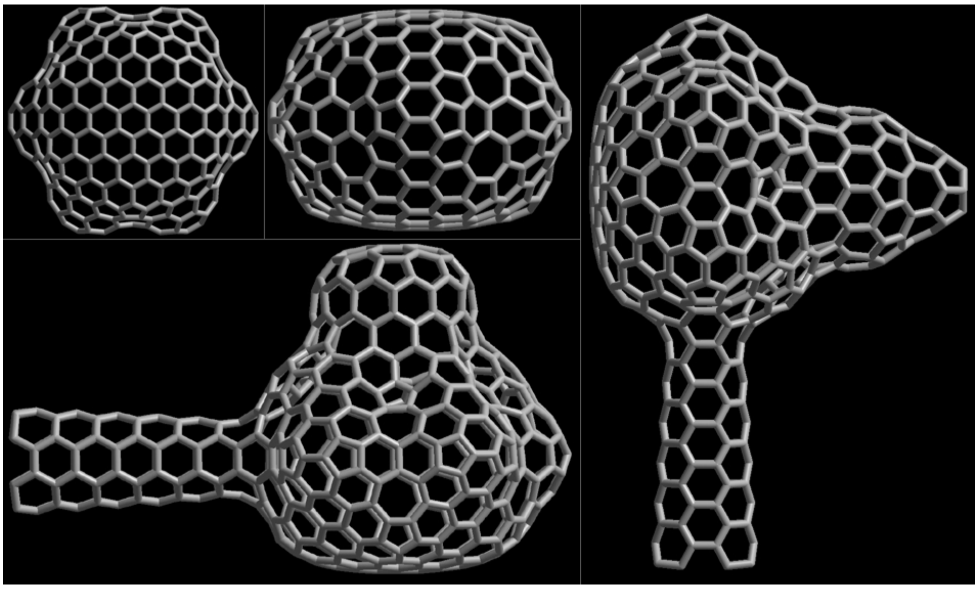


Figure 32: Formation of a junction point using two G(12,12) and then adding G(4,4) and G(12,0) . Next, C(4,4) is connected to G(4,4) and G(6,0) is connected to G(12,0).

CHAPTER 4

STRUCTURAL PROPERTIES OF THE NANOSYSTEMS STUDIED

Analysis of the systems produced in the previously described ways were conducted in three methods: The junction formation between two nanogears forced to intersect with each other was tested using molecular dynamics simulations. The nanotori and nanogears systems were firstly inspected for their temperature threshold in which they can no longer maintain their stability and bond-breakings begin to take place. For the case of nanogears, additional stability tests were applied under periodic boundary conditions which ensured that the pivot regions of the nanogears were connected to infinitely many other nanogears' pivot regions in a repeating way. Then, rough orbital information using the Extended Hückel method was obtained. Among this orbital information, the total energy of the system and the HOMO-LUMO energy levels were presented.

4.1 Molecular Dynamics Simulations

4.1.1 Junction Formation

After the system described in Section 2.1 was simulated with the methods and procedures described, they were observed to form a junction which was heterogeneous. The development snapshots of the two versions of the system – one with the graphene layers covering only the junction area while in the other, the graphene layers cover the whole system – are given in Figure 33 and Figure 34, respectively.

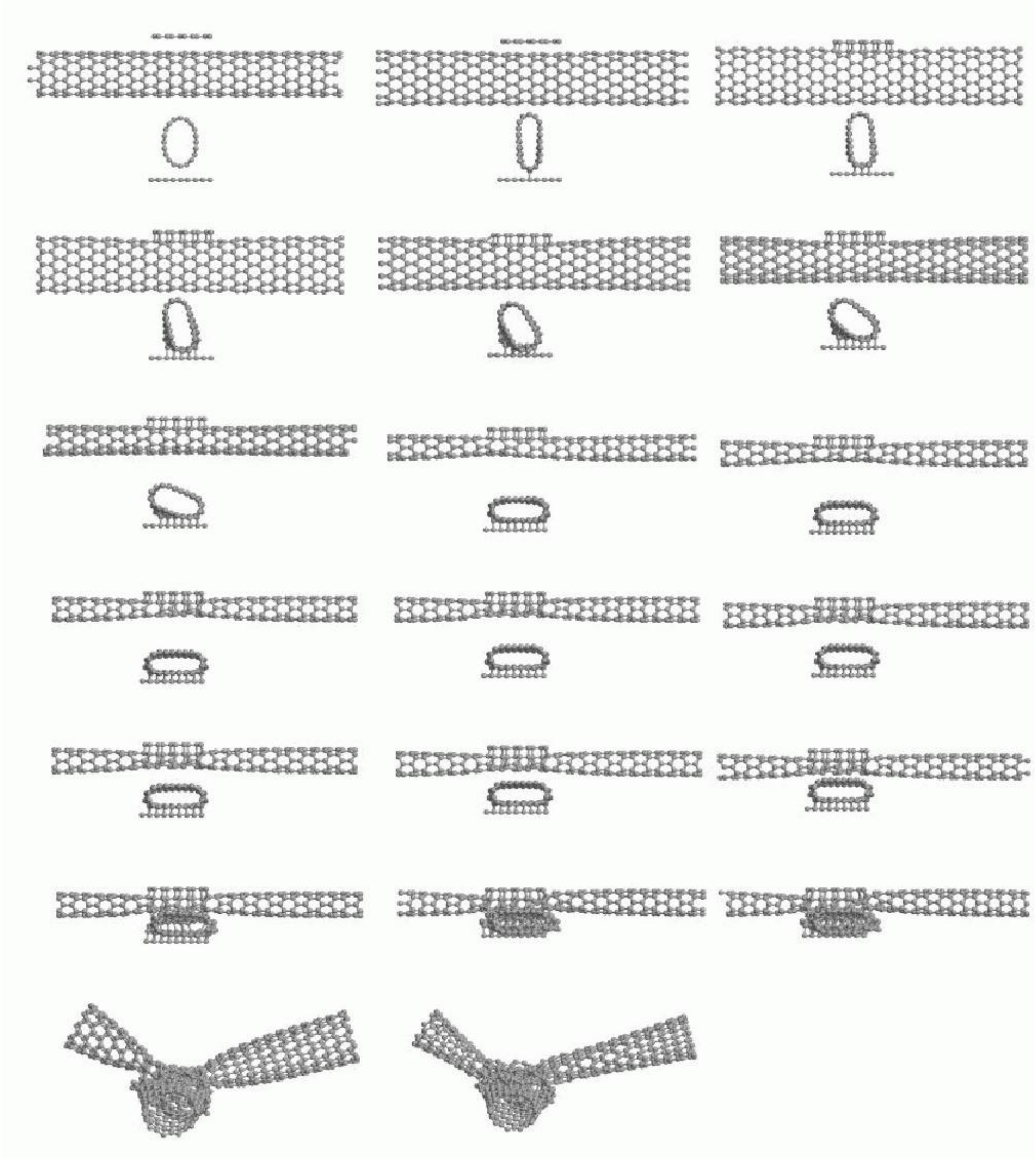


Figure 33: The development snapshots of the cross junction with the graphene layers covering only the junction area.

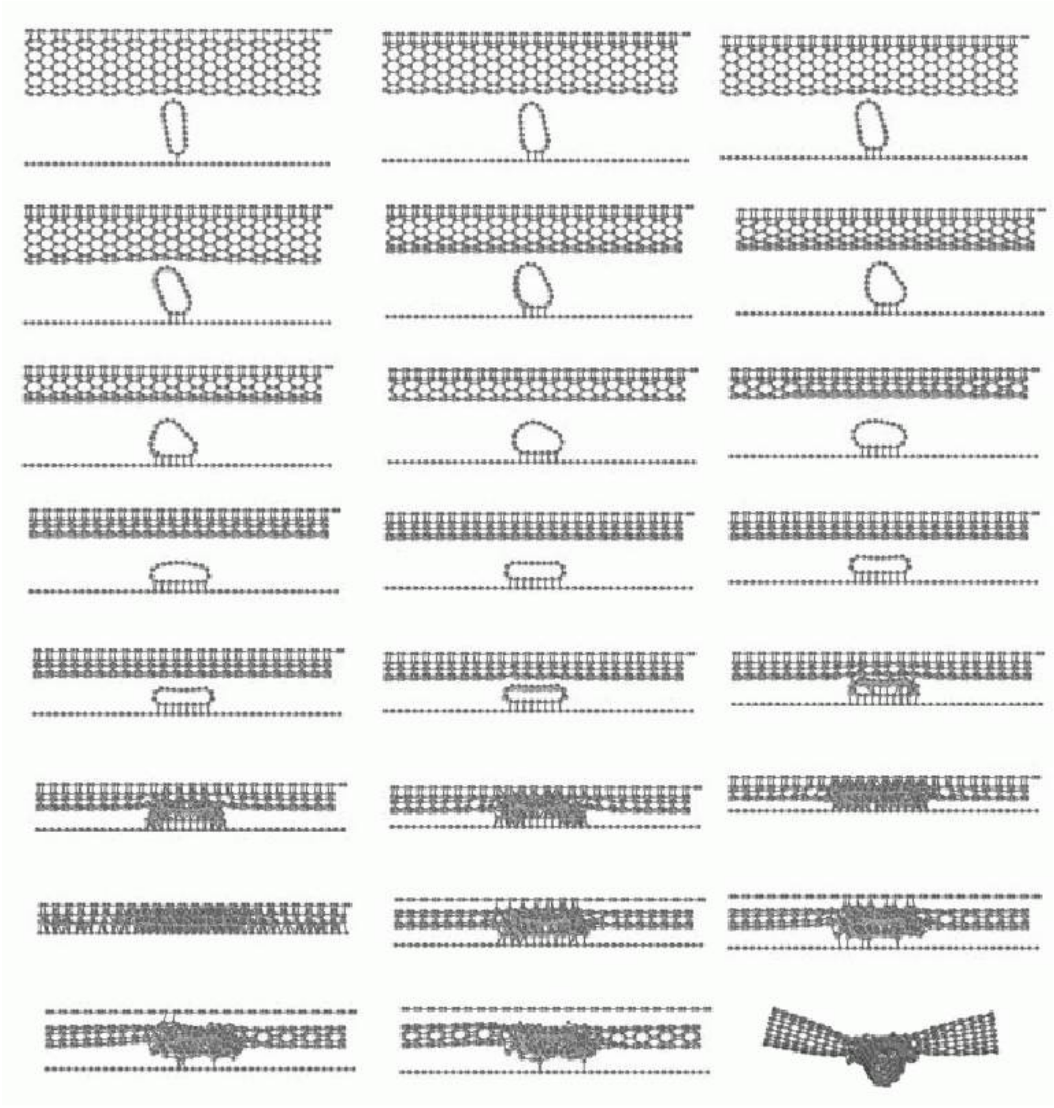


Figure 34: The development snapshots of the cross junction with the graphene layers covering the whole system.

4.1.2 Stability Tests

The generated structures are run for stability tests in various temperatures to obtain the decomposition behavior and determine the limit temperature in which the break down happens. All of the simulations were conducted using the Tersoff potential, NVT conditions regarded per run, with the temperature gradually increasing by 100 K at each run. One time step was taken to be 10^{-16} second and at each run a relaxation with 50000 time steps were performed. The simulation processes are summarized in Table 5 with the original systems' formation and their forms before and just after the deformation are presented through Figures 36-40 and 42-49. Deformation temperature may also be determined by looking at the variation of the total potential energy vs. temperature. Such a energy vs. temperature variation for the nanogears C_{240} and C_{252} is shown in Figure 35. Similar behavior is observed from the other models.

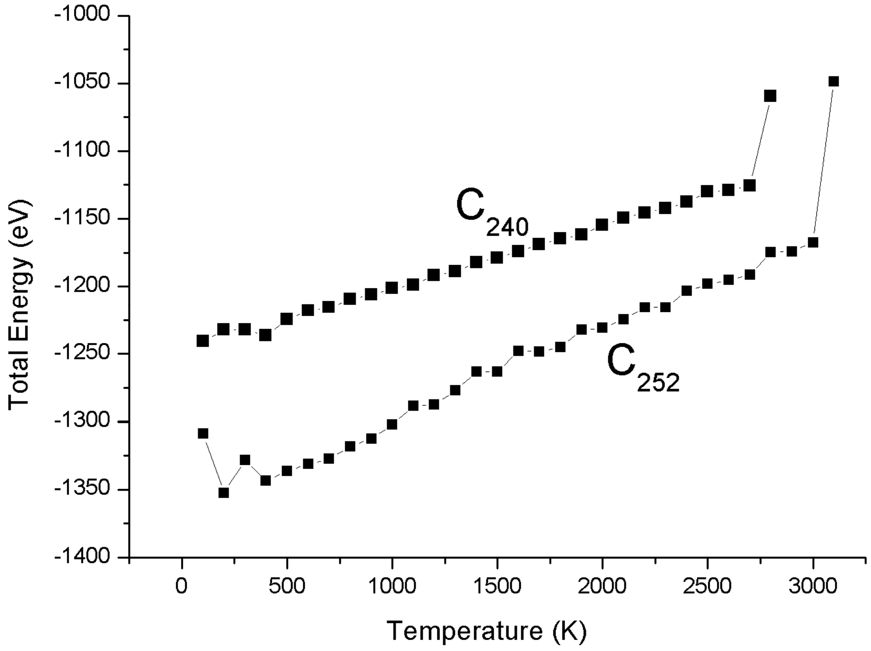


Figure 35: Energy vs. Temperature graph for C_{240} and C_{252}.

Table 5: The temperature thresholds for various carbon nanosystems.

System:	**Deformation Temperature**
Nanotorus: C_{170}	3700 K
Nanotorus: C_{250}	3000 K
Nanotorus: C_{360}	4500 K
Nanotorus: C_{520}	4200 K
Nanotorus: C_{750}	4100 K
3 toothed armchair nanogear: C_{240}	2800 K
3 toothed zigzag nanogear: C_{252}	3100 K
4 toothed armchair nanogear: C_{320}	2600 K
4 toothed zigzag nanogear: C_{336}	3200 K
5 toothed armchair nanogear: C_{400}	2400 K
5 toothed zigzag nanogear: C_{420}	2900 K
6 toothed armchair nanogear: C_{480}	1700 K
6 toothed zigzag nanogear: C_{504}	1200 K

4.1.2.1 Carbon Nanotori

Among the nanotori considered, C_{170} is the smallest possible toroidal structure that can be generated with the algorithm used. The remaining 4 tori are selected by increasing r and R defined in Figure 16. The stability of the toroidal structures against heat treatment shows a complex dependence on geometrical parameters.

Figure 36 shows the evolution of the C_{170} nanotorus. This nanotorus has almost circular appearance from its top view. The five-fold symmetry is not as apparent as others. It deformed at about 3700 K. The nanotorus has not distorted much until the deformation temperature at which, it became distorted significantly and the deformation started at a knee region.

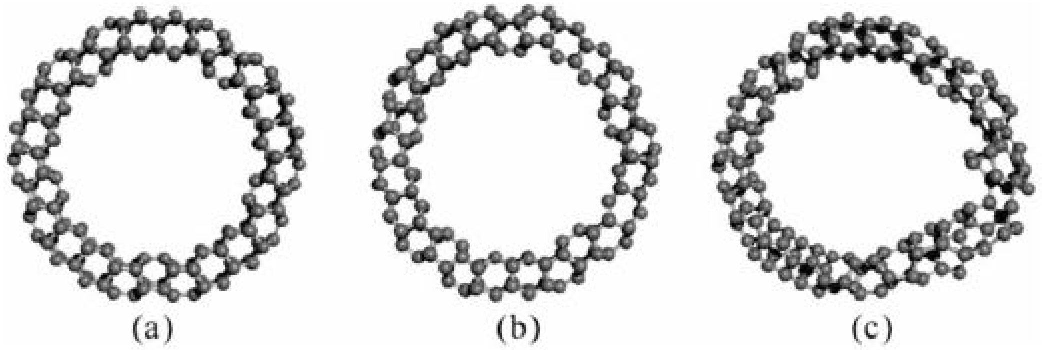


Figure 36: C_{170} (a) Relaxed structure at 1 K, (b) at 3600 K, (c) at 3700 K where the structure deforms.

C_{250} is shown in Figure 37. Pentagon-like geometry can be seen in the inner region. Although there is very little difference in *r*, *R* is changed considerably when compared to C_{170}. Geometry was not distorted much up to the deformation temperature at about 3000 K. C_{250} also starts to deform at a knee region.

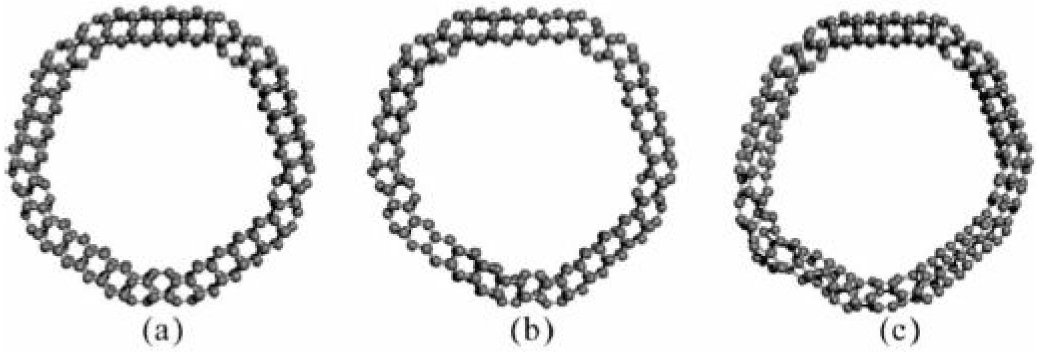


Figure 37: C_{250} (a) Relaxed structure at 1 K, (b) at 2900 K, (c) at 3000 K where the structure deforms.

The inner region geometry of C_{360} also resembles a pentagon but more round when compared to C_{250} as can be seen in Figure 38. Overall geometry was somewhat distorted when exposed to heat. This nanotorus persisted up to 4500 K, which is the highest temperature endured among the nanotori considered. It is interesting to note that this nanotorus deformed at a location away from the knee regions, unlike the other nanotori that were considered in this work.

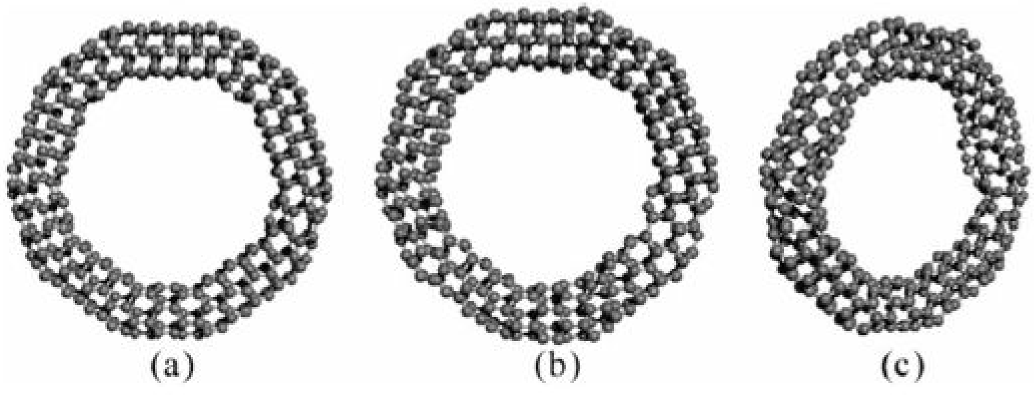


Figure 38: C_{360} (a) Relaxed structure at 1 K, (b) at 4400 K, (c) at 4500 K where the structure deforms.

R/r ratio for the nanotori C_{520} is the lowest among all the models considered. This can be observed in Figure 39. Inner region has circular geometry where the five fold symmetry is not apparent, whereas the five fold symmetry in the outer region is clearer. This nanotorus deformed at about 4200 K. Inner geometry became distorted when exposed to heat. This nanotorus also began its deformation at a knee region.

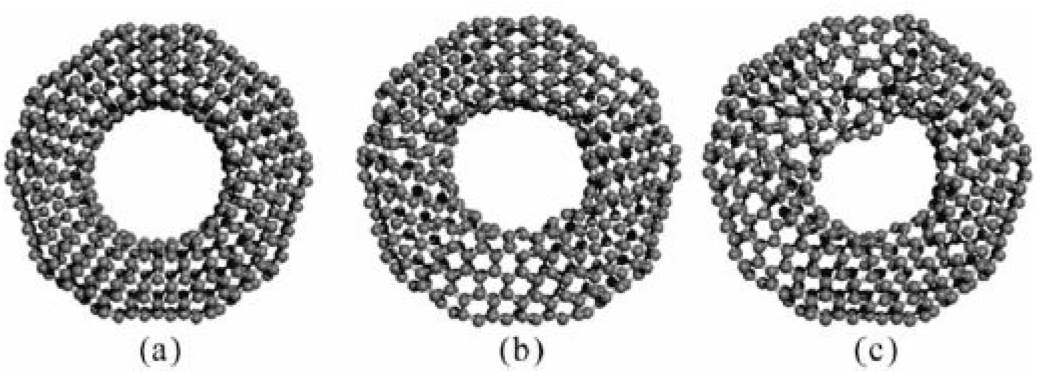


Figure 39: C_{520} (a) Relaxed structure at 1 K, (b) at 4100 K, (c) at 4200 K where the structure deforms.

C_{750} is the largest nanotorus considered in this work. It has a pentagon-like inner geometry that can be observed from Figure 40. This nanotorus deformed at about 4100 K at a knee region.

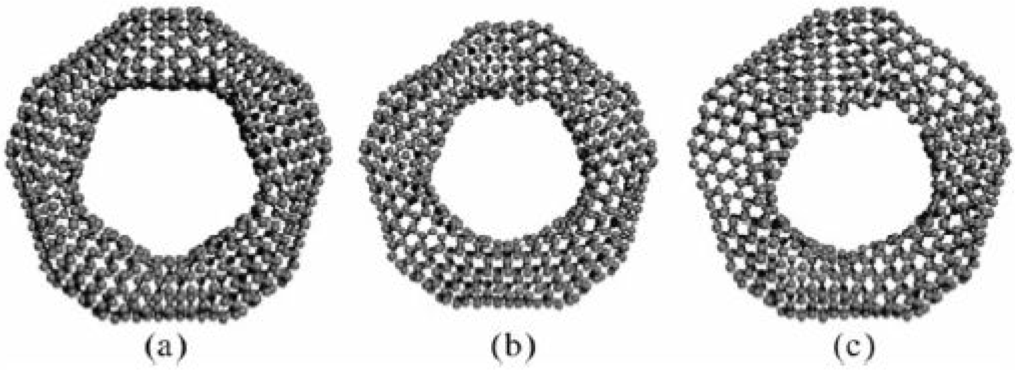


Figure 40: C_{750} (a) Relaxed structure at 1 K, (b) at 4000 K, (c) at 4100 K where the structure deforms.

In a similar work [78], thermal stability of nanotori with different geometry (different locations of pentagons and heptagons) but similar sizes has been investigated. In that work, C_{120} nanotorus was found to deform at about 4600 K with a little bit of distortion in the structure before the deformation temperature. Two different isomers of C_{240} had deformation temperatures of 4100 K and 5400 K. It may be argued that the smooth geometry of these nanotori have increased the thermal stability considerably, since strain due to curvature is distributed along the whole structure, and not concentrated at the knee regions as in the present case.

Overall, all the nanotori considered in this work are found to be thermally stable up to the elevated temperatures. The thermal stability of the nanotori considered has a complex dependence on geometrical parameters, as noted earlier. The variation of deformation temperature with respect to R / r is plotted in Figure 41.

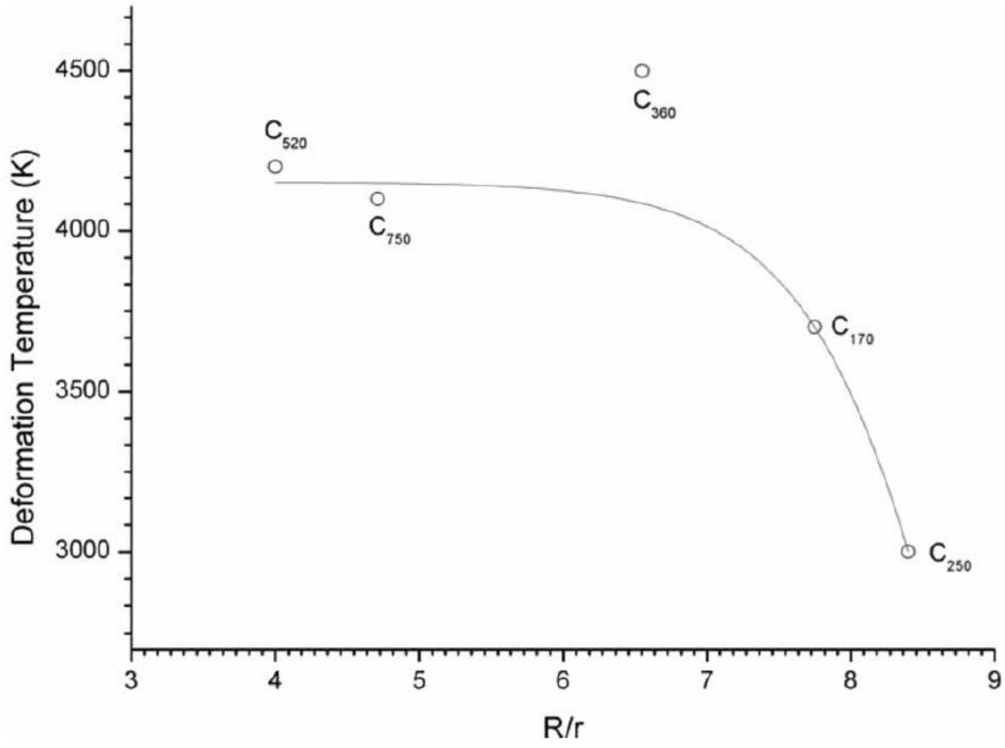


Figure 41: Dependence of the deformation temperature on *R*/*r*.

Considering the dependency, C_{360} is seen to be an exception, possibly due to the fact that the deformation location is away from the knee regions. It may be argued that this nanotorus satisfies some magic condition that increases the thermal stability of knee regions considerably. Deformation temperatures of nanotori considered are comparable with those of the carbon nanotubes and nanorods of similar cross section radius [79,80].

4.1.2.2 Carbon Nanogears

8 types of nanogears (4 being armchair yielding and 4 being zigzag yielding) were tested against increasing temperature. The armchair yielding ones were a 3-toothed nanogear with a C(9,0) nanotube acting as the pivot (total number of atoms in the system is 240); a 4-toothed nanogear with a C(12,0) pivot (total number of atoms in the system is 320); a 5-toothed nanogear with a C(15,0) pivot (total number of

atoms in the system is 400) and a 6-toothed nanogear with a C(18,0) pivot (total number of atoms in the system is 384). The corresponding values for the zigzag yielding nanogears are: a 3-toothed nanogear with a C(12,0) acting as the pivot (total number of atoms in the system is 252); a 4-toothed nanogear with a C(16,0) acting as the pivot (total number of atoms in the system is 336); a 5-toothed nanogear with a C(20,0) acting as the pivot (total number of atoms in the system is 420); a 6-toothed nanogear with a C(24,0) acting as the pivot (total number of atoms in the system is 504); The gear parts were made from G(4,4) protuberances for armchair yielding nanogears and from G(8,0) protuberances for zigzag yielding nanogears. The evolutions of the systems with respect to the increasing temperature were as follows:

3-toothed armchair nanogear was the longest lasting nanogear among the 4 armchair systems tried. The system's break down occurred at 2800 K. Until this temperature, the system vibrated but no new bonds formed and it was always possible to restore to the original form by means of optimization or a similar process like the cooling part of a stimulated annealing. System's snapshots at 1 K, 2700 K and 2800 K can be seen in Figure 42.

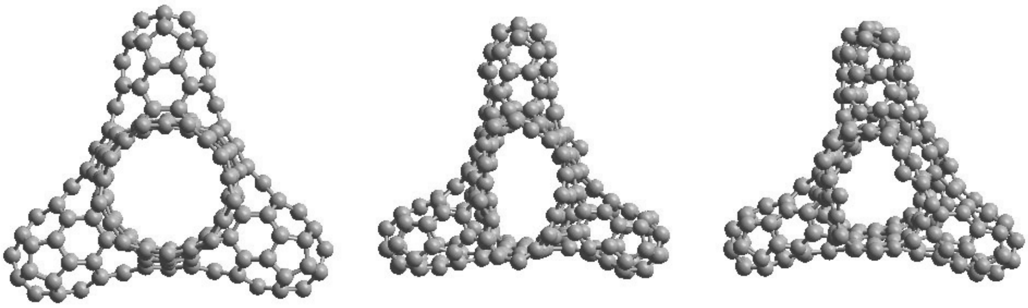


Figure 42: C_{240} (a) Relaxed structure at 1 K, (b) at 2700 K, (c) at 2800 K where the structure deforms.

4-toothed armchair nanogear imploded upon itself and then went unstable at 2600 K. The implosion began at 2300 K. At 2800 K bond breaking and atom separation started. Unlike the 3-toothed armchair nanogear, the deformed shape did not resemble its original form with the parabolic shaped regions between the teeth smoothed and gave way to valleys. This behavior can be observed in Figure 43.

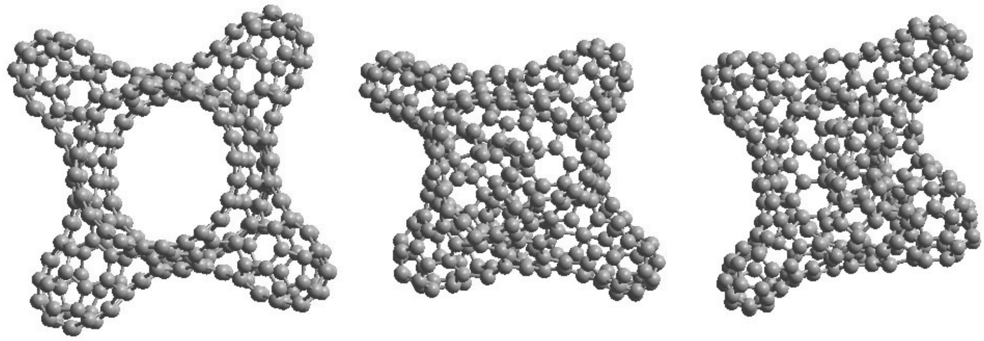


Figure 43: C_{320} (a) Relaxed structure at 1 K, (b) at 2700 K, (c) at 2800 K where the structure deforms.

5-toothed armchair nanogear retained its form but it imploded in the pivot acting nanotube region at 2400 K. Until that temperature, it retained its original form and the slight deformations were totally reversible as can be checked in Figure 44.

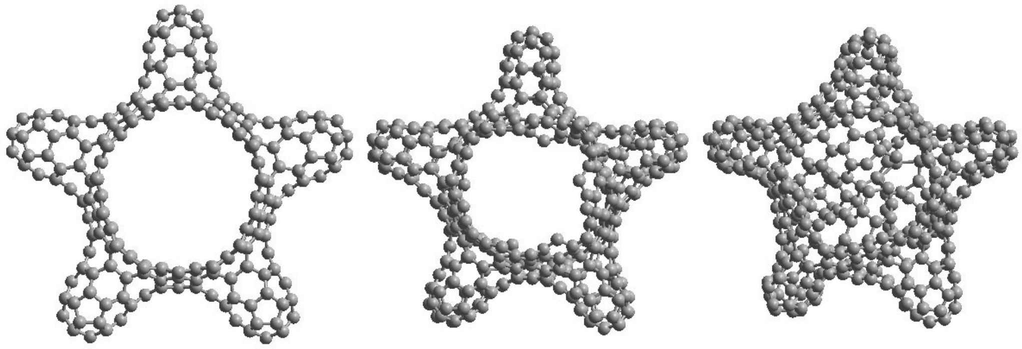


Figure 44: C_{400} (a) Relaxed structure at 1 K, (b) at 2300 K, (c) at 2400 K where the structure deforms.

Finally, among the armchair systems simulated, 6-toothed armchair nanogear also managed to retain its original form as shown in Figure 44 and its deformation was very regular that by inspecting it, one could think that by means of relaxation or geometry optimization, a reverse process towards its initial state can be achieved, but it has proven that this was not the case. The gear resisted the attempts to return it back to its optimized state.

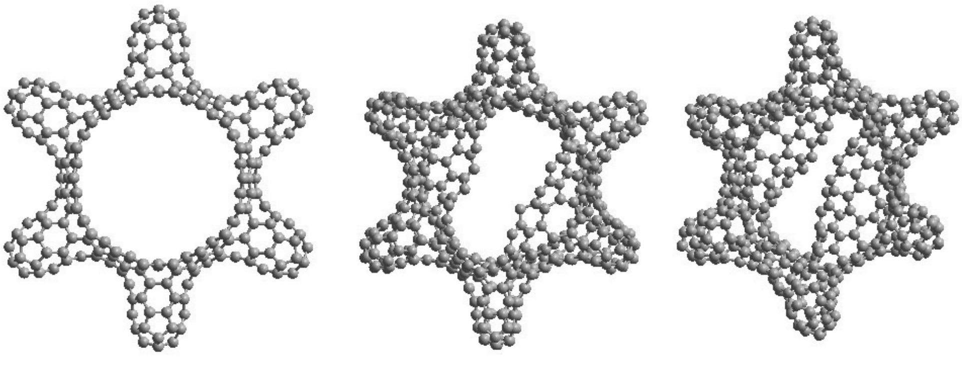


Figure 45: C_{480} (a) Relaxed structure at 1 K, (b) at 1600 K, (c) at 1700 K where the structure deforms.

The zigzag yielding nanogears used in the simulation had the same kind of nanotubes acting as their pivots as their armchair counterparts. The protuberances G(8,0) that formed the tooth regions had the same $\{o_i\}$ and $\{p_i\}$ series with those of the armchair yielding nanogear protuberances G(4,4) but the different arrangement of the pentagons and heptagons resulted in G(8,0) protuberances include more atoms (52 atoms for the generating cell) than the G(4,4) (48 atoms for the generating cell) thus the zigzag yielding nanogears had more atoms in their system compared to equivalent armchair nanogears.

The 3-toothed zigzag yielding nanogear's evolution snapshots with respect to temperature is presented in Figure 46. The system retains its shape but non-reversible (considering molecular mechanics) bond forming occur.

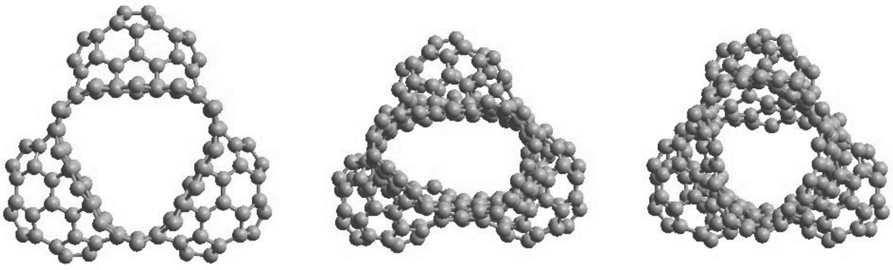


Figure 46: C_{252} (a) Relaxed structure at 1 K, (b) at 3000 K, (c) at 3100 K where the structure deforms.

In Figure 47, 4-toothed zigzag yielding nanogear is shown. Like its predecessor, this nanogear also retains the overall shape but due to new bond forming, fails to evolve back to its original form. Also, its deformation temperature is near to that of the 3-toothed G(8,0) bases nanogear.

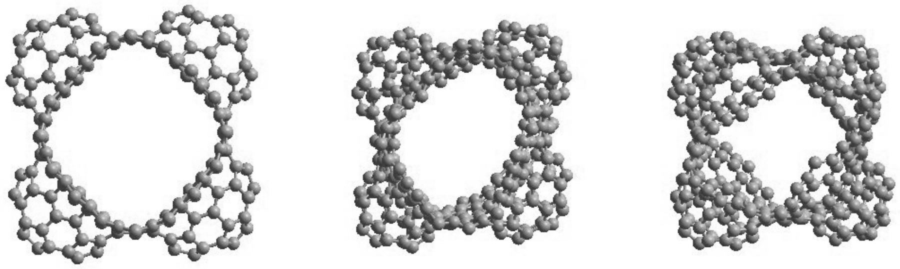


Figure 47: C_{336} (a) Relaxed structure at 1 K, (b) at 3100 K, (c) at 3200 K where the structure deforms.

The 5-toothed, zigzag yielding nanogear also retained its overall shape when it deformed at 2900K as shown in Figure 48. The system's gear locations remained stable with respect to the fluctuations.

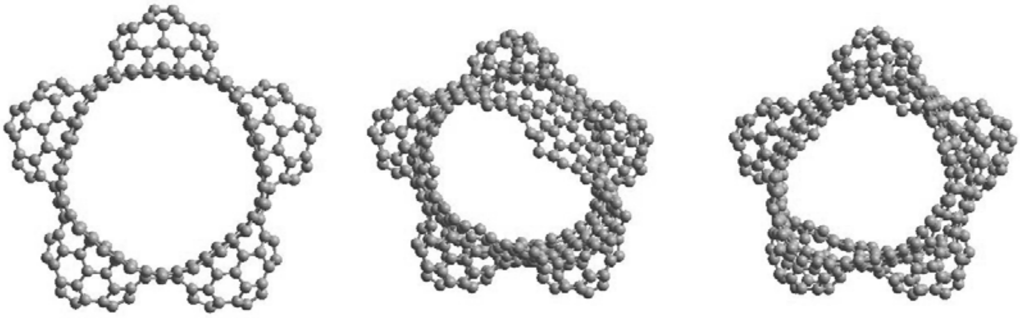


Figure 48: C_{420} (a) Relaxed structure at 1 K, (b) at 2800 K, (c) at 2900 K where the structure deforms.

The 6-toothed version of the G(8,0) based nanogear, resembling a Mandelbrot fractal, collapsed at a premature temperature compared to its predecessors due to deformations in the pivot region. This collapse resulted in the deformation of the near site located tooth regions as can be seen in Figure 49.

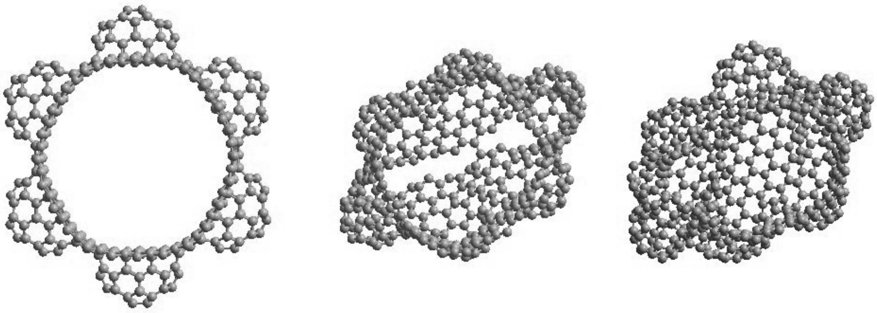


Figure 49: C_{504} (a) Relaxed structure at 1 K, (b) at 1100 K, (c) at 1200 K where the structure deforms.

In the above simulations, with the exception of the 4-toothed armchair nanogear, all the systems preserved their gear formation and the deformations occurred at the pivot regions, not at the tooth parts. This information suggests that with relatively long enough nanotubes acting as pivots, stability at much higher temperatures can be achieved. The curvature arising from the pentagons and heptagons strengthen the cage structure by means of stretching and relaxation in opposite directions with respect to the force applied. Since the tooth regions last longer than their pivot counterparts, these nanogears can safely be implemented into every system consisting of nanotubes.

When periodic boundary conditions (PBC) were applied through the azimuthal axis, we were expecting the systems to endure higher temperatures because, in the case of not applying PBC because the systems' deformations were occurring mostly at the pivot acting nanotube regions but it turned out that this was not the case. While armchair yielding carbon nanogears raised their critical temperature threshold by about 1000 K, the situation for the zigzag yielding carbon nanogears were very unappealing with the critical temperature threshold lowered by about 1500 K. The results for 3, 4 and 5 toothed nanogears with PBC applied are presented in Table 6

Table 6: The temperature thresholds for nanosystems, with periodic boundary conditions applied.

System:	**Deformation Temperature**
3 toothed armchair nanogear: C_{240}	3600 K
3 toothed zigzag nanogear: C_{252}	2000 K
4 toothed armchair nanogear: C_{320}	3700 K
4 toothed zigzag nanogear: C_{336}	1500 K
5 toothed armchair nanogear: C_{400}	3300 K
5 toothed zigzag nanogear: C_{420}	1100 K

4.1.3 Nanogear Interactions

The prepared nanogears were put to interact with each other. In this model, they were “mounted” on two carbon nanotubes acting as pivot. While one of the nanogears was unconstrained in every direction and open to interaction, the other one was controlled and rotated 1 degree per turn. Enough timesteps were spent for the relaxation after each rotation for the freely interacting nanogear to shift itself according to the interaction imposed on it by the controlled gear.

The first kind of interaction was chosen to be MM+ force field. The calculation consisted of solely the determination of the minimum energy yielding conformation. It was seen that, the nanogears responded well in this model as can be seen in the selected sequence of the run in Figure 50.

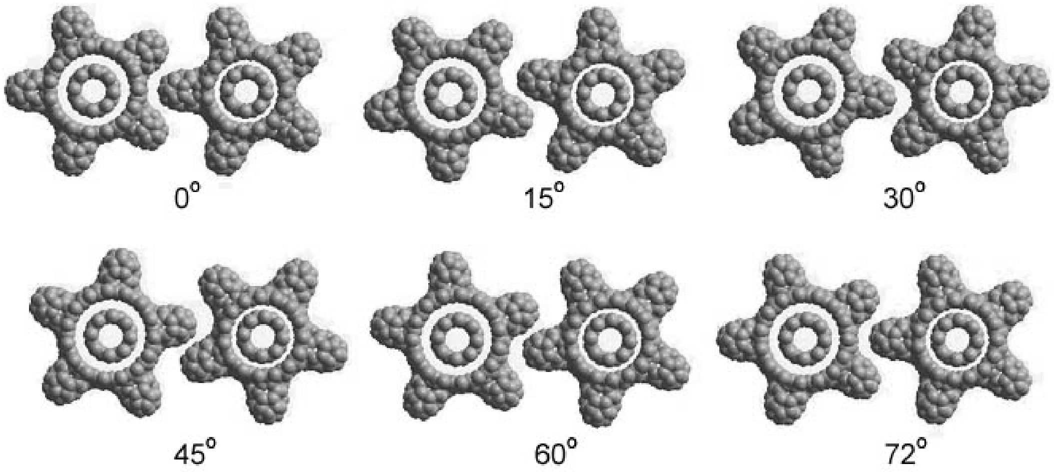


Figure 50: Selected sequence of the nanogear interaction simulation using MM+ run.

Since the molecular mechanics simulations are mostly used for deriving information about a possible formation of a given system, this simulation was far from having a relevance to that of a possible real life system. For this reason, the simulation was repeated another time using the Tersoff potential energy function for defining the interactions among the atoms and hence the systems. It was seen that due to an overlook on the properties of the potential energy function having no storage and usage of the bond information, making preference only according to the distance and angle, hence acting as a "sticky" potential for the situation which led to a very unsuccessful modeling as can be observed in the selected sequence of the run in Figure 51.

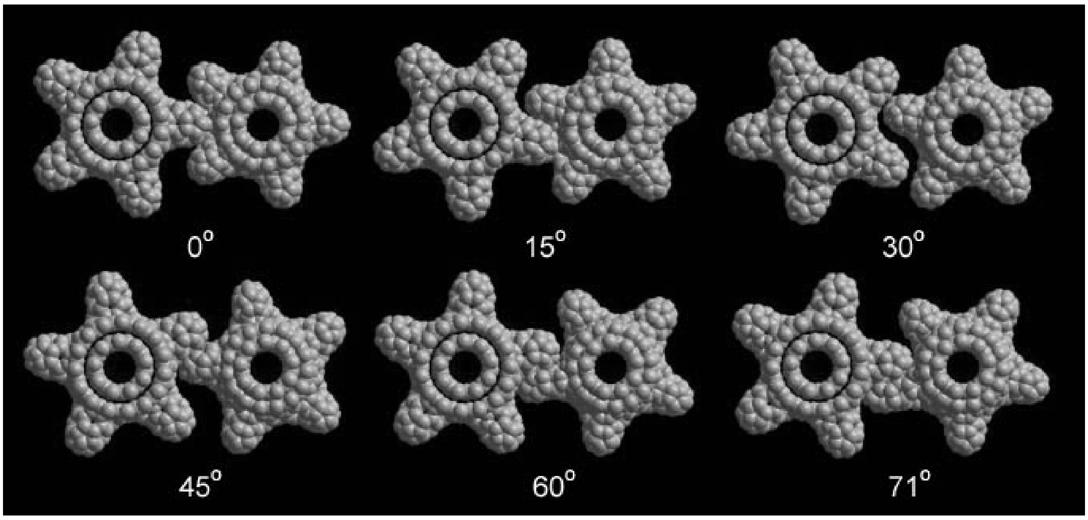


Figure 51: Selected sequence of the nanogear interaction simulation using only Tersoff PEF run.

To overcome this failure, we realized that, as a result of the already formed sp^2 bonds between the atoms of the same molecule, either in nanotubes or nanogears would show a very limited affinity towards new atoms because of the saturated state in molecular orbitals. Introduction of a new and weak interaction potential such as a Lennard-Jones type potential [81] to be used among the atoms of different molecules as was the case in Han's [82] nanogear simulations solved the problem. In this final simulation, the intramolecular atoms interacted via Tersoff potential while the intermolecular atoms interacted via Lennard-Jones potential. The result was successful as depicted in the selected sequence of the run in Figure 52.

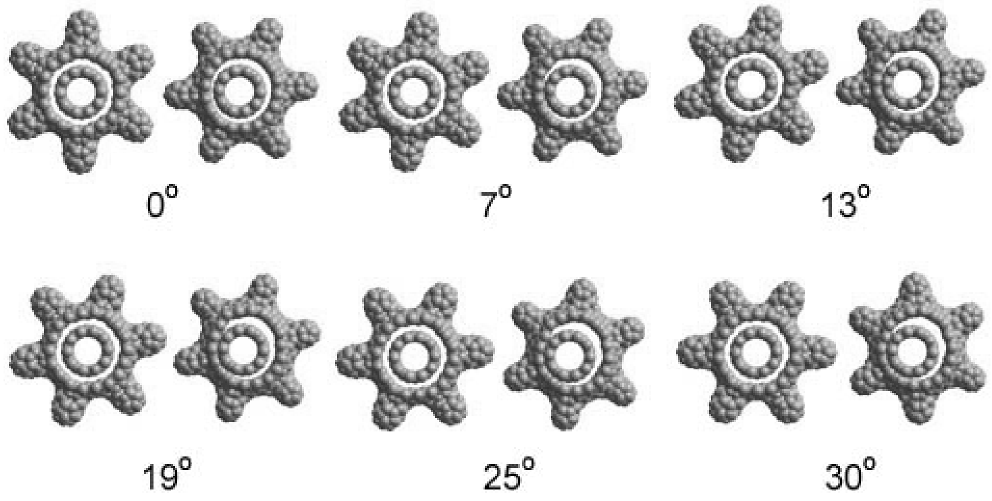


Figure 52: Selected sequence of the nanogear interaction simulation using Tersoff with L-J type PEF run.

4.2 Electronic Structure Calculations

In the electronic calculations, density of states (DOS), charge densities and the gap between the highest occupied molecular orbital (HOMO) and the lowest unoccupied molecular orbital (LUMO) that plays a fundamental role in conductivity were investigated for nanotori and nanogears. These properties were obtained by performing single-point energy calculations through the Extended-Hückel method with unweighted Hückel constant 1.75. The Extended-Hückel method calculates the energy eigenvalues for independent delocalized π electrons using an incidence matrix in which the diagonal elements are the binding energies of electrons and the off-diagonal elements are the interaction energies between two atomic orbitals. The particular method employed uses empirical values for the incidence matrix, thus the overall calculation is semi-empirical. The calculations for the optimized structures obtained from the algorithms derived above, were done using the HyperChem software [23].

The calculation results for the nanotori are listed in Table 7 and the nanogears' calculated properties can be found in Table 8.

Table 7: Geometrical parameters (*R*, *r*) and calculated energies of carbon nanotoroidal structures (at 0 K).

Toroid	*r* (Å)	*R* (Å)	*E* (eV)	HOMO (eV)	LUMO (eV)	Δ*E* (eV)
C_{170}	1.2	9.3	-11699.50	-9.726	-9.592	0.134
C_{250}	1.5	12.6	-17213.16	-9.550	-9.485	0.065
C_{360}	2.0	13.1	-25039.52	-9.786	-9.553	0.233
C_{520}	3.0	12.0	-36339.12	-10.076	-10.038	0.038
C_{750}	3.1	14.6	-52432.73	-10.085	-9.769	0.316

Table 8: Calculated energies of selected carbon nanogears (at 0 K).

Nanogear	*E* (eV)	HOMO (eV)	LUMO (eV)	Δ*E* (eV)
C_{240}	-16810.50	-10.944	-10.897	0.047
C_{320}	-22426.90	-10.862	-10.855	0.007
C_{400}	-28032.93	-10.972	-10.970	0.002
C_{480}	-33638.69	-10.969	-10.967	0.002

The DOS diagrams for the nanotori and nanogears are plotted in Figure 53 and Figure 54, respectively. It should be kept in mind that the electronic structure data obtained by the Extended Hückel method must be taken into account with caution. These electronic results can not be compared neither with experimental values, nor with the ab-initio method results.

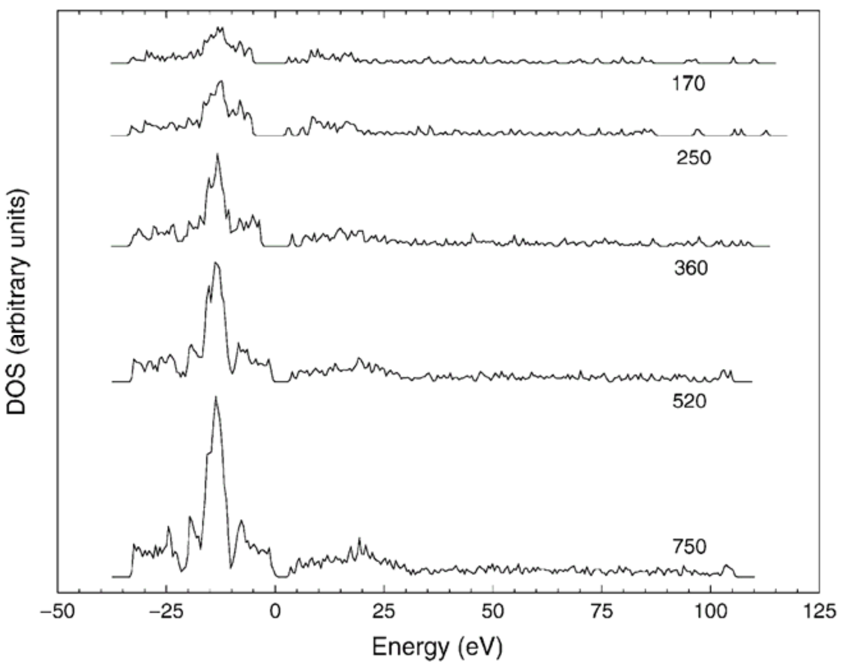


Figure 53: DOS distributions of the nanotori. Gaussian broadening has been used.

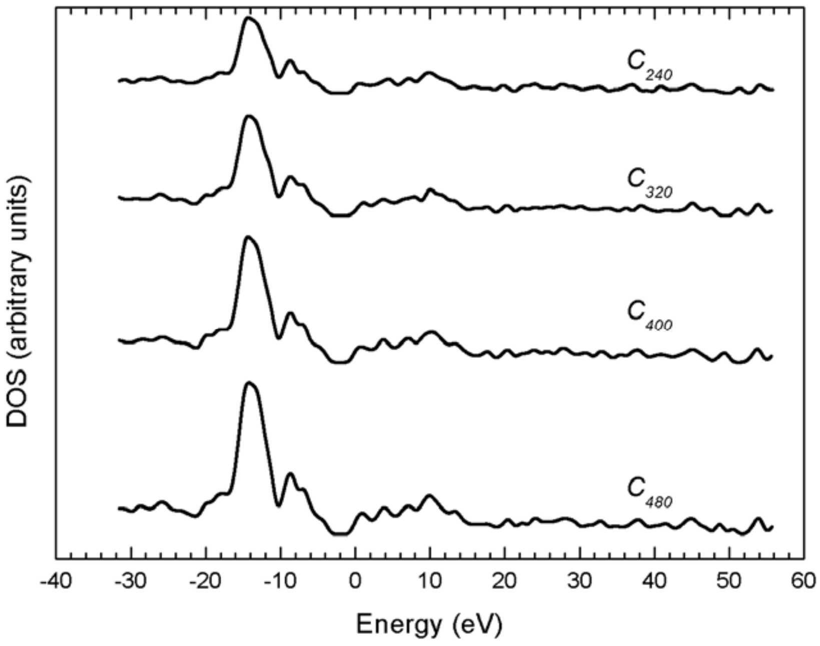


Figure 54: DOS distributions of the armchair nanogears. B-Spline method has been used.

For the systems considered, being nanotori and nanogears, the maximum distribution of the Hückel eigenvalues takes place at about -15 eV. Also, as it can be seen in the DOS plots, there is a gap located at around 0 eV. This gap tends to reduce as the number of atoms in the molecule increases for the nanotori but approximately stays the same for nanogears. This gap is also apparent in the work of Bai et al. [82] where partial eigenvalue sums for C_{240}, C_{480} and C_{960} are computed. This gap separates only the positive and negative eigenvalues into two groups. Both HOMO and LUMO eigenvalues take place in the same region (among the negative energies) of the DOS distribution and located at the bottom of the main peak (at about 10 eV).

The magnitude of the peak of the DOS increases as the number of atoms increase, as expected, which is an indication of the increasing number of the 'mobile' electrons. Since the system has closed geometry, the behavior of mobile electrons is a point of debate. One point of interest may be the contribution of these mobile electrons to the magnetic behaviour [83].

For the nanotori, the HOMO-LUMO gap values found in the present calculation, ranging from 0.04 to 0.032 eV. Hückel calculations [83] on different isomers of C_{240} showed that the HOMO-LUMO gaps range in 0 – 0.5 eV which is in good correspondence of the values encountered in our calculations.

The 2D and 3D excess charge density distributions for the nanotori are displayed in Figure 55. Charge distribution in the interior region of the larger toroids, C_{520} and C_{750}, is relatively less with respect to the rest of the toroids considered. From geometrical point of view, this is observed in the case r >2 Å. Furthermore, negative excess charge is localized mainly at the knee regions.

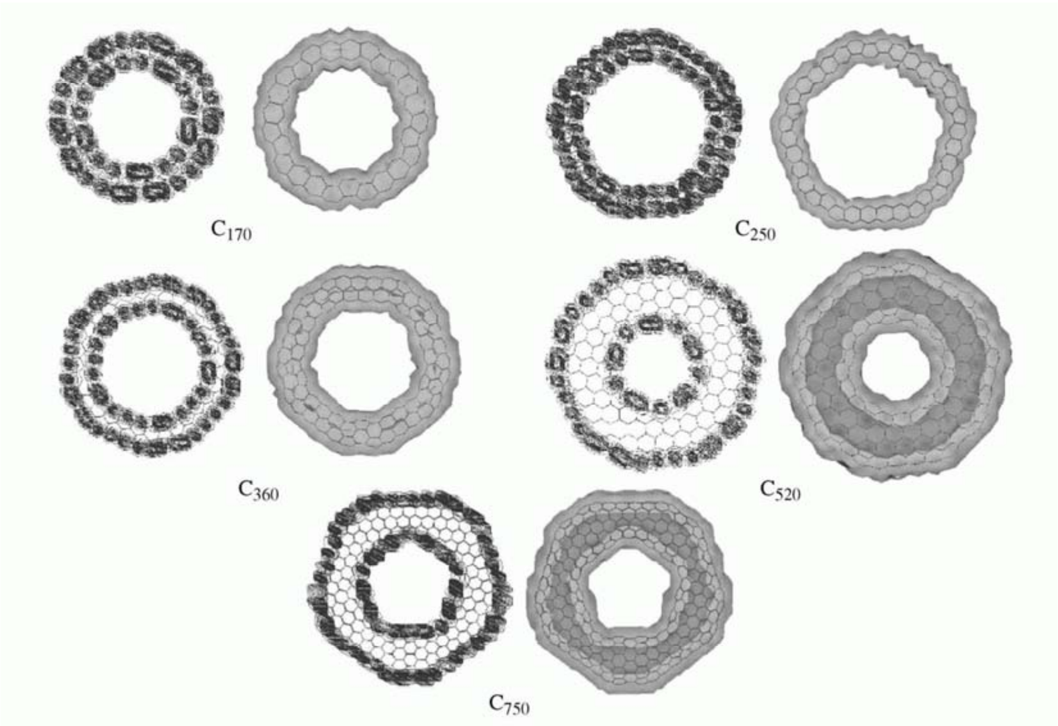


Figure 55: 2D and 3D charge distributions of the nanotori considered.

The 2D and 3D excess charge density distributions for the nanogears can be observed in Figure 56 and Figure 57. It can be seen that, for both types of nanogears, the charge distribution is mainly localized on the outer shell, acting as a screen for the inbound atoms.

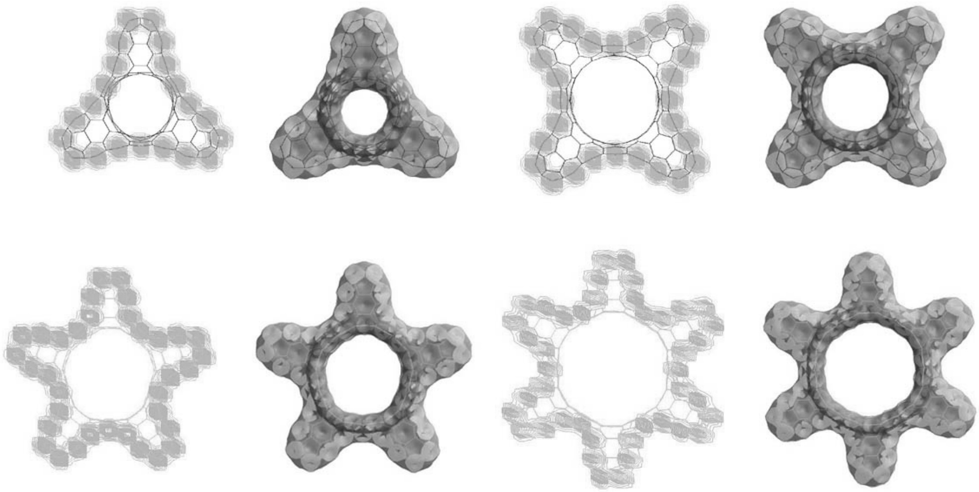


Figure 56: 2D and 3D charge distributions of the armchair type nanogears considered.

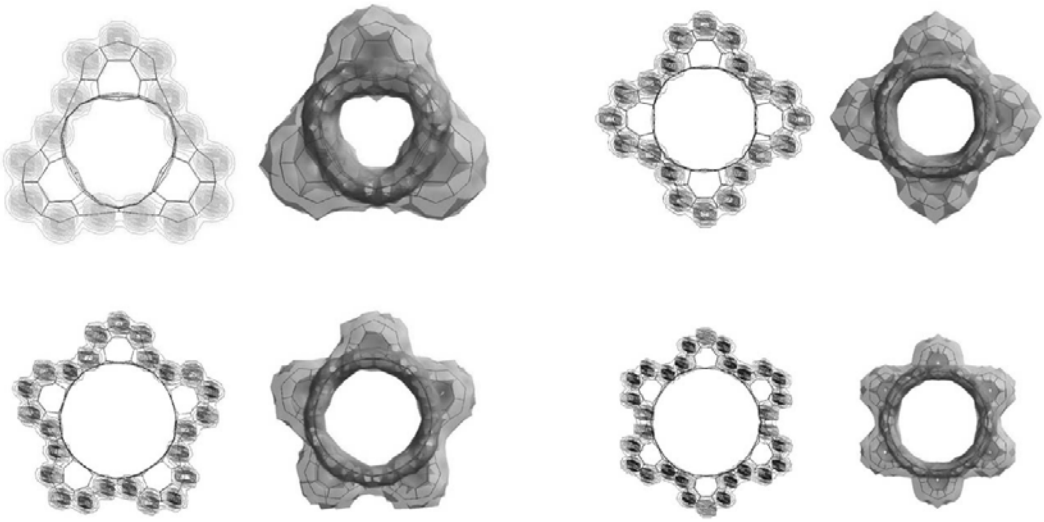


Figure 57: 2D and 3D charge distributions of the zigzag type nanogears considered.

CHAPTER 5

CONCLUSION

Throughout this thesis it was aimed to describe a structure analysis starting from the construction stage, then continuing with optimization and simulation processes. Carbon graphitic systems are the most probable structures among the candidates. Carbon atoms' ability to perform various hybridizations and the nanotubes' changing electronic conductance with respect to their geometrical properties as well as the graphite's hydrogen storage ability [84-87] assures that carbon will be the building block of the future.

Basing our assumptions on these facts, we have proposed novel carbon structures such as carbon nanotori and nanogears in accordance with their possible applications. After our investigations, we have seen that, they are endurable to well above daily environment temperatures and possess interesting electronic properties as is presented in charge density populations.

The industrial advancements are far from complete and at the moment the technological methods prevent us from practically manufacturing various systems but instead, most of the time we are left with the randomly mixed outcomes from mostly uncontrolled processes. Also, the production cost is heavy and this makes the decision to be carefully made in which system to produce mandatory. Because of this limitations, the candidates for production are first digitally synthesized and investigated in simulation and after that, if they have proven to be useful, they are passed to the fabrication process.

It was aimed in this thesis to be complete in ways that it supplies the intended reader with an insight about the tools, the methods and processes it employed. In this context, molecular mechanics and molecular dynamics methods have been revised before advancing to the thesis' original subject, being the "Generation of Nanostructures of Cage Structures" so that, afterwards when investigating the systems constructed by the proposed methods, mechanically and electronically, it was hoped that the reader would be given the opportunity to fully comprehend the ongoing processes.

Whether the system in question is a simple nano-mechanical (NMS) or a more advanced nano-electro-mechanical (NEMS) system, junctions play the crucial role of maintaining the relations between the components. For NMS, crude junctions like the one proposed in Section 2.1 can be used but NEMS require subtle junctions that are formed by means of joining via open-end nanogears. In the two of the recent articles by MacKinnon [88] and Schwab et al. [89], the re-coming of a mechanics era, this time in quantum mechanics, is strongly emphasized.

We are hoping in the near future to see the mechanical properties of the nanogears to come into play. Currently, the alternative fuels and related hydrogen storage is a promising subject and many devices are being designed to play part for these 'machines'. If we can supply appropriate means of systems that can directly benefit from the energy sources considering their nano-scale dimensions, the efficiency can be increased dramatically because of the less transportation load. Institues like the IMM [90] (Institute for Molecular Manufacturing) already began designing nanomotors competing with Drexler's dreams [15].

Some of the studies depicted in detail in this thesis has been summarized in article formats and has been published. The nanotori construction algorithm and their electronic properties are summarized in Ref. [91,92]; their stabilities under heat treatment are investigated in Ref. [93]; the junction formation in crossed nanotubes under pressure is presented in Ref. [94]; the construction algorithm for protuberances is included in Ref. [95].

REFERENCES

[1] Iijima, S. *Nature* **354**, 56 (1991).

[2] Wei, B.Q., Vajtai, R., Ajayan, P.M. *Appl. Phys. Lett.* **79**, 1172 (2001).

[3] Baughman, R.H. *Science* **290**, 1310 (2000).

[4] Cao, J., Wang, Q., Dai, H. *Phys. Rev. Lett.* **90**, 156401 (2003).

[5] Zhang, M., Fang, S., Zakhidov, A.A., Lee, S.B., Aliev, A.E., Williams, C.D., Atkinson, K.R., Baughman, R.H. *Science* **309**, 1215 (2005).

[6] Postma, H.W., Teepen, T., Yao, Z., Grifoni, M., Dekker, C. *Science* **293**, 76 (2001).

[7] Fink, R.L. *Recent advances in large-screen Carbon Nanotube Field Emisson Displays*. http://www.nano-proprietary.com/PDFs/CNT_TV_RFink.pdf; Last visited: 07/2007.

[8] Dillon, A.C., Jones, K.M., Bekkedahl, T.A., Kiang, C.H., Bethune, D.S., Heben, M.J. *Nature* **386**, 377 (1997).

[9] Lee, S.M., An, K.H., Lee, Y.H., Seifert, G., Frauenheim, T. *J. Am. Chem. Soc.* **123**, 5059 (2001).

[10] Hirscher, M., Becher, M., Haluska, M., Ouintel, A., Skakalova, V., Choi, Y.M., Dettlaff, U., Roth, W.S., Fink, J. *J. Alloys Comp.* **330**, 654 (2002).

[11] Leuking, A., Yang, P.T. *J. Catalysis* **206**, 165 (2002).

[12] Cheng, H.M., Yang, Q.H., Liu, C. *Carbon* **39**, 1447 (2001).

[13] Zuttel, A., Sudan, P., Mauron, P., Kiyobayashi, T., Emmeneger, C., Schlapbach, L. *Int. J. Hydrogen Energy* **27**, 203 (2002).

[14] Feynman, R. *There's Plenty of Room at the Bottom*. http://www.zyvex.com/nanotech/feynman.html; Last visited: 07/2007.

[15] Drexler, K.E. *Nanosystems: molecular machinery, manufacturing and computation*. USA : Wiley Interscience (1992).

[16] Chen, P., Wu, X., Lin, J., Tan, K.L. *Science* **285**, 91 (1999).

[17] Steiner, S.A., Baumann, T.F., Kong, J., Satcher, J.H., Dresselhaus, M.S. *Langmuir* **23**, 5161 (2007).

[18] Carroll, D.L., Redlich, P., Blasé, X., Charlier, J.C., Curran, S., Ajayan, P.M., Roth, S., Rühle, M. *Phys. Rev. Lett.* **81**, 2332 (1998).

[19] Oshima, Y., Onga, A. *Phys. Rev. Lett.* **91**, 205503 (2003).

[20] Mayo, S.L., Olafson, B.D., Goddard III, W.A. *J. Chem. Phys.* **94**, 8897 (1990).

[21] Allinger, N.L., Yuh, Y.H., Lii, J.H. *J. Am. Chem. Soc.* **111**, 8551 (1989).

[22] Allinger, N.L. *Molecular Mechanics*. USA : American Chemical Society (1982).

[23] Hypercube Inc. *Hyperchem 7*.5 Gainesville, FL, USA.

[24] Weiner, S.J., Kollman, P.A., Case, D.A., Singh, V.C. *J. Am. Chem. Soc.* **106**, 765 (1984).

[25] Jorgensen, W.L., Tirado-Rives, J. *J. Am. Chem. Soc.* **110**, 1657 (1988).

[26] Johnson, R.A. *Phys. Rev.* **134**, A1329 (1964).

[27] Tersoff, J. *Phys. Rev. B* **37**, 6991 (1988).

[28] Tersoff, J. *Phys. Rev. Lett.* **61**, 2879 (1988).

[29] Polak, E. *Computational Methods in Optimization - A Unified Approach*. USA : Academic Press (1971).

[30] Polak, E. *EE227B: Optimization Theory, Spring 2004 Lecture Notes*. http://www.eecs.berkeley.edu/~polak/EE227B.d/EE227B.html; Last visited: 07/2007.

[31] Hyman, R.A., Doporcyk, B., Tetzlaff, J. *Been Down So Long it Looks Like Up to Me: A Unified Derivation of Conjugate Gradient and Variable Metric Minimization*. http://arxiv.org/abs/physics/0307119; Last visited: 07/2007.

[32] Polak, E., Sargent, R.W.H., Sebastian, D.J. *J. Optimiz. Theory App.* **14**, 439 (1974).

[33] Stote, R., Dejaegere, A., Kuznetsov, D., Falquet, L. *CHARMM Molecular Dynamics Simulations Tutorial*. http://www.ch.embnet.org/MD_tutorial/pages/MD.Part1.html#Verlet; Last visited: 07/2007.

[34] Verlet, L. *Phys. Rev.* **159**, 98 (1967).

[35] Verlet, L. *Phys. Rev.* **165**, 201 (1967).

[36] Young, P. *Leapfrog method and other 'symplectic' algorithms for integrating Newton's laws of motion*. http://bartok.ucsc.edu/peter/115/leapfrog.pdf; Last visited: 07/2007.

[37] Ercolessi, F. *A Molecular dynamics primer*. http://www.fisica.uniud.it/~ercolessi/md/md/; Last visited: 07/2007.

[38] Heermann, D.W. *Computer Simulation Methods in Theoretical Physics*. Germany : Springer-Verlag (1990).

[39] Haile, J.M. *Molecular Dynamics Simulation*. USA : John-Wiley & Sons, Inc. (1997).

[40] Erkoç, Ş. *Molecular Dynamics Program For Cluster Simulations (md-tpc-pbc.f)* METU, TR.

[41] Springborg, M. *Methods of Electronic-Structure Calculations: From molecules to solids*. England : John-Wiley & Sons, Ltd. (2000).

[42] Hoffmann, R. *J. Chem. Phys.* **39**, 1397 (1963).

[43] Hoffmann, R. *J. Chem. Phys.* **40**, 2047 (1964).

[44] Hückel, E. *Z. Phys.* **70**, 204 (1931).

[45] Hückel, E. *Z. Phys.* **72**, 310 (1931).

[46] Hückel, E. *Z. Phys.* **76**, 628 (1932).

[47] Roothaan, C.C.J. *Rev. Mod. Phys.* **23**, 69 (1951).

[48] Kroto, H.W., Heath, J.R., Obrien, S.C., Curl, R.F., Smalley, R.E. *Nature* **318**, 162 (1985).

[49] Johnson, J.K., Davidson, B.N., Peterson, M.R., Broughton, J.Q. *Phys. Rev. B* **50**, 17575 (1994).

[50] Itoh, S., Ihara, S., Kitakami, J. *Phys. Rev. B* **47**, 1703 (1993).

[51] Itoh, S., Ihara, S. *Phys. Rev. B* **48**, 8323 (1993).

[52] Itoh, S., Ihara, S. *Phys. Rev. B* **49**, 13970 (1994).

[53] Ihara, S., Itoh, S., Kitakami, J. *Phys. Rev. B* **47**, 12908 (1993).

[54] Ihara, S., Itoh, S., Kitakami, J. *Phys. Rev. B* **48**, 5642 (1993).

[55] Hod, O., Rabani, E., Baer, R. *Phys. Rev. B* **67**, 195408-1 (2003).

[56] Chernozatonskii, L.A. *Phys. Lett. A* **170**, 37 (1992).

[57] Meunier, V., Lambin, P.H., Lucas, A.A. *Phys. Rev. B* **57**, 14886 (1998).

[58] Setton, R., Setton, N. *Carbon* **35**, 497 (1997).

[59] Berger, J., Avron, J.E. *J. Chem. Soc. Faraday Trans.* **91**, 4037 (1995).

[60] Diudea, M.V., Silaghi-Dumitrescu, I., Pary, B. *Internet Electron. J. Mol. Des.* **1**, 10 (2002).

[61] Fonseca, A., Hernadi, K., Nagy, J.B., Lambin, P.H., Lucas, A.A. *Carbon* **33**, 1759 (1995).

[62] Avron, J.E., Klein, M., Pnueli, A., Sadun, L. *Phys. Rev. Lett.* **69**, 128 (1992).

[63] Shapere, A., Wilczek, F. *Geometric Phases in Physics*. Singapore : World Scientific (1989).

[64] Balasz, N., Voros, A. *Phys. Rep.* **143**, 103 (1986).

[65] IN-VSEE, *World of Carbon*. http://invsee.asu.edu/nmodules/Carbonmod/bonding.html; Last visited: 07/2007.

[66] Research Collaboratory for Structural Bioinformatics, *RCSB Protein Data Bank*. http://www.pdb.org; Last visited: 07/2007.

[67] RCSB, *Protein Data Bank Contents Guide: Atomic Coordinate Entry Format Description v2.3*. http://www.wwpdb.org/documentation/format2.3-0108-a4.pdf; Last visited: 07/2007.

[68] Schmidt, O.G., Deneke, C., Kiravittaya, S., Songmuang, R., Heidemeyer, H., Nakamura, Y., Zapf-Gottwick, R., Muller, C., Jin-Phillipp, N.Y. *IEEE J. Sel. Top. Quant.* **8**, 1025 (2002).

[69] Ferguson, A.J., Cain, P.A., Williams, D.A., Briggs, G.A.D. *Phys. Rev. A* **65**, 034303 (2002).

[70] Ionicioiu, R., Amaratunga, G., Udrea, F. *Int. J. Mod. Phys. B* **15**, 125 (2001).

[71] Noid, D.W., Sumpter, B.G., Tuzun, R.E. *Dynamics and Friction in Submicrometer Confining System (ACS Symposium Series 882) eds. Y. Braiman,*. USA : American Chemical Society (2004).

[72] Han, J., Globus, A., Jaffe, R., Deardorff, G. *Nanotechnology* **8**, 95 (1997).

[73] Zsoldos, I., Kakuk, G., Reti, T., Szasz, A. *Model. Simul. Mater. Sci. Eng.* **12**, 1251 (2004).

[74] Zsoldos, I., Kakuk, G., Janik, J., Pek, L. *Diamod Relat. Mater.* **14**, 763 (2005).

[75] An, B., Fukuyama, S., Yokogawa, K., Yoshimura, M., Egashira, M., Korai, Y., Mochida, I. *Appl. Phys. Lett.* **78**, 3696 (2001).

[76] Beuneu, F. *Solid State Commun.* **136**, 462 (2005).

[77] Weaire, D., Rivier, N. *Contemp. Phys.* **25**, 59 (1984).

[78] Erkoç, Ş, Vural, D.C. *Int. J. Mod. Phys. C* **12**, 685 (2001).

[79] Erkoç, Ş *Int. J. Mod. Phys. C* **11**, 1247 (2000).

[80] Malcıoğlu, O.B., Erkoç, Ş *Int. J. Mod. Phys. C* **14**, 441 (2003).

[81] Xue, Y., Chen, M. *Nanotechnology* **17**, 5216 (2006).

[82] Bai, Z., Fahey, M., Golub, G., Menon, M., Richter, E. *Computing Partial Eigenvalue Sum in Electronic Structure Calculations - SCCM-98-03*. http://www-sccm.stanford.edu/nf-publications-tech.html#start-1998; Last visited: 07/2007.

[83] Liu, L., Guo, G.Y., Jayanthi, C.S., Wu, S.Y. *Phys. Rev. Lett.* **88**, 217206 (2002).

[84] Lueking, A.D., Yang, R.T., Rodriguez, N.M., Baker, R.T.K. *Langmuir* **20**, 714 (2004).

[85] Lachawiec, A.J., Oi, G.S., Yang, R.T. *Langmuir* **21**, 11418 (2005).

[86] Yang, F.H., Lachawiec, A.J., Yang, R.T. *J. Phys. Chem. B* **110**, 6236 (2006).

[87] Ahn, C.C., Vajo, J.V., Fultz, B., Yazami, R., Brown, D.W., Bowman, R.C. *Hydrogen Storage in*

Metal-Modified Single-Walled Carbon Nanotubes.

http://www1.eere.energy.gov/hydrogenandfuelcells/pdfs/32405b30.pdf; Last visited: 07/2007.

[88] MacKinnon, A. *Physica E* **29**, 399 (2005).

[89] Schwab, K., Roukes, M. *Phys. Today* **7**, 36 (2005).

[90] Institute for Molecular Manufacturing, *Nanotechnology*. http://www.imm.org/; Last visited: 07/2007.

[91] Yazgan, E., Taşcı, E., Erkoç, Ş *Int. J. Mod. Phys. C* **15**, 267 (2004).

[92] Yazgan, E., Taşcı, E., Malcıoğlu, O.B., Erkoç, Ş *J. Mol. Struc. (THEOCHEM)* **681**, 231 (2004).

[93] Taşcı, E., Yazgan, E., Malcıoğlu, O.B., Erkoç, Ş *Full. Nanot. Carb. Nanostruct.* **13**, 147 (2005).

[94] Taşcı, E., Malcıoğlu, O.B., Erkoç, Ş *Int. J. Mod. Phys. C* **16**, 1371 (2005).

[95] Taşcı, E., Erkoç, Ş *J. Nanosci. Nanotechnol* **7**, 1653 (2007).

APPENDIX

Two ad-hoc methods for generating the series

These two methods are developed from the results of Table 3 and Table 4 and successfully generate the series values but caution must be taken if they are to be applied for values of *(n+m)*>26. The first method takes *n* and o_0 values as input parameters whereas the second method takes only the *n* value as input. '**%**' is the remainder operator and 'ip(*x*)' is a function returning the integer part of a rational number *x*. To obtain the series for armchair yielding nanogears, *n* must be set to *(n+m)* as proposed in Section: 3.3.2.2. The second method is proposed by the anonymous refree for the related article [93].

Method 1

```
if ($n$ % 2 = 1) $o_{k+1} = o_0 + 1$
    else $o_{k+1} = o_0$

if ($n$ % 3 = 0) $k=(n - 3) / 3$
    else if ($n$ % 3 = 1) $k=(n - 1) / 3$
    else $k=(n - 5) / 3$
```

$k_+ = (o_{k+1} - o_0 + k + 1)/2$ and $k_- = k - k_+ + 1$ as usual.

Method 2

```
if ($n$ % 6 = 0) $o_0=o_{k+1} = (n / 6) + 1$
    else if ($n$ % 6 = 1) $o_0 = ip(n / 6)$; $o_{k+1} = o_0 + 1$
    else if ($n$ % 6 = 2) $o_0=o_{k+1} = ip(n / 6) + 2$;
    else if ($n$ % 6 = 3) $o_0 = ip(n / 6) + 1$; $o_{k+1} = o_0 + 1$
    else if ($n$ % 6 = 4) $o_0=o_{k+1} = ip(n / 6) + 1$;
    else if ($n$ % 6 = 5) $o_0 = ip(n / 6) + 1$; $o_{k+1} = o_0 + 1$
```

$k = (n - o_0 - o_{k+1}) / 2$

$k_+ = (o_{k+1} - o_0 + k + 1)/2$ and $k_- = k - k_+ + 1$ as usual.

CURRICULUM VITAE

Emre Taşcı was born in Istanbul in 1977. Studied theoretical physics (quantum field theory) under the advisorship of Assoc. Prof. Dr. Neşe Özdemir for his B.S. at the Department of Physics Engineering at Istanbul Technical University where he graduated in 2000. He then came to the Physics Department at Middle East Technical University for M.S. (graduated in 2002) and Ph.D. programme where he studied molecular physics under the advisorship of Prof. Dr. Şakir Erkoç. He has since been working as a research assistant and computer coordinator at the department.

Since 2002, he is married to Bengü Yazıcıoğlu Taşcı. In 2006, their daughter Ece was born.

More information on his studies, publications, activities and life can be accessed via his blog site at http://www.emresururi.com (in Turkish) and his 'more scientific' personal web page at http://www.emresururi.com/etasci/ (in English).