Predictive Model of Cap Formation for Carbon Nanotube Synthesis via Chemical Vapour Deposition

Patrick Bowe

Supervisor: Barry Cox, Michael Chen

May 2021

Thesis submitted for the degree of Masters in 2021

SCHOOL OF MATHEMATICAL SCIENCES

THE UNIVERSITY
OF ADELAIDE
AUSTRALIA

Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint award of this degree.

I give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship.

Signed by student:

Date: 11/5/2021

Signed by supervisor:

Date: 12/5/2021

Acknowledgements

I would first like to thank the University of Adelaide for awarding me the Master of Philosophy (No Honours) scholarship and allowing me to pursue a postgraduate degree.

Finally I would like to thank my supervisors, Dr Barry Cox and Dr Michael Chen, for their insight and support throughout my research and the writing of my thesis. Their help over the course of my degree has been invaluable to me.

Abstract

Since their discovery in 1993, carbon nanotubes have been studied for their unique properties which make them an attractive option for many potential applications. Often such applications require the use of nanotubes with specific chiralities and therefore properties. Because of this, methods by which the structure of carbon nanotubes can be controlled during synthesis are desirable. Such methods may represent a significant decrease in the cost of producing carbon nanotubes for new technologies.

In this thesis we develop a model for the growth and dewetting of graphene caps via chemical vapour deposition. This model will allow us to predict the conditions which favour the production of carbon nanotubes with a specific chirality.

To model the growth of graphene caps wetted to a metallic catalyst particle, we use the calculus of variations to determine the optimal configuration of the system for various cap sizes. The growth of graphene caps is simulated by treating their surface area as a time-like variable. We use this model to predict when dewetting from the metallic catalyst particle becomes energetically favourable by comparing the optimal energies of the wetted and dewetted states. When dewetted, the structure of the graphene cap determines the structure of the resultant nanotube.

Testing our model on three metallic catalysts we found that our model consistently predicted lower CNT diameters than those observed in experiments by other researchers. For iron carbide we predicted diameters of 6.5–7.1 Å compared to 30 Å in experiments, and for gold we predicted diameters of 7–9 Å compared to 9–18 Å

in experiments. For nickel carbide our model predicted no CNT production despite it being a common catalyst material in CVD. Our model predicted that iron carbide catalyst particles would produce more metallic CNTs compared to gold, with 50% of CNTs produced using iron carbide predicted to be metallic compared to 33–46% of CNTs produced using gold.

Due to the inaccuracy of our model when compared to experimental results, we recommend that future research focuses on improving the model. This could be achieved by reducing the amount or restrictiveness of our assumptions, or increasing the detail with which our model accounts for various factors of the cap growth and dewetting processes.

Contents

Chapter 1

Introduction

Solid state carbon has a wide variety of allotropes, the most widely known being diamond and graphite. While diamond has a uniform 3-dimensional structure, graphite comprises many layered sheets of graphene each of which is effectively a 2-dimensional carbon structure. The study of carbon allotropes led to the discovery of the C_{60} fullerene in 1985 [1] which is one example of a general fullerene structure. Fullerenes can be classified as allotropes of carbon with a hollow pseudo-spherical geometry consisting of only pentagonal and hexagonal carbon rings. We can further classify fullerenes according to the number of carbon atoms in their structure with the smallest fullerene, C_{20}, having only 20 carbon atoms arranged into twelve pentagonal rings [2]. Furthermore, as a result of Euler's polyhedron formula all fullerenes have exactly twelve pentagonal rings regardless of size [3]. The number of valid fullerenes which can be constructed using a certain number of carbon atoms increases rapidly with the number of atoms. For example, it has been calculated that C_{60} has 1812 isomers [4] and C_{100} has 285,914 isomers [5]. In 1993, a new form of carbon known as carbon nanotubes (CNTs) was discovered [6, 7]. This form of carbon in particular has a wide range of useful properties which has motivated substantial research into the synthesis and use of CNTs.

CNTs are cylindrical structures of carbon which resemble a rolled up sheet of

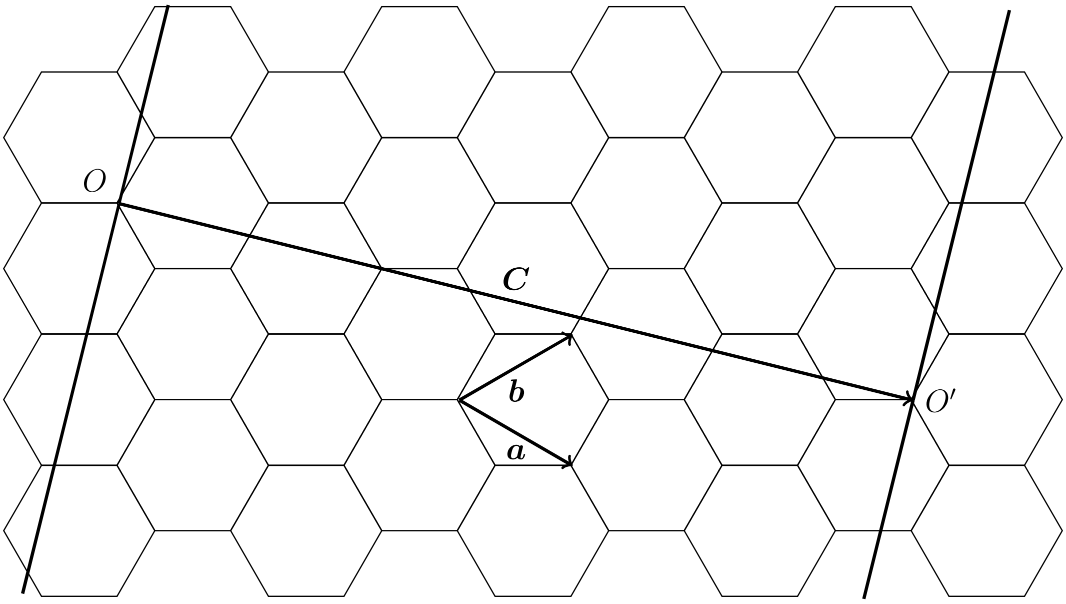

Figure 1.0.1: Diagram of a graphene sheet with lattice vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. A CNT can be produced by folding the sheet such that O and O' meet

graphene. CNTs can be capped, being closed at one or both ends, or uncapped. They can also be classified into single-walled or multi-walled CNTs, where multi-walled CNTs are effectively a nested group of single-walled CNTs. Furthermore, all single-walled CNTs can be classified according to their chirality. The chirality of a CNT can be represented by a pair of integers of the form (n, m) which are determined by its chiral vector, $\boldsymbol{C} = n\boldsymbol{a} + m\boldsymbol{b}$ [8]. This is illustrated in Fig. 1.0.1 where $\boldsymbol{C}$ is the vector joining O and O'. In addition to classifying CNTs, the chirality also determines the physical properties. One important property of CNTs is their electrical conductivity, since they can be either conductors or semiconductors. If $n - m$ is divisible by 3 then the CNT is metallic, or a conductor, otherwise it is a semiconductor [9, 10]. Furthermore, the band gap of a semiconducting CNT is inversely proportional to its diameter [11].

The various properties of CNTs make them attractive for a wide range of potential applications. For example, their high tensile strength has seen them proposed as a material for constructing space elevators [12]. Zhang et al. have been successful in creating high strength transparent sheets made up of single or multi-walled

CNTs [13]. The electrical conductivity of these sheets could see them incorporated in windows or other materials for uniform heating. Many other potential applications require a population of CNTs which has a uniform set of properties, for example, room-temperature CNT transistors require all semiconducting CNTs [14]. Achieving this with current synthesis methods requires sorting out the required nanotubes from a more broad population of synthesis output.

Given a particular population of CNTs, there are a wide range of techniques which can be used to separate them according to a particular physical property. Collins et al. found that semiconducting CNTs could be separated from an assortment of nanotubes via the application of an electrical current which oxidises the metallic CNTs [15]. This destroys the metallic CNTs leaving only semiconducting CNTs. Chattopadhyay et al. also found that metallic and semiconducting CNTs could be separated, this time by treating CNTs with octadecylamine and dissolving them in tetrahydrofuran [16]. This results in the precipitation of metallic CNTs, allowing them to be separated from the semiconducting CNTs. Unlike the previous method, both types of CNTs are preserved throughout the separation process. A method for sorting nanotubes by diameter, bandgap, and electronic type was outlined by Arnold et al. which utilised a variety of surfactants to alter the buoyant density of specific CNTs [17]. This allowed them to separate the required CNTs using density-gradient ultracentrifugation. This method, however, does not perfectly separate CNTs and requires repetition of the centrifugation process to increase the purity of the separated CNTs. Arnold et al. reported that for $(6,5)$ CNTs they achieved a solution in which greater than 97% of CNTs were within 0.2 Å of the required radius after three iterations. Overall this method gives a greater degree of control over the population of CNTs produced but producing pure samples of the desired CNT can be costly and time consuming due to the required repetition. For any method of separation, the yield of the required CNTs is ultimately limited to the amount synthesised initially. Therefore it is desirable to synthesise CNT populations with a greater proportion of the product having the desired chirality.

CNTs can be synthesised via various methods such as arc discharge [18], chemical vapour deposition (CVD) [19], high-pressure carbon monoxide disproportionation [20], and laser ablation [21]. In order to develop a model of CNT synthesis, we will consider CVD synthesis in particular. CVD is a method for CNT synthesis in which a carbon-bearing gas is heated to a high temperature in the presence of metallic catalyst particles. Some examples of carbon-bearing feedstock are carbon monoxide (CO) [22], methane (CH_4) [23], and ethanol (C_2H_6O) [24]. Typically, the metallic catalyst particles are very small with nanoscale (1–100 nm) diameters [25, 26], and are made up of transition metals such as iron, nickel, and cobalt [27]. Carbon bearing molecules from the feedstock gas interact with metallic catalyst particles such that the carbon is dissociated. The dissociated carbon is then adsorbed onto the catalyst, a process which eventually results in a hemispherical graphene cap forming over the catalyst particle [28]. Helveg et al. observed that nickel catalyst particles would deform such that they fill the inside of a growing CNT, until at some point the catalyst particle contracts back to a roughly spherical shape [29]. At this point the graphene cap has lifted off from the particle to form a capped CNT which continues to grow from the catalyst over time. The graphene cap and metallic catalyst system is referred to as “wetted” before cap lift off, and after lift off is referred to as “dewetted”. It is possible to produce both single and multi-walled CNTs using CVD, however, since single-walled CNTs are of particular interest for many applications we will focus our modeling on producing those CNTs. CVD is not uniquely selective [28, 19], thus CNTs with various chiralities and therefore various properties are usually produced.

The CVD process has many controllable parameters such as furnace temperature, gas pressure, catalyst particle size and composition and feedstock composition. Research into catalyst composition has had some limited success in finding conditions under which CNTs with specific chiralities are predominantly produced. Yang et al.

found that using catalysts made up of tungsten-based bimetallic alloy nanocrystals would result in greater than 92% of CNTs having $(12, 6)$ chirality [30]. Additionally, Lolli et al. found that a cobalt molybdenum catalyst on a magnesium oxide support results in the production of a small range of CNTs with specific chiralities [31]. The same authors also found that increasing the temperature of the process results in CNTs with larger radii. Hiraoka et al. reported that increasing the pressure or the size of the catalyst particle would similarly result in the production of CNTs with larger radii [32].

However, simply controlling the radius of CNTs is not sufficient to select for specific chiralities. This is because the radius of a CNT is determined by its chirality, and is proportional to $\sqrt{n^2 + nm + m^2}$ [33]. Because of this, pairs of CNTs with similar radii can have different chiralities. For example, $(6, 1)$ and $(5, 2)$ CNTs have very similar radii but the $(5, 2)$ CNT is a conductor whereas the $(6, 1)$ CNT is a semiconductor. With this in mind, the research into catalyst shapes and compositions has been the most successful so far at achieving the goal of producing large quantities of CNTs with specific chiralities, and therefore properties [30, 31]. However, research into other parameters may lead to the discovery of more favourable conditions for the production of CNTs with the chiralities selected for by the choice of catalyst. Because direct experimentation of the effects of various parameters is costly and time-consuming, models of CNT production via CVD may provide useful insights by identifying combinations of parameters that are likely to give the desired control over resulting chirality.

To produce a model of CNT production via CVD we consider how the graphene cap evolves on an individual catalyst particle. Existing research into the growth of graphene caps on catalyst particles typically utilise molecular dynamics simulations or density functional theory [34, 35]. Instead, we build on the work of Morton in which the dewetting of graphene caps is modelled by comparing the energies of the wetted and dewetted states [36]. This is achieved by first deriving expressions for the energy of the graphene cap and metallic catalyst system for both the wetted and dewetted states. We then use these expressions to determine the shape of the system by finding its minimum energy configuration, since physical systems tend to seek configurations in which its energy is minimised. Comparing the minimum energy state of both the wetted and dewetted states will give insights into when dewetting becomes energetically favourable, and this in turn will influence the expected structure of the resulting CNT. We expand on Morton's work by allowing the contact angle between the graphene cap and metallic catalyst particle to vary as the graphene cap grows. We also take a different approach to the effect of graphene cap curvature on the energy of the system whereby we consider the energy of interatomic bonds within the graphene cap.

Chapter 2

Base model for graphene cap growth

In order to develop a model for graphene cap growth, we begin with an energy minimisation problem to determine the optimal shape of the graphene cap and metallic catalyst particle system according to the surface free energy. We expect the physical cap-particle system to adopt the shape that leads to the minimum energy state because physical systems tend to be governed by the principle of minimum potential energy, and the interaction energies within the cap-particle system are much greater than its thermal energy. This base model may then be expanded with the addition of other energy terms to produce a more realistic model of graphene cap growth. The broad view goal in developing this model is to ultimately predict when dewetting will occur under specific conditions, and the cap-particle system conformation if and when it does.

2.1 Cap-particle geometry

The cap-particle system that we consider here comprises an unsupported metallic catalyst particle and a curved graphene cap resting on the particle's surface. Since the catalyst particle has a nanoscale diameter, we assume that the effects of gravity are negligible compared to the interatomic forces within the system. Because of this we may assume that the cap-particle system comprises two spherical caps as shown in Fig. 2.1.1. The thick black line represents the graphene cap while the cross hatched region corresponds to the metallic catalyst particle which, when wetted, extends to the inside of the graphene cap. The graphene cap is assumed to be a spherical cap with radius r_1, height h_1, and a circular edge with radius a. The volume of the metallic catalyst particle which is not enclosed by the graphene cap is referred to as the exposed metallic catalyst particle, and it is a spherical cap with radius r_2 and height h_2.

The volume of the metallic catalyst particle enclosed by the graphene cap is V_1, while the volume of the exposed metallic catalyst particle is V_2. We assume that the total volume of the metallic catalyst particle is constant and equal to V_{m}, so we have

$$V_{\mathrm{m}} = V_1 + V_2. \tag{2.1.1}$$

The values of r_1, r_2, h_1, h_2, and a are determined by the minimum energy state of the cap-particle system. By prescribing values for the surface area of the graphene cap, A_{g}, we can model the conformation of the system as the graphene cap grows.

2.2 Surface free energy

We formulate an initial model for the energy of the cap-particle system using the surface free energy of the system. The surface free energy is determined by the surface area, A, and energy per unit area, γ, of each interface between different components of the system. This is given by

$$E = \gamma_{\mathrm{gm}} A_{\mathrm{gm}} + \gamma_{\mathrm{ga}} A_{\mathrm{ga}} + \gamma_{\mathrm{ma}} A_{\mathrm{ma}}, \tag{2.2.1}$$

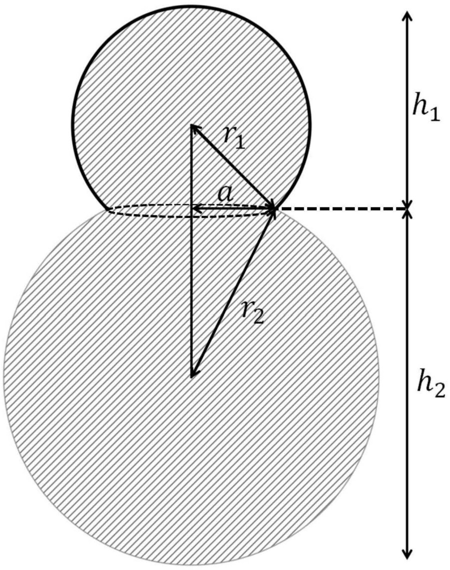

Figure 2.1.1: Geometry of cap-particle system

where the subscripts gm, ga, and ma refer to the graphene-metal, graphene-atmosphere, and metal-atmosphere interfaces respectively. The value of γ_{gm} is related to γ_{ga} and γ_{ma} by Young's equation,

$$\gamma_{\text{gm}} = \gamma_{\text{ga}} - \gamma_{\text{ma}} \cos \theta_c, \tag{2.2.2}$$

where θ_c is the contact angle of the metal wetted on graphene [37, 38]. Since the graphene cap is atomically thin, we have that in the wetted state $A_{\text{gm}} \approx A_{\text{ga}} = A_{\text{g}}$. Moreover, since we prescribe values for A_{g} we can simplify (2.2.1) to

$$E = A_{\text{g}} \left(\gamma_{\text{gm}} + \gamma_{\text{ga}} \right) + \gamma_{\text{ma}} A_{\text{ma}}, \tag{2.2.3}$$

and the minimum energy state is determined by the minimum value of A_{ma}. Because of this, we want to derive expressions which allow us to minimise A_{ma} in terms of the variables in Fig. 2.1.1.

The area, A, and volume, V, of a spherical cap are given in terms of its height,

h, and radius, r, by

$$A = 2\pi r h, \tag{2.2.4}$$

$$V = \frac{\pi h^2}{3}\left(3r - h\right). \tag{2.2.5}$$

Substituting for the graphene cap in (2.2.4) and rearranging yields

$$r_1 = \frac{A_{\mathrm{g}}}{2\pi h_1}, \tag{2.2.6}$$

which relates r_1 and h_1. Substituting for the graphene cap in (2.2.5) and eliminating r_1 using (2.2.6) gives

$$V_1 = \frac{\pi h_1}{3}\left(\frac{3A_{\mathrm{g}}}{2\pi} - h_1^2\right), \tag{2.2.7}$$

which relates V_1 and h_1. Additionally, we can substitute for the exposed metallic catalyst particle to obtain

$$V_2 = \frac{\pi h_2^2}{3}\left(3r_2 - h_2\right). \tag{2.2.8}$$

Substituting for (2.2.7) and (2.2.8) in (2.1.1) and rearranging gives

$$r_2 = \frac{V_{\mathrm{m}}}{\pi h_2^2} - \frac{1}{3h_2^2}\left(\frac{3A_{\mathrm{g}}}{2\pi}h_1 - h_1^3 - h_2^3\right), \tag{2.2.9}$$

which gives r_2 in terms of h_1 and h_2. Substituting for the exposed metallic catalyst particle in (2.2.4) and eliminating r_2 using (2.2.9) gives

$$A_{\mathrm{ma}} = \frac{1}{3h_2}\left(6V_{\mathrm{m}} - 3A_{\mathrm{g}}h_1 + 2\pi\left(h_1^3 + h_2^3\right)\right), \tag{2.2.10}$$

which gives the area of metal exposed to the atmosphere in terms of h_1 and h_2.

Now that we have an expression for A_{ma}, we may use differential calculus to determine the minimum energy configuration in terms of h_1 and h_2. However to ensure that the resulting system is physically relevant, we must constrain the possible values of h_1 and h_2. To do so we consider the two spherical caps which leads to two expressions for the interfacial radius a,

$$a = \sqrt{2r_1h_1 - h_1^2}, \tag{2.2.11}$$
$$a = \sqrt{2r_2h_2 - h_2^2}. \tag{2.2.12}$$

We now equate the right hand side of (2.2.11) and (2.2.12), and then eliminate r_1 and r_2 using (2.2.6) and (2.2.9) to obtain the constraint

$$f(h_1, h_2) = \pi \left(h_1^3 + h_2^3\right) + 3\left(A_\mathrm{g} - \pi h_1^2\right)(h_1 + h_2) - 6V_\mathrm{m} = 0. \tag{2.2.13}$$

We may minimise A_ma subject to $f = 0$ by instead considering the modified area

$$A^\star = A_\mathrm{ma} + \lambda f, \tag{2.2.14}$$

where λ is a Lagrange multiplier. Extrema of $A^\star$ occur where

$$\frac{\partial A^\star}{\partial h_1} = \frac{\partial A^\star}{\partial h_2} = \frac{\partial A^\star}{\partial \lambda} = 0, \tag{2.2.15}$$

and these correspond to extrema in A_ma subject to the constraint (2.2.13).

2.3 Solving for h_2 analytically in terms of h_1

Using the constraint (2.2.13) we can determine h_2 in terms of h_1, which reduces the problem by one dimension. The constraint (2.2.13) may be rearranged to

$$0 = h_2^3 + \frac{3}{\pi}\left(A_\mathrm{g} - \pi h_1^2\right)h_2 - 2h_1^3 + \frac{3}{\pi}A_\mathrm{g}h_1 - \frac{6V_\mathrm{m}}{\pi}, \tag{2.3.1}$$

which is a cubic in h_2 of the form $h_2^3 + ph_2 + q = 0$. Given $4p^3 + 27q^2 > 0$, there exists a single real solution for h_2 which is given by Cardano's formula. If we substitute for A_{g} in p using 2.2.4 we obtain

$$\begin{aligned} p &= \frac{3}{\pi}\left(2\pi r_1 h_1 - \pi h_1^2\right) \\ p &= 3h_1\left(2r_1 - h_1\right), \end{aligned}$$

and since $2r_1 \geq h_1$, we require that $4p^3 \geq 0$. We also substitute for A_{g} in q to obtain

$$\begin{aligned} q &= -2h_1^3 + 6r_1 h_1^2 - \frac{6V_{\mathrm{m}}}{\pi} \\ q &= 2h_1^2\left(3r_1 - h_1\right) - \frac{6V_{\mathrm{m}}}{\pi} \\ q &= \frac{6}{\pi}\left(V_1 - V_{\mathrm{m}}\right). \end{aligned}$$

Since q is always real, we have that $27q^2 \geq 0$, and therefore $4p^3 + 27q^2 \geq 0$. The only case in which Cardano's formula does not apply is the case where $h_1 = 2r_1$ and $V_1 = V_m$, that is, the case in which the graphene cap completely encapsulates the metallic catalyst particle. In this case (2.3.1) reduces to $h_2^3 = 0$ and the solution is trivially $h_2 = 0$. In all other cases we have only one real solution to (2.3.1) which is given by

$$h_2 = s - \frac{p}{3s}, \tag{2.3.2}$$

where

$$s = \left[\left(\frac{q^2}{4} + \frac{p^3}{27}\right)^{1/2} - \frac{q}{2}\right]^{1/3}.$$

2.4 Dewetted state

When dewetting occurs, the system changes such that the metallic catalyst particle no longer fills the graphene cap. This results in a new system comprising a spherical metallic catalyst particle with a spherical graphene cap resting on top of the particle

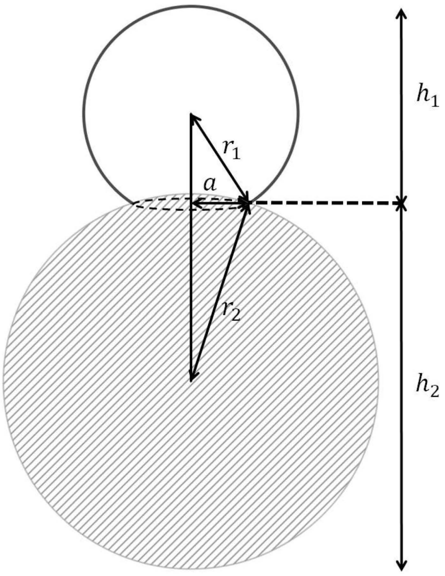

Figure 2.4.1: Geometry of dewetted cap-particle system

as illustrated in Fig. 2.4.1. In this state the surface free energy is given by

$$E = 2\gamma_{\mathrm{ga}} A_{\mathrm{g}} + \gamma_{\mathrm{ma}} A_{\mathrm{m}}, \tag{2.4.1}$$

where A_{m} is the total surface area of the metallic catalyst particle. Note that the energy of the dewetted state is constant with respect to r_1, r_2, h_1, and h_2. We expect that dewetting is favoured to occur when the energy of the dewetted state is less than the minimum energy of the wetted state as given by (2.2.1) while (2.2.15) is satisfied

2.5 Analytic solution

The initial model of the wetted state energy is sufficiently tractable that we may determine an extremal energy state analytically. The partial derivatives of $A^\star$ with respect to h_1, h_2, and λ are

$$\frac{\partial A^\star}{\partial h_1} = \frac{1}{h_2}\left(2\pi h_1^2 - A_\mathrm{g}\right) + \lambda\left[3A_\mathrm{g} - 6\pi h_1(h_1+h_2)\right], \tag{2.5.1}$$

$$\frac{\partial A^\star}{\partial h_2} = -\frac{1}{3h_2^2}\left[6V_\mathrm{m} - 3A_\mathrm{g}h_1 + 2\pi(h_1^3 - 2h_2^3)\right] + \lambda\left[3A_\mathrm{g} - 3\pi(h_1^2 - h_2^2)\right], \tag{2.5.2}$$

$$\frac{\partial A^\star}{\partial \lambda} = \pi\left(h_1^3 + h_2^3\right) + 3\left(A_\mathrm{g} - \pi h_1^2\right)(h_1+h_2) - 6V_\mathrm{m}, \tag{2.5.3}$$

and we require that all these partial derivatives equal zero for all possible values of our analytic solution. Now, suppose that $r_1 = r_2 = r$ and that $h_1 + h_2 = 2r$, that is, the system is shaped like a single sphere with a cap covering part of its surface. Given this assumption, the partial derivative (2.5.3) becomes

$$\frac{\partial A^\star}{\partial \lambda} = 2\pi r\left(h_1^2 - h_1h_2 + h_2^2\right) + 6r\left(A_\mathrm{g} - \pi h_1^2\right) - 6V_\mathrm{m}, \tag{2.5.4}$$

and we may substitute for A_g using (2.2.4). Since the system is spherically shaped we may also substitute for V_m using the formula for the volume of a sphere,

$$V_\mathrm{m} = \frac{4}{3}\pi r^3. \tag{2.5.5}$$

Thus the equation (2.5.4) becomes

$$\begin{aligned}
\frac{\partial A^\star}{\partial \lambda} &= 2\pi r\left(h_1^2 - h_1h_2 + h_2^2\right) + 6\pi r h_1\left(2r - h_1\right) - 8\pi r^3\\
\frac{\partial A^\star}{\partial \lambda} &= 2\pi r\left(h_1^2 - h_1h_2 + h_2^2 + 3h_1h_2 - (h_1+h_2)^2\right)\\
\frac{\partial A^\star}{\partial \lambda} &= 2\pi r\left((h_1+h_2)^2 - (h_1+h_2)^2\right)\\
\frac{\partial A^\star}{\partial \lambda} &= 0,
\end{aligned}$$

as required. This result demonstrates that our solution satisfies the constraint (2.2.13). Substituting the same way in (2.5.1) gives

$$\begin{aligned}
\frac{\partial A^\star}{\partial h_1} &= \frac{2\pi h_1}{h_2}(h_1 - r) - 6\lambda\pi r h_1, \\
\frac{\partial A^\star}{\partial h_1} &= \frac{\pi h_1(h_1 - h_2)}{h_2} - 6\lambda\pi r h_1.
\end{aligned}$$

We obtain λ by solving $\partial A^\star/\partial h_1 = 0$,

$$\lambda = \frac{h_1 - h_2}{6rh_2}. \tag{2.5.6}$$

Finally, substituting for our solution and (2.5.6) into (2.5.2) gives

$$\begin{aligned}
\frac{\partial A^\star}{\partial h_2} &= -\frac{\pi}{3h_2^2}\left(8r^3 - 6rh_1^2 + 2(h_1^3 - 2h_2^3)\right) + \frac{\pi(h_1 - h_2)}{2rh_2}\left(2rh_1 - (h_1^2 - h_2^2)\right), \\
\frac{\partial A^\star}{\partial h_2} &= -\frac{\pi}{3h_2^2}\left((h_1 + h_2)^3 - 3(h_1 + h_2)h_1^2 + 2(h_1^3 - 2h_2^3)\right) \\
&\quad + \frac{\pi(h_1 - h_2)}{2rh_2}\left(2rh_1 - 2r(h_1 - h_2)\right), \\
\frac{\partial A^\star}{\partial h_2} &= -\frac{\pi}{3h_2^2}\left(3h_1h_2^2 - 3h_2^3\right) + \pi(h_1 - h_2), \\
\frac{\partial A^\star}{\partial h_2} &= -\pi(h_1 - h_2) + \pi(h_1 - h_2), \\
\frac{\partial A^\star}{\partial h_2} &= 0.
\end{aligned}$$

Since each partial derivative (2.5.1)–(2.5.3) equates to 0 when $r_1 = r_2 = r$ and $h_1 + h_2 = 2r$, this solution provides an extremum of the energy for the wetted state.

2.6 Results

Using the base model for graphene cap growth developed in this chapter, we may study how changes in parameter values affect the shape and energy of the wetted and dewetted systems. Since we can determine r_1, r_2, h_1, and h_2 given only one of these values, we will classify the shape of the system according to the graphene cap

Parameter	Value	Citations
γ_{ga}	0.0646 eV/Å^2	[39]
γ_{gm}	0.137 eV/Å^2	[39, 40]
γ_{ma}	0.0918 eV/Å^2	[40]

Table 2.6.1: Parameter values

radius r_1. We can think of A_{g} increasing as representing the time evolution of the system since we expect the graphene cap to grow monotonically over time during nanotube nucleation. With this in mind, we vary A_{g} so that we may observe the effect of particle size on the system. We also vary V_{m} since the range of A_{g} values depends on the metallic catalyst particle size. Moreover since γ_{ga}, γ_{gm}, and γ_{ma} are controllable parameters, we will use values consistent with CNTs growing on an iron carbide catalyst particle at 800°C. We choose a carbide for the particle since we expect the particle to be saturated with carbon before cap growth occurs. Rhee reported that the surface free energy coefficient across the basal plane (across a graphene sheet) of pyrolytic graphite is given by

$$\gamma = 1174 - 0.13T \text{ erg/cm}^2 \tag{2.6.1}$$

where T is the temperature in degrees Kelvin [39]. We use this to estimate our value of γ_{ga} at 800°C to be 0.0646 eV/Å^2. For iron carbide, Nizhenko et al. reported a surface energy coefficient of 1470 erg/cm^2 and a contact angle on graphene of 142° [40]. Converting to the required units gives us our estimate for γ_{ma}, 0.0918 eV/Å^2. Using 2.2.2 we calculate our estimate for γ_{gm} to be 0.137 eV/Å^2. Our parameter values are summarised in Table 2.6.1.

Fig. 2.6.1 shows that the analytic solution for the radius of the graphene cap in the wetted system is accurate to the behaviour observed by our model. That is, the radius of the graphene cap is constant with respect to A_{g} and the shape of the

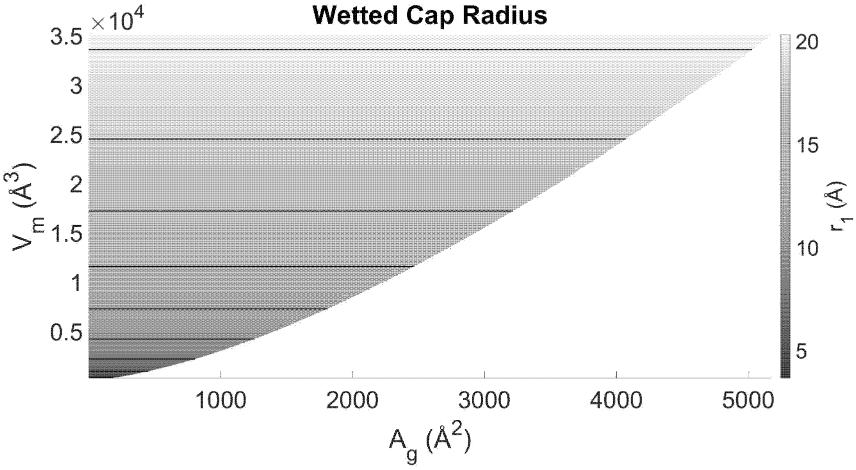

Figure 2.6.1: Graphene cap radius of the wetted state as V_m and A_g are varied

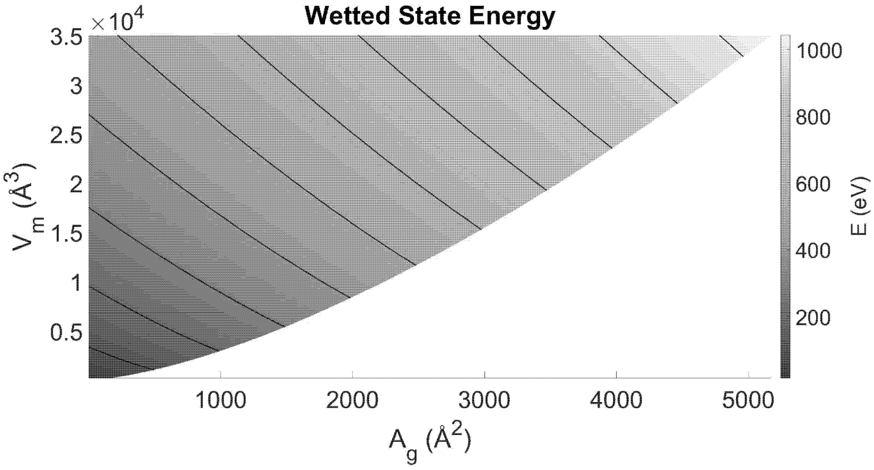

Figure 2.6.2: Energy of the wetted state as V_m and A_g are varied

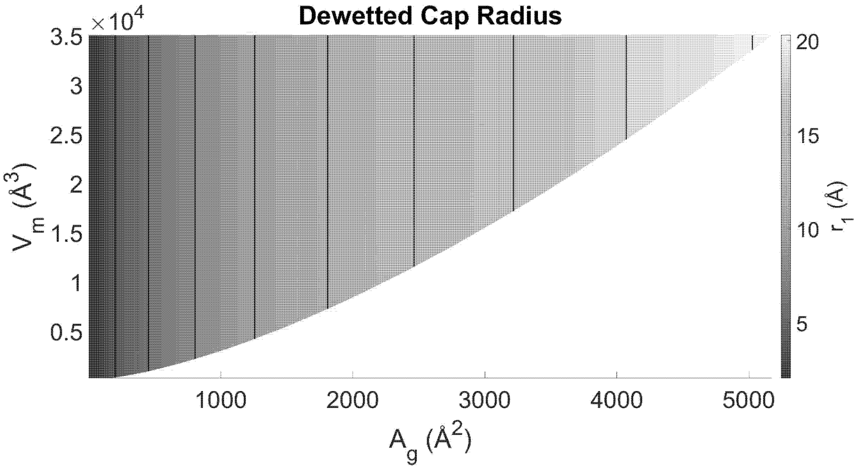

Figure 2.6.3: Graphene cap radius of the dewetted state as V_{m} and A_{g} are varied

system is defined by the system of equations

$$r_1 + r_2 = r, \quad h_1 + h_2 = 2r,$$

for some constant r. The value of r is determined by V_{m} as it is equal to the radius of a sphere with volume V_{m}. This dependence on V_{m} can also be observed in Fig. 2.6.1 as the radius increases with V_{m}, and the locations of the contour lines are constant with respect to A_{g}.

Fig. 2.6.2 illustrates how the energy of the wetted state changes as V_{m} and A_{g} are varied. As either of these variables are increased, the energy of the wetted state increases. Although increasing the area of the graphene cap decreases the exposed area of the metallic catalyst particle, this still results in an overall increase in energy since the energy cost per unit area of the graphene cap is greater than that of the metallic catalyst particle.

The radius of the graphene cap for the dewetted state is shown in Fig. 2.6.3. Since the graphene cap radius does not affect the energy of the dewetted state in this model, its value is always as low as possible. This issue results in a dependence on A_{g} but not V_{m}, as indicated by the vertical contour lines.

The effect of V_{m} and A_{g} on the energy of the dewetted state is seen in Fig. 2.6.4.

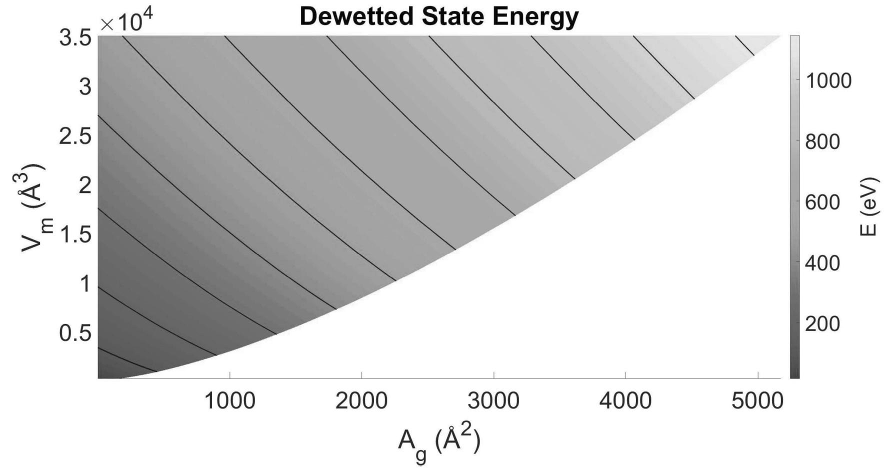

Figure 2.6.4: Energy of the dewetted state as V_{m} and A_{g} are varied

As with the wetted state, the energy of the system increases as either A_{g} or V_{m} is increased. However, increasing A_{g} does not reduce the energy contribution of the metal-atmosphere interface for the dewetted system so that the energy of the dewetted state is always greater than that of the wetted state. This is due to the area of the metal-atmosphere interface being precisely the total surface area of the metallic catalyst particle. However, if γ_{gm} is greater than γ_{ga}, then it is still possible for the energy of the wetted state to exceed that of the dewetted state.

2.7 Conclusion

Due to the exclusion of important physical effects, such as dangling chemical bonds, this initial model for graphene cap growth predicts a constant shape for the wetted system regardless of the given parameter values. The minimum energy configuration of the wetted state is always described by the solution outlined in §2.4 which results in a system with a constant shape regardless of A_{g} and V_{m}. Moreover, this model does not take into account the energy contribution from other factors such as carbon-metal bonds at the edge of the graphene cap. In the next chapter we account for these factors by adding terms to the energy formulae of the wetted and dewetted states which describe the energy contribution due to these factors. To expand on our model we first account for carbon-metal bonds at the edge of the cap by adding a line tension term. This term imposes an energy cost to the dewetted and wetted states based on the circumference of the graphene cap's edge, as well as its contact angle. One would expect that this term should make graphene caps with lower radii relatively more energetically favourable.

Chapter 3

Accounting for line tension

We now expand on the graphene cap growth model by adding a line tension energy term. This introduces an energy cost that depends on the length of the circular edge where the graphene cap meets the metallic catalyst particle, and the contact angle between them. Adding a line tension term means that the model will prefer graphene caps with smaller radii since they will be relatively more energetically favourable compared with the surface free energy model of Chapter 2. This is because the graphene cap adopts a complete spherical shell when its radius is minimised and the length of its edge becomes zero.

3.1 Line tension energy

Where the edge of the graphene cap meets the metallic catalyst particle, carbon atoms in the graphene cap are bound only to two neighbouring carbon atoms. For each of these carbon atoms we impose an energy cost on the system equivalent to the carbon-carbon bond dissociation energy in the fullerene to account for missing bonds along the graphene cap edge. Since the circumference of the graphene cap is given by $2\pi a$, we calculate this cost using

$$E_e = 2\pi a \gamma_{\ell t}, \tag{3.1.1}$$

where $\gamma_{\ell t}$ is our energy cost per unit length. Along the edge of the graphene cap, carbon atoms interact with the surface of the metallic catalyst particle. By forming bonds with the metal and absorbed carbon atoms in the surface of the catalyst particle the cost of the unbonded carbon atoms on the graphene cap edge is offset. However, if the graphene cap does not meet the catalyst particle at a right angle then this offset is reduced due to strain resulting from a suboptimal bond angle. Therefore we define the energy offset to be

$$E_o = 2\pi a \gamma_{\ell t}^{\star} \sin\theta_b, \tag{3.1.2}$$

where θ_b is the contact angle, and $\gamma_{\ell t}^{\star}$ is the mean energy per unit length for bonds between carbon atoms in the graphene cap and atoms on the metallic catalyst particle's surface. If the surface of the metallic catalyst particle consisted only of carbon atoms, we would have $\gamma_{\ell t}^{\star} = \gamma_{\ell t}$ in order to cancel the energy cost. If the surface of the metallic catalyst particle was purely iron atoms, for example, then $\gamma_{\ell t}^{\star}$ would be equal to the bond dissociation energy per unit length for iron-carbon bonds along the graphene cap edge. Subtracting (3.1.2) from (3.1.1) gives our line tension energy,

$$E_{\ell t} = 2\pi a \left(\gamma_{\ell t} - \gamma_{\ell t}^{\star} \sin\theta_b\right), \tag{3.1.3}$$

The contact angle is equivalent to the angle between r_1 and r_2 as shown in Fig. 3.1.1, therefore we have that

$$\theta_b = \cos^{-1}\left(\frac{a}{r_1}\right) + \cos^{-1}\left(\frac{a}{r_2}\right). \tag{3.1.4}$$

Substituting (3.1.4) into $\sin(\theta_b)$ and simplifying yields

$$\sin(\theta_b) = \frac{a}{r_1 r_2}\left(r_1 - h_1 + r_2 - h_2\right). \tag{3.1.5}$$

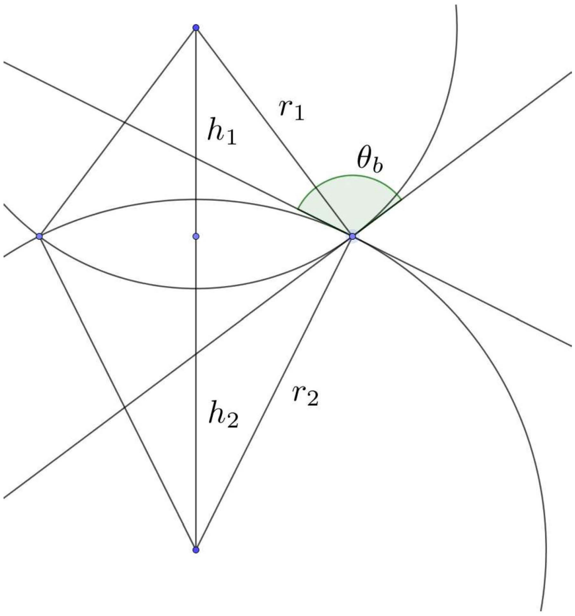

Figure 3.1.1: Contact angle, θ_b, between the graphene cap and exposed metallic catalyst particle

We can also derive an expression for $\cos(\theta_b)$ in the same way which produces

$$\cos(\theta_b) = \frac{a^2}{r_1 r_2} - \frac{1}{r_1 r_2}\left(r_1 - h_1\right)\left(r_2 - h_2\right). \tag{3.1.6}$$

We can obtain (3.1.5) and (3.1.6) in terms of h_1 and h_2 only by substituting either (2.2.11) or (2.2.12) for a, and then substituting for r_1 and r_2 using (2.2.6) and (2.2.9), respectively.

3.2 Dewetted state

The line tension energy of the dewetted state is calculated using the same formula as the wetted state, so the total energy of the dewetted state is simply the sum of (2.4.1) and (3.1.3). However, with the addition of the line tension term the energy of the dewetted state also depends on the total conformation of the system. Because of this we now require expressions for r_1, r_2, h_1, and h_2 so we can find the minimum energy of the dewetted state. Given h_1, we again have r_1 in terms of h_1 from (2.2.6). Since the metallic catalyst particle is a sphere in the dewetted state, the value of r_2

is constant and is given by

$$r_2 = \left(\frac{3V_m}{4\pi}\right)^{1/3}. \tag{3.2.1}$$

Using (2.2.11) and (2.2.12), we obtain a quadratic expression in h_1 and h_2,

$$h_2^2 - 2r_2h_2 - h_1^2 + 2r_1h_1 = 0. \tag{3.2.2}$$

We can then use the quadratic formula to obtain the following expressions for h_2,

$$h_2 = r_2 + \sqrt{r_2^2 - 2r_1h_1 + h_1^2}, \tag{3.2.3}$$

$$h_2 = r_2 - \sqrt{r_2^2 - 2r_1h_1 + h_1^2}. \tag{3.2.4}$$

For $A_{\mathrm{g}} \leq A_{\mathrm{m}}/2$, however, the solution (3.2.4) is not physically relevant and we always use (3.2.3) in this domain. Since we are able to determine r_1, r_2, and h_2 given only h_1, we do not need to incorporate any constraints into the minimisation problem for the dewetted state.

3.3 Results

We can now use the model developed in this chapter to study how changes in parameter values affect the system when line tension is incorporated. We will again consider an iron carbide catalyst particle at 800°C, so we use the same parameter values as those used in §2.6. However, we also need values for $\gamma_{\ell t}$ and $\gamma_{\ell t}^{\star}$. For the carbon-carbon bond dissociation energy within a fullerene we use 4.86 eV [41], and we can estimate $\gamma_{\ell t}$ by multiplying this by an estimate for the length per atom across the graphene cap edge. We obtain this estimate by considering a C_{60} fullerene cut in half. Where the fullerene is cut in half, there is a circular edge of radius 3.55 Å with 9 carbon atoms. This gives us our estimate of 2.48 Å/atom along a graphene cap edge and thus we have $\gamma_{\ell t} = 1.96$ eV/Å. For $\gamma_{\ell t}^{\star}$, we use a weighted average

Parameter	Value	Citations
γ_{ga}	0.0646 eV/Å^2	[39]
γ_{gm}	0.137 eV/Å^2	[39, 40]
γ_{ma}	0.0918 eV/Å^2	[40]
$\gamma_{\ell t}$	1.96 eV/Å	[41]
$\gamma_{\ell t}^{*}$	0.80 eV/Å	[42, 43]

Table 3.3.1: Parameter values

between the iron-carbon and carbon-carbon bond dissociation energies. Riordan and Halpern found the bond dissociation energy between iron and a hexagonal carbon ring to be 31 kcal/mol, or 1.3 eV/bond [42]. The energy per unit length for iron-carbon bonds is therefore 0.52 eV/Å. The bond dissociation energy between carbon and a hexagonal carbon ring was reported by Cottrell to be 389 kJ/mol, or 4.03 eV/bond [43]. Therefore the energy per unit length for carbon-carbon bonds is 1.63 eV/Å. Since iron carbide consists of three iron atoms for each carbon atom, our weighted average for $\gamma_{\ell t}^{\star}$ is

$$\gamma_{\ell t}^{\star} = \frac{3(0.52) + 1.63}{4} \approx 0.80 \text{ eV/Å} \quad \text{(2 decimal places).}$$

Table 3.3.1 summarises our parameter values so far. Again, we consider the graphene cap area A_{g} to be a time-like variable so we will consider how V_{m} affects the shape and energy of the system as A_{g} is increased.

Fig. 3.3.1 illustrates the effect of A_{g} and V_{m} on the wetted cap radius. With the addition of the line tension term, the model now favours lower radii for smaller graphene caps instead of maintaining a constant radius for each value of A_{g}. In the present model, the radius is dependent only on A_{g} in the region being observed. This is due to the edge of the graphene cap disappearing as the radius approaches its minimum, which results in the line tension term minimising at the smallest possible radius given A_{g}. It can also be seen that for very low values of A_{g} the radius is constant. This is due to the minimum radius being constrained to the radius of the C_{20} fullerene.

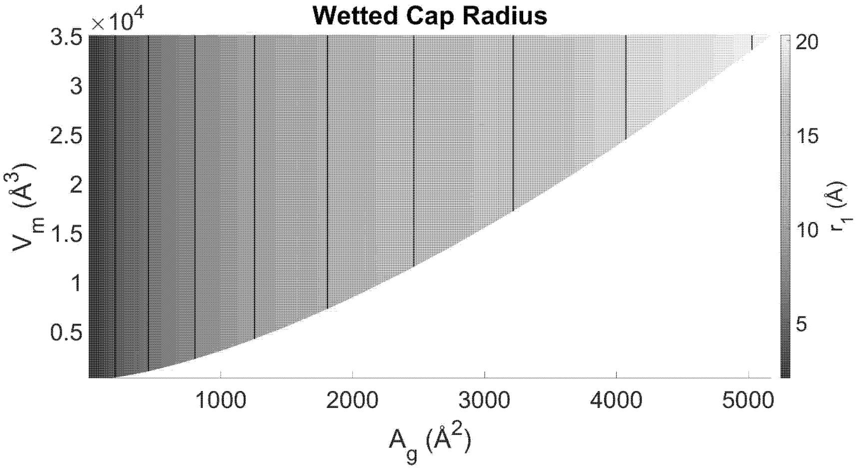

Figure 3.3.1: Graphene cap radius of the wetted state as particle volume and graphene area are varied

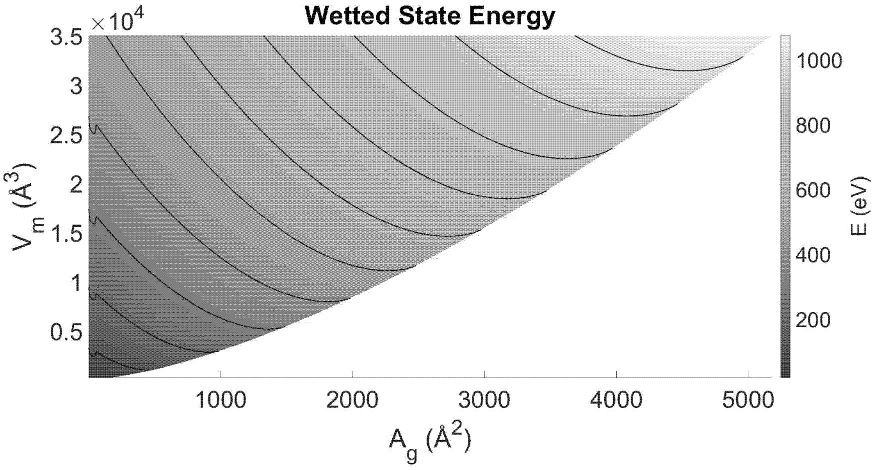

Figure 3.3.2: Energy of the wetted state as particle volume and graphene area are varied

The energy distribution seen in Fig. 3.3.2 is similar to that seen in Fig. 2.6.2 for the previous model. However, the contour lines now curve upwards as A_{g} approaches its maximum value. This indicates the existence of a local energy maximum for each value of V_{m}. Additionally the contour lines show a change in behaviour for the energy at low values of A_{g}. This is due to the minimum radius constraint which forces the system to occupy a minima with greater energy than that for lower radii in this region. Overall, the addition of the line tension term had a noticeable effect on the energy of the minimum energy state.

For the dewetted state, Fig. 3.3.3 shows the radius of the graphene cap as

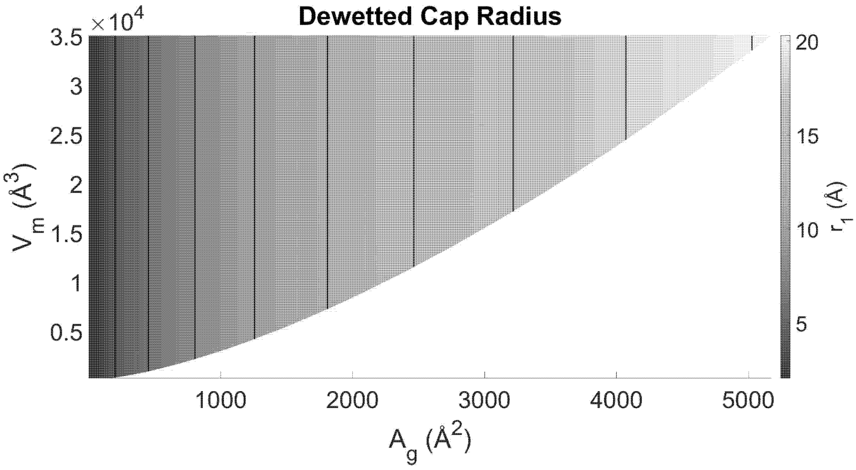

Figure 3.3.3: Graphene cap radius of the dewetted state as particle volume and graphene area are varied

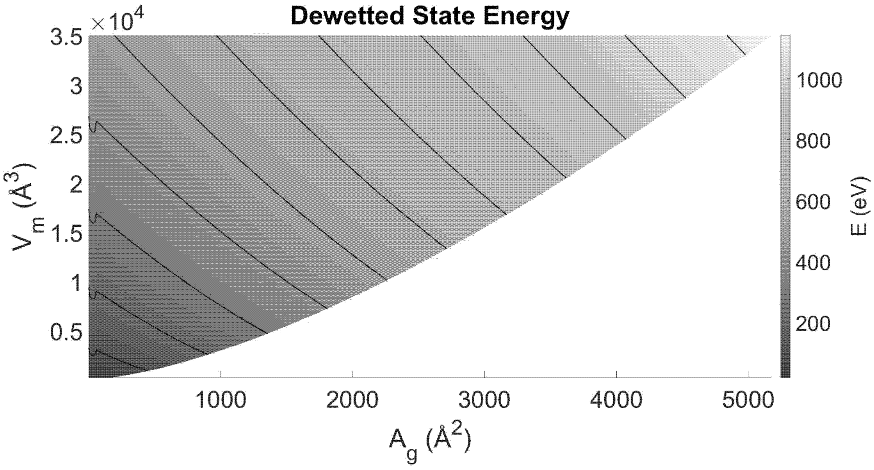

Figure 3.3.4: Energy of the dewetted state as particle volume and graphene area are varied

graphene area and particle volume are varied. Similar to Fig. 3.3.1, the system favours the smallest possible graphene cap radius. This solution is reinforced by the addition of the line tension term since this term vanishes when the graphene cap radius is minimised provided that the graphene cap area is greater than the area of the C_{20} fullerene. This is because minimising the graphene cap radius results in a fully spherical graphene cap with $a = 0$. For any other value of the graphene cap radius the line tension energy is positive. As with the model from Chapter 2, the volume of the metallic catalyst particle has no effect on the graphene cap radius since the energy is always minimised when the graphene cap radius is minimised.

Again, the energy distribution seen in Fig. 3.3.4 is similar to those of the Chapter 2 model. However, we also see that the change in behaviour at low values of A_{g} seen in Fig. 3.3.2 is present here. As with the previous model, the energy of the dewetted state is greater than that of the wetted state for our given parameter values. Unlike the wetted state, the addition of the line tension term did not have a significant effect on the energy of the minimum energy state.

3.4 Conclusion

By adding the line tension term to the wetted and dewetted energy formulae, we have imposed a cost on each system that depends on the circumference of the graphene cap edge. As expected, this has resulted in optimal configurations with smaller graphene cap radii since the line tension term favours those graphene caps with smaller edges. However, due to the added complexity, we are no longer able to derive an analytic solution for the minimum energy configuration. Additionally this model still does not account for many factors which may affect the energy of the system. To expand on this model in the next chapter, we consider a new term that imposes an energy cost on more curved graphene caps by considering the type and number of carbon-carbon bonds we would expect to see in the graphene cap. By doing so we strive to produce a more accurate model of graphene cap growth and can more directly incorporate chirality into the energy minimisation problems.

Chapter 4

Graphene cap geometry

Our goal is to produce a model that can predict the structure of any resulting CNTs once dewetting has occured. To advance this aim we need to account for the discrete geometry of the graphene cap in our model. In this chapter we assume that the graphene cap comprises a section of a C_n fullerene for some n, and consider the energy contribution of carbon-carbon bonds within the graphene cap. We categorise the carbon-carbon bonds by the number of sides of the carbon rings they belong to, and therefore there are three types of carbon-carbon bonds. As illustrated by Fig. 4.0.1, these are

- 5-5 bonds: shared by two pentagonal carbon rings,
- 5-6 bonds: shared by one pentagonal and one hexagonal carbon ring, and
- 6-6 bonds: shared by two hexagonal carbon rings,

where each bond type contributes a different amount of energy to the system. For C_n fullerenes with $n \geq 60$ the isolated pentagon rule applies, ensuring that all pentagonal carbon rings are surrounded by hexagonal carbon rings. This is because fullerene structures with isolated pentagonal rings are more stable than other isomers, and such isomers exist only for $n \geq 60$ [44]. That is, only 5-6 and 6-6 carbon-carbon bonds exist in fullerenes C_n with $n \geq 60$. Because there are always exactly 12

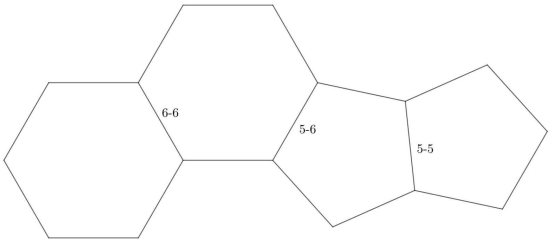

Figure 4.0.1: Bond types in fullerenes

pentagons in the fullerenes considered here, the number of 5-6 bonds will always be equal to 60 in fullerenes with $n \geq 60$. For fullerenes with $n < 60$ however, there are no guarantees as to the number of 5-5, 5-6, or 6-6 bonds. To overcome this we will use two different methods for calculating the energy contribution of the graphene cap.

4.1 C_n for $n \geq 60$

For the case where $n \geq 60$ we may determine the number of 6-6 bonds by subtracting the number of 5-6 bonds, which is known to be 60, from the total number of carbon-carbon bonds in the fullerene. Since each carbon atom is bonded to three neighbouring carbon atoms, and each bond is shared between two carbon atoms, we have that there are $3n/2$ bonds in a C_n fullerene. The total number of 6-6 bonds is therefore

$$n_{66} = \frac{3n}{2} - 60. \tag{4.1.1}$$

We can approximate the total number of bonds in the graphene cap using

$$b = \frac{3nA_{\mathrm{g}}}{2A_n}, \tag{4.1.2}$$

where A_n is the surface area of a C_n fullerene. We can also approximate the value of A_n by assuming that each bond in the fullerene contributes two isosceles triangles to the total surface area with base length equal to the bond length. Thus a 6-6 bond contributes two equilateral triangles, and a 5-5 bond contributes two isosceles triangles with a peak angle of $2\pi/5$ radians. Similarly, a 5-6 bond contributes both an equilateral and an isosceles triangle to the total surface area. The area of an equilateral triangle with side length ℓ is given by

$$A = \frac{\sqrt{3}}{4}\ell^2. \tag{4.1.3}$$

The area of an isosceles triangle with base length ℓ, equal sides of length s, and peak angle $\theta = 2\pi/5$ is given by

$$A = \frac{1}{2}s^2 \sin\left(\frac{2\pi}{5}\right). \tag{4.1.4}$$

Using the cosine rule, we have that

$$s^2 = \frac{\ell^2}{2[1-\cos(2\pi/5)]}. \tag{4.1.5}$$

Substituting (4.1.5) into (4.1.4) gives the area in terms of ℓ,

$$A = \frac{\ell^2 \sin(2\pi/5)}{4[1-\cos(2\pi/5)]}. \tag{4.1.6}$$

Using (4.1.3) and (4.1.6), we obtain the total surface area contribution for each bond type where n is the number of bonds and ℓ is the bond length.

$$A_{55} = n_{55}\frac{\ell_{55}^2 \sin(2\pi/5)}{2[1-\cos(2\pi/5)]} \tag{4.1.7}$$

$$A_{56} = n_{56}\left(\frac{\ell_{56}^2 \sin(2\pi/5)}{4[1-\cos(2\pi/5)]} + \frac{\sqrt{3}}{4}\ell_{56}^2\right) \tag{4.1.8}$$

$$A_{66} = n_{66}\frac{\sqrt{3}}{2}\ell_{66}^2 \tag{4.1.9}$$

The total surface area of a C_n fullerene is therefore

$$A_n \approx A_{55} + A_{56} + A_{66}. \tag{4.1.10}$$

We can simplify this expression by assuming that each bond has the same contribution to surface area as a 6-6 bond. for large n, the proportion of 6-6 bonds within the fullerene C_n approaches 1 and the relative error of our simplified expression becomes vanishingly small. The simplified approximation for A_n reduces to

$$A_n \approx \frac{3\sqrt{3}n}{4}\ell_{66}^2. \tag{4.1.11}$$

Now that we have an expression for A_n, we can derive expressions for the number of each bond type we would expect to see in a graphene cap with area A_{g} shaped like a C_n fullerene. If we assume that the number of each bond type in the graphene cap is distributed hypergeometrically, then the mean number of 5-6 and 6-6 bonds in the graphene cap are given by

$$b_{56} = 60\frac{A_{\mathrm{g}}}{A_n}, \tag{4.1.12}$$

$$b_{66} = \frac{A_{\mathrm{g}}}{A_n}\left(\frac{3n}{2} - 60\right). \tag{4.1.13}$$

Since for $n \geq 60$ there are no 5-5 bonds, we also have $b_{55} = 0$.

4.2 C_n for $20 \leq n \leq 60$

As with the previous section, we approximate the number of bonds in the graphene cap using (4.1.2). However, for all such fullerenes, the isolated pentagon rule does not apply. Thus to determine the number of each bond type in a C_n fullerene we will instead use a piecewise linear approximation for the number of bonds in the least strain energy isomers for various fullerenes. The number of each bond type for these isomers are shown in Fig. 4.2.1. For $20 \leq n \leq 28$, the number of each bond type is described exactly by

$$n_{55} = 60 - \frac{3n}{2}, \tag{4.2.1}$$

$$n_{56} = 3n - 60, \tag{4.2.2}$$

$$n_{66} = 0. \tag{4.2.3}$$

For the region $28 \leq n \leq 60$ we linearly interpolate between the values at the end points to obtain

$$n_{55} = \frac{135}{4} - \frac{9n}{16}, \tag{4.2.4}$$

$$n_{56} = \frac{9n}{8} - \frac{15}{2}, \tag{4.2.5}$$

$$n_{66} = \frac{15n}{16} - \frac{105}{4}. \tag{4.2.6}$$

Fig. 4.2.1 compares the piecewise linear approximation with the data in Fig. 4.2.1.

Using the formulae (4.2.1)–(4.2.3) or (4.2.4)–(4.2.6) as appropriate, we obtain the expected number of each bond type in the graphene cap as before,

$$b_{55} = \frac{A_{\mathrm{g}}}{A_n} n_{55}, \tag{4.2.7}$$

$$b_{56} = \frac{A_{\mathrm{g}}}{A_n} n_{56}, \tag{4.2.8}$$

$$b_{66} = \frac{A_{\mathrm{g}}}{A_n} n_{66}. \tag{4.2.9}$$

4.3 Fullerene structure determination

In order to obtain the data shown in Fig. 4.2.1, along with other useful information about fullerene structures, we require an automated method for obtaining data about bonds in fullerenes. Firstly, we formulate each fullerene as a graph in which each node represents a carbon atom in the fullerene, and each edge represents a bond

Fullerene	5-5 Bonds	5-6 Bonds	6-6 Bonds
C_{20}	30	0	0
C_{24}	24	12	0
C_{26}	21	18	0
C_{28}	18	24	0
C_{30}	17	26	2
C_{32}	15	30	3
C_{34}	14	32	5
C_{36}	12	36	6
C_{38}	11	38	8
C_{40}	10	40	10
C_{42}	9	42	12
C_{44}	8	44	14
C_{46}	8	44	17
C_{48}	7	46	19
C_{50}	5	50	20
C_{52}	6	48	24
C_{60}	0	60	30

Table 4.2.1: Bond types in stable isomers

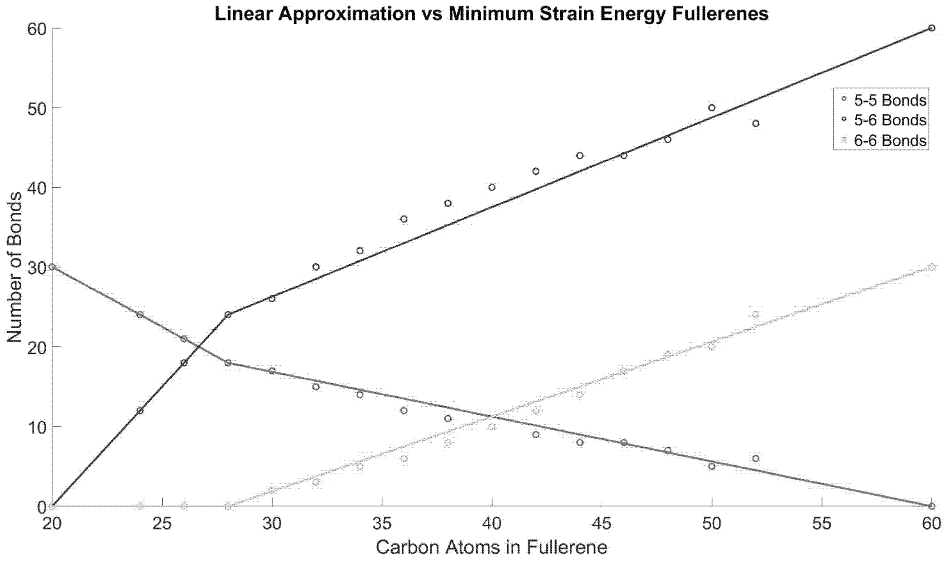

Figure 4.2.1: Piecewise linear approximation of bonds in fullerenes

between carbon atoms. We can then produce a line graph for each fullerene in which nodes represent the bonds, and nodes are connected by edges if the corresponding bonds share a carbon atom. Using the line graph we may determine bond type by performing a depth-first search on each node to find cycles with five unique members. Nodes belonging to zero five member cycles represent 6-6 bonds, while those belonging to only one five member cycle represent 5-6 bonds. The remaining nodes represent 5-5 bonds. This method for classifying bond types from fullerene structures has been automated using the MATLAB code in Appendix B, and can also be used to gather data about bond lengths. The bond length data obtained using this code can be employed to produce more accurate values of A_n.

The data we are studying was published by Frederick and Tomanek [45] and comprises 2,486 fullerene geometries with sizes ranging from $n = 20$ to $n = 720$. Of these fullerene geometries, 1,249 have $20 \leq n < 60$ while the remaining 1,237 have $n \geq 60$. This structured data was computed by reoptimising structures found in M. Yoshida's fullerene library [46] using a Dreiding-like force field. We use the code in Appendix B to analyse these fullerene geometries and obtain bond length data for 5-5, 5-6, and 6-6 bonds.

Figs. 4.3.1, 4.3.2, and 4.3.3 show the distributions of bond lengths for various isomers of C_n fullerenes with $20 \leq n < 60$. Fig. 4.3.1 shows that the distribution of 5-5 bond lengths is trimodal with peaks at approximately 1.33 Å, 1.36 Å, and 1.38 Å. The central peak is the largest and is close to the mean, which is approximately 1.359 Å. It can be seen from Fig. 4.3.2 that the distribution of 5-6 bond lengths is also trimodal with peaks at approximately 1.37 Å, 1.4 Å, and 1.42 Å. However, the peaks in this case are less varied in terms of frequency. The mean 5-6 bond length is greater than the mean 5-5 bond length at approximately 1.384 Å. The distribution of 6-6 bond lengths seen in Fig. 4.3.3 does not appear to be trimodal with only two distinct peaks at approximately 1.41 Å and 1.43 Å. The mean bond lengths is greater than that for the 5-5 and 5-6 bond length distributions at approximately

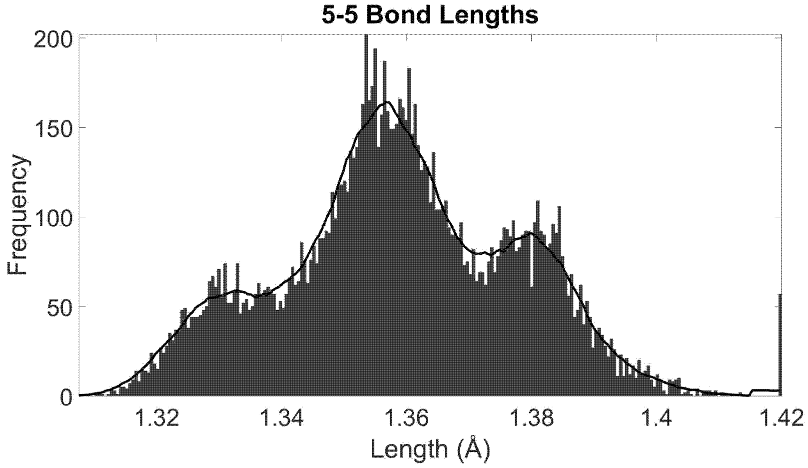

Figure 4.3.1: Distribution of 5-5 bond lengths seen in various isomers of C_n fullerenes with $20 \leq n < 60$

1.411 Å.

Figs. 4.3.4 and 4.3.5 show the distributions of bond lengths for various isomers of C_n fullerenes with $n \geq 60$. The distribution of 5-6 bond lengths seen in Fig. 4.3.4 appears to be approximately normal with a mean of 1.372 Å. However, the distribution of 6-6 bond lengths seen in Fig. 4.3.5 has many small peaks. The mean bond length for this distribution is approximately 1.4 Å, which is again greater than that for the 5-6 distribution.

Using these mean bond lengths and the linear approximation in Fig. 4.2.1, we are able to estimate the surface area of fullerenes for any given value of n using (4.1.7)–(4.1.10). This allows us to determine the expected number of each bond type in the graphene cap given only A_g and n.

We can further classify each bond type according to the number of neighbouring hexagons in the fullerene structure. We define neighbouring hexagons to be those hexagonal carbon rings that share a carbon atom with the bond of interest, but do not contain that bond. This is illustrated in Fig. 4.3.6. Figs. 4.3.7, 4.3.8, and 4.3.9 show the bond length data for isomers of C_n fullerenes with $20 \leq n < 60$, separated according to the number of neighbouring hexagons. The data in Fig. 4.3.7 shows that 5-5 bonds with two neighbouring hexagons tend to be distributed around the central

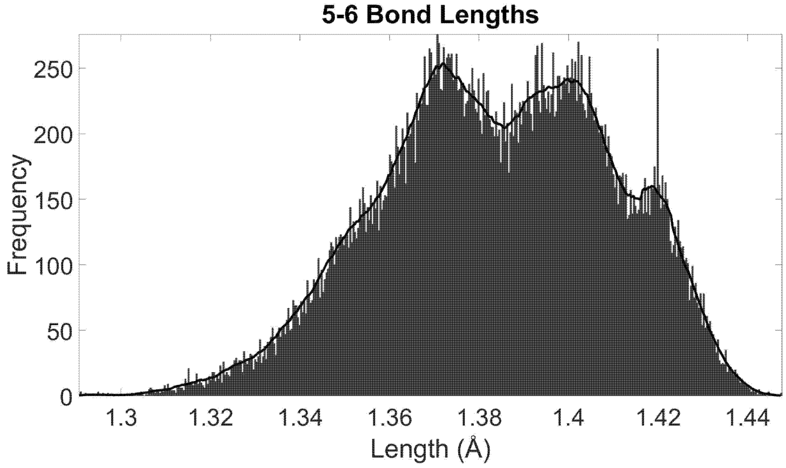

Figure 4.3.2: Distribution of 5-6 bond lengths seen in various isomers of C_n fullerenes with $20 \leq n < 60$

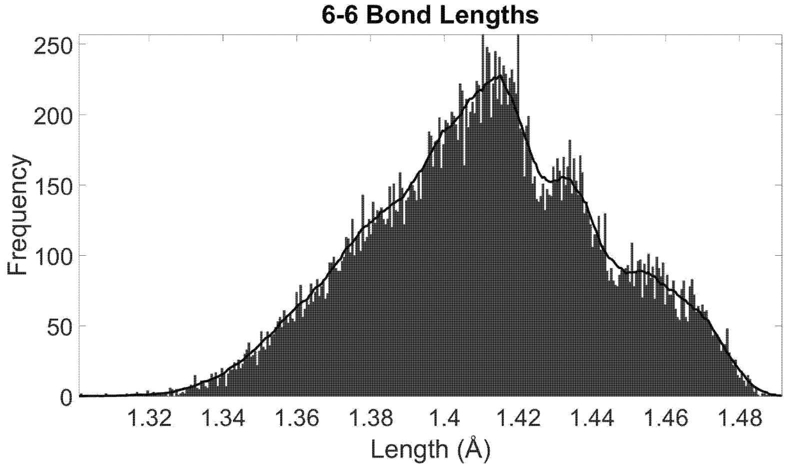

Figure 4.3.3: Distribution of 6-6 bond lengths seen in various isomers of C_n fullerenes with $20 \leq n < 60$

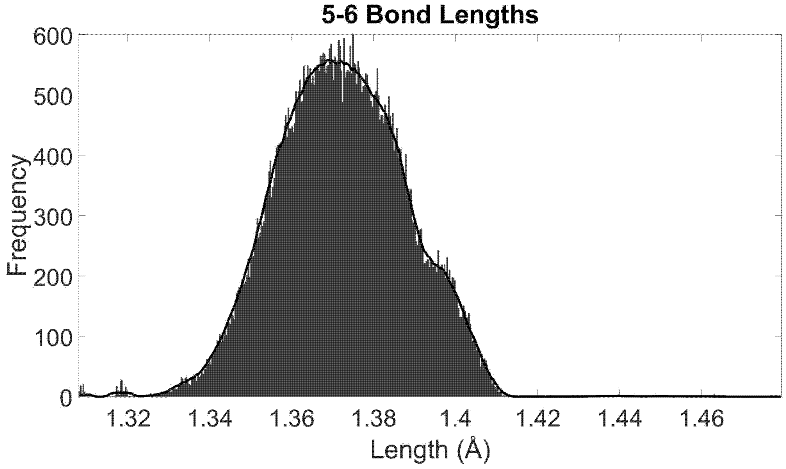

Figure 4.3.4: Distribution of 5-6 bond lengths seen in various isomers of C_n fullerenes with $n \geq 60$

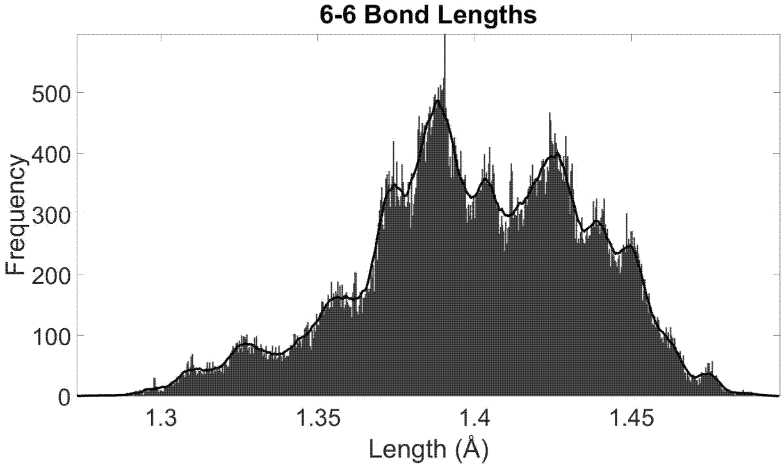

Figure 4.3.5: Distribution of 6-6 bond lengths seen in various isomers of C_n fullerenes with $n \geq 60$

and rightmost peaks of the total distribution. However, the 5-5 bonds with only one neighbouring hexagon tend to be shorter and have lengths distributed across all peaks of the total distribution. The 5-5 bonds with zero neighbouring hexagons are significantly rarer than other kinds of 5-5 bonds, as indicated by the low frequencies seen in its distribution compared to the total distribution. As shown in Fig. 4.3.8, the length data for different numbers of neighbouring hexagons separates into distinct distributions. Each of these distributions appear approximately normal, except for the 5-6 bonds with two nearby hexagons which appear to have the bond length distributed bimodally. Again, the bonds with two neighbouring hexagons tend to have longer lengths than those with only one neighbouring hexagon. Additionally, the bonds with zero neighbouring hexagons tend to be the shortest of the three categories. The 6-6 bond length data shown in Fig. 4.3.9 is similar to that in Fig. 4.3.8, however, each distribution now appears to be approximately normal and the frequency of bonds with zero neighbouring hexagons is similar to that of bonds with two neighbouring hexagons. This bond length data could be used to further improve approximations for A_n, however this has not been attempted here because that would significantly increase the amount of computation required to calculate A_n.

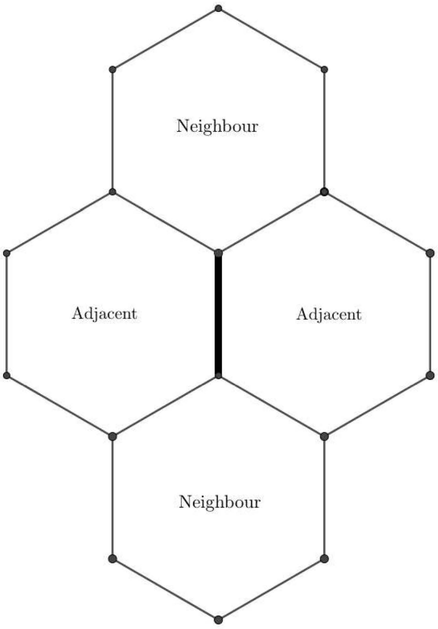

Figure 4.3.6: Diagram of neighbouring carbon rings

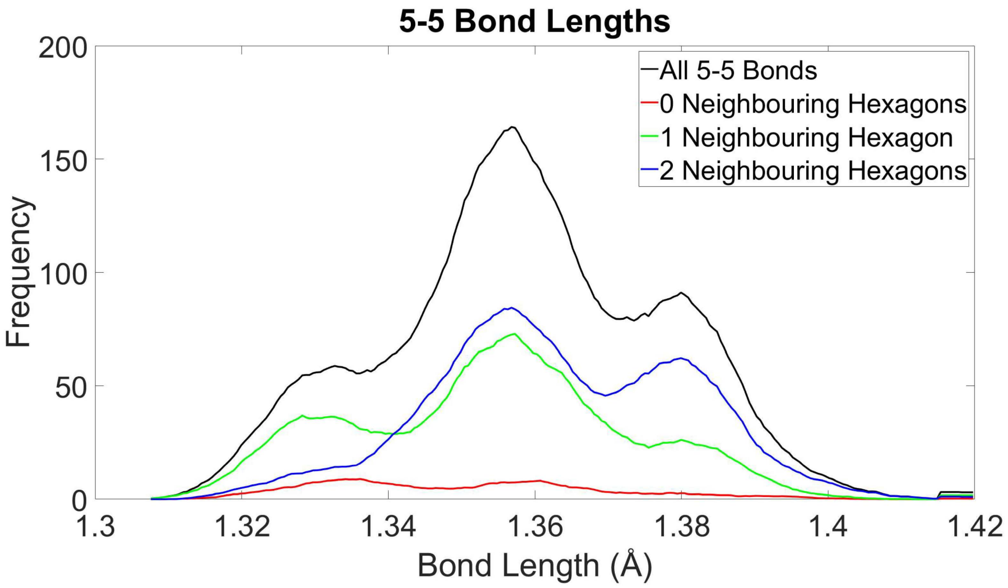

Figure 4.3.7: Smoothed plot of 5-5 bond lengths

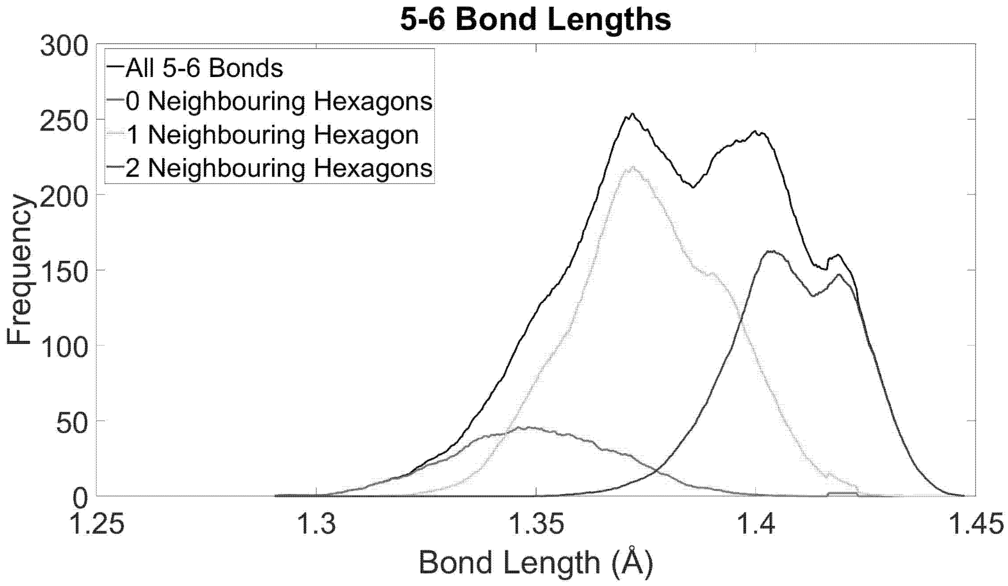

Figure 4.3.8: Smoothed plot of 5-6 bond lengths

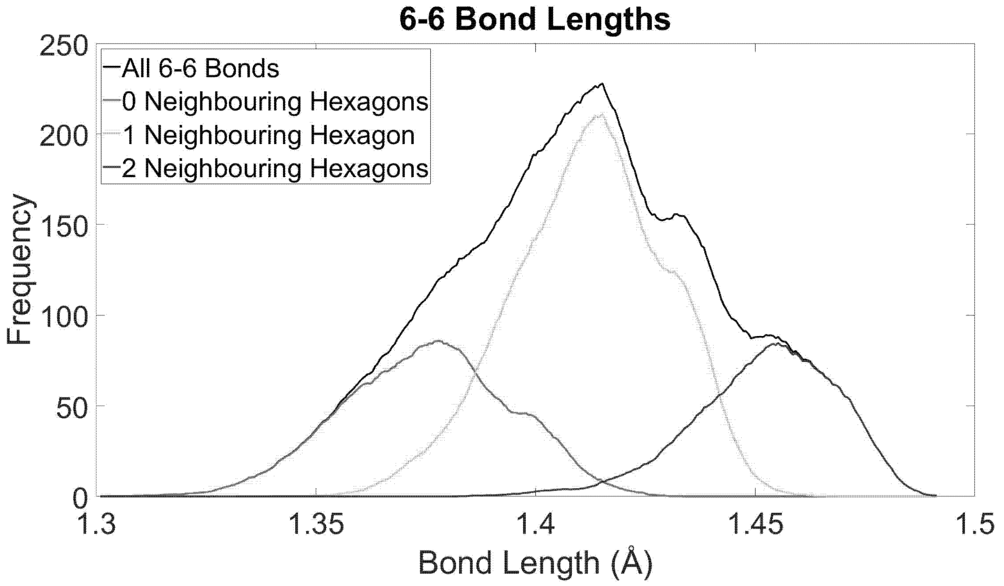

Figure 4.3.9: Smoothed plot of 6-6 bond lengths

4.4 Energy contribution

Let the bond energy of 5-5, 5-6, and 6-6 bonds be E_{55}, E_{56}, and E_{66} respectively. The total energy contribution of the graphene cap to the system is then

$$E_c = -E_{55}n_{55} - E_{56}n_{56} - E_{66}n_{66}. \tag{4.4.1}$$

The energy contribution is negative as it represents how strongly bound the carbon atoms are in the graphene cap. The graphene cap energy term (4.4.1) applies for both the wetted and dewetted states. This is because the graphene cap energy term is determined by the graphene cap height and radius.

4.5 Results

With the addition of the graphene cap geometry term, we are now able to study how changes in graphene area, A_g, and particle volume, V_m, affect the system while taking into account the surface free energy, the line tension energy, and the geometry of the graphene cap. For the analysis here we will use the same parameter values as those used in §3.3 as well as the dissociation energies of each bond type, $E_{55} = 2.62$ eV, $E_{56} = 4.54$ eV, and $E_{66} = 4.86$ eV [41]. The values for 6-6 and 5-6 bonds were derived from the average bond dissociation energy of the C_{60} fullerene, and the value for 5-5 bonds was derived from the average bond dissociation energy of the C_{58} fullerene. All of our parameter values are summarised in Table 4.5.1

Fig. 4.5.1 shows the effect of A_g and V_m on the radius of the graphene cap. As was seen in the previous chapter, the radius of the cap remains constant at the lowest values of A_g. The point at which the radius begins to increase, however, occurs much later with the addition of the cap geometry term. Furthermore the radius of the graphene cap in this region of constant radius is consistent with that of a C_{60} fullerene. This is due to the existence of 5-5 bonds in fullerenes C_n where

Parameter	Value	Citations
γ_{ga}	0.0646 eV/Å^2	[39]
γ_{gm}	0.137 eV/Å^2	[39, 40]
γ_{ma}	0.0918 eV/Å^2	[40]
$\gamma_{\ell t}$	1.96 eV/Å	[41]
$\gamma^*_{\ell t}$	0.80 eV/Å	[42, 43]
E_{55}	2.62 eV	[41]
E_{56}	4.54 eV	[41]
E_{66}	4.86 eV	[41]

Table 4.5.1: Parameter values

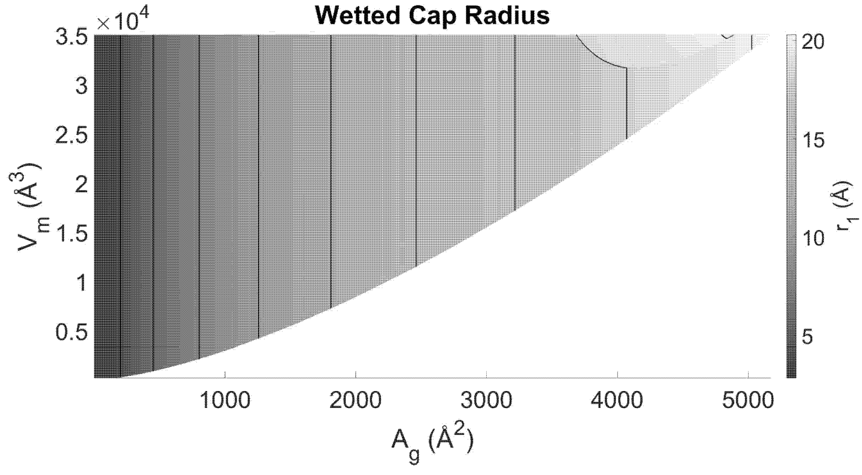

Figure 4.5.1: Graphene cap radius of the wetted state as V_{m} and A_{g} are varied

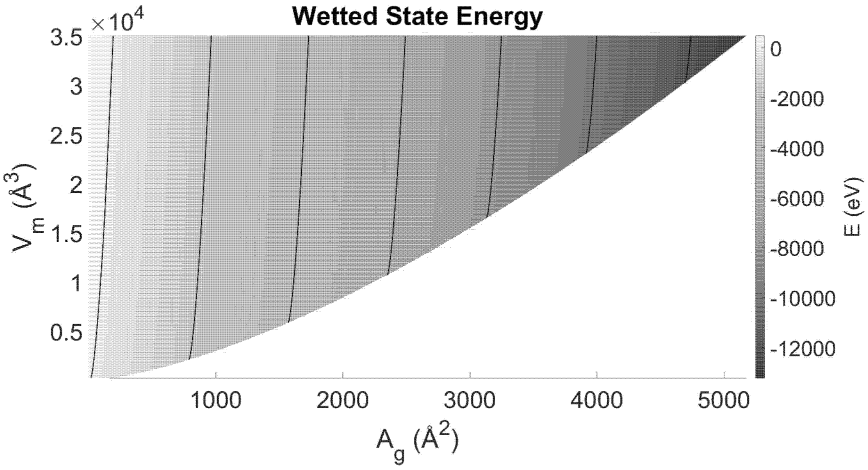

Figure 4.5.2: Energy of the wetted state as V_m and A_g are varied

$n < 60$. These bonds have a significantly lower bond energy which results in a large increase in energy for lower radii. Overall the radius of the graphene cap is similar to the previous model, however for larger values of A_g and $V_\mathrm{m} > 3000$ Å^3 we see a curved region corresponding to a larger cap radius.

The energy of the wetted system is illustrated by Fig. 4.5.2. As with the previous models the energy of the system still increases as V_m increases, however the addition of the cap geometry term causes the energy of the system to decrease as A_g increases. This justifies the growth of graphene caps in our model as larger graphene caps are now more energetically favourable than smaller ones. The addition of the cap geometry term has also resulted in minimum energy states with negative energy for the majority of parameter values. This is due to the strong binding energy between carbon atoms in the cap, which contributes a negative energy cost on the system. Furthermore, the change in behaviour at low values of A_g observed in the previous chapter is no longer present.

The distribution of radii seen in Fig. 4.5.3 again shows that the graphene cap radius begins to increase at a higher value of A_g compared to the previous model. However, for values of A_g beyond this the values of r_1 are consistent with those from the previous model. Unlike the graphene cap radius of the wetted state in Fig. 4.5.1, the graphene cap radius of the dewetted state does not undergo any

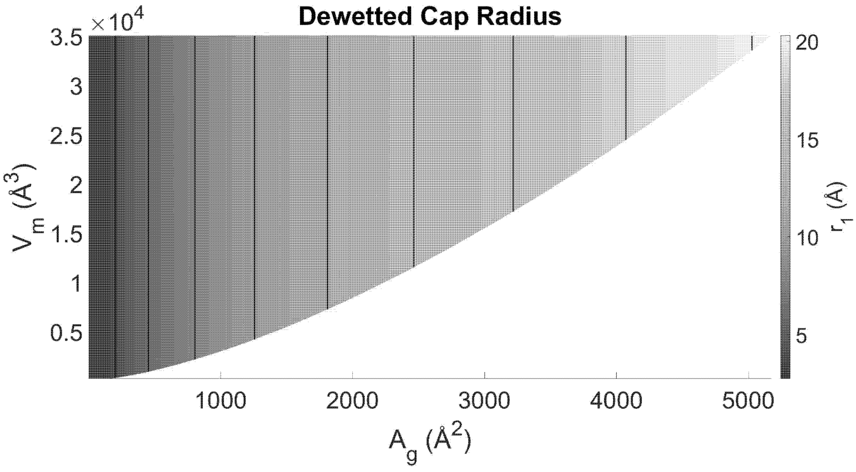

Figure 4.5.3: Graphene cap radius of the dewetted state as V_m and A_g are varied

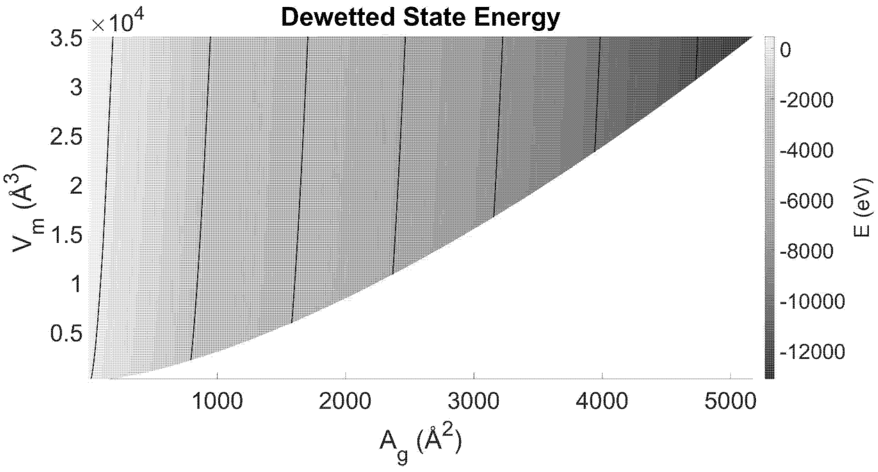

Figure 4.5.4: Energy of the dewetted state as V_m and A_g are varied

change in behaviour for the values of A_g and V_m observed in Fig. 4.5.3.

The effect of A_g and V_m on the energy of the dewetted state is shown in Fig. 4.5.4. As with the wetted state, the addition of the cap geometry term has resulted in the energy becoming negative, and the energy of the system now decreases as A_g is increased. The jump between solution types observed in Fig. 4.5.3 cannot be seen in the energy distribution. Instead there is a smooth change in the energy across this jump and there is little to no difference in the rate of change with respect to either variable on each side. Again, the change in behaviour of the energy at low values of A_g seen in the previous model is not present.

4.6 Conclusion

By adding the cap geometry term we have imposed an energy cost to the system that depends on the curvature of the graphene cap. As expected, this change to the model has shifted the graphene cap radius of the minimum energy configuration towards larger values but only modestly. The greatest impact that the cap geometry term has had on the model is the effect on the minimum energy of the wetted and dewetted states. Now as A_{g} increases the total energy of the system always decreases. As a consequence, larger graphene caps are more energetically favourable than smaller graphene caps and we might expect to see caps grow over time. This is more consistent with our understanding of CVD than our earlier models, which predict that larger caps necessarily have greater energies.

Chapter 5

Prediction of chirality

Using our model we may compare the energies of the optimal wetted and dewetted states to determine when dewetting is favoured energetically. This allows us make predictions for when dewetting is in some sense likely to occur for various parameter values. Moreover we may make predictions about the graphene cap structure at the time of likely dewetting and hence the cap structure of the resultant CNT. To achieve this, we first determine the minimal graphene cap area for which the energy of the wetted state exceeds that of the dewetted state. We assume that once dewetting has occured, further CNT growth does not change the shape of the existing cap structure. Therefore the radius of the resulting CNT will be approximately equal to the radius a of the edge from which it grows for the dewetted graphene cap. We determine the expected CNT chirality by finding the smallest CNT with integer-valued chiral vector numbers and radius at least equal to the cap-particle interface radius a. More realistically, the chirality of the resultant CNT will also depend on the arrangement of pentagonal carbon rings throughout the graphene cap. However, accounting for this would significantly complicate our model as we would likely need to consider a large number of graphene cap structures for any particular radius.

Parameter	Value	Citations
γ_{ga}	0.0646 eV/Å^2	[39]
γ_{gm}	0.137 eV/Å^2	[39, 40]
γ_{ma}	0.0918 eV/Å^2	[40]
$\gamma_{\ell t}$	1.96 eV/Å	[41]
$\gamma_{\ell t}^*$	0.80 eV/Å	[42, 43]
E_{55}	2.62 eV	[41]
E_{56}	4.54 eV	[41]
E_{66}	4.86 eV	[41]

Table 5.1.1: Parameter values for iron carbide catalyst particles at 800°C

5.1 Iron carbide catalyst particles

Using our model we can predict the graphene cap sizes for which dewetting becomes energetically favourable given various combinations of parameter values. Due to local minima in the energy surfaces, graphene caps do not necessarily dewet whenever it is energetically favourable to do so, and therefore it is necessary to consider the entire range for which dewetting is energetically favourable. By varying the volume of the metallic catalyst particle and again using the graphene cap area as a time-like variable we produce a two dimensional plot of the region in which dewetting is energetically favourable. We are then able to examine the dewetted state in this region to determine the chiralities, and therefore properties, of CNTs we can expect under various conditions. Furthermore, we may study how different catalyst materials affect this region and the range of CNT structures produced. We will again consider an iron carbide catalyst particle at 800°C, with parameters summarised in Table 5.1.1.

Fig. 5.1.1 shows a region in yellow in which the dewetted state is favoured energetically for iron carbide catalyst particles at 800°C. We can see that this region comprises the majority of the physically relevant portion of the plot, with only small regions to the left and right hand side where the wetted state is energetically favoured. This indicates that there is both a minimum and maximum graphene cap size required for dewetting to be favoured at any given value of V_{m}. The model there-

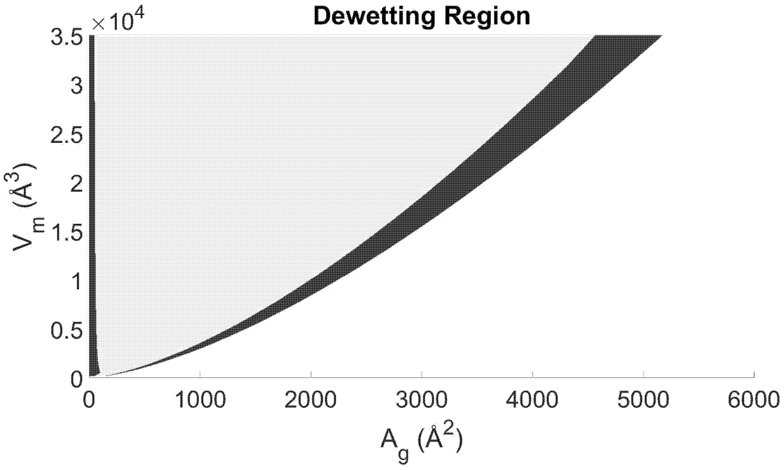

Figure 5.1.1: The region in which the dewetted state is energetically favourable, coloured yellow, for iron carbide catalyst particles

fore predicts that if a catalyst particle accumulates sufficient carbon on its surface the graphene cap will no longer dewet and instead grow to eventually encapsulate the catalyst particle in a graphene layer. This encapsulation of metallic catalyst particles by carbon during CVD has been observed in experiments [47, 48]. Further, since the maximum graphene cap size increases with V_m we would expect a larger range of potential CNT chiralities for larger metallic catalyst particles.

The range of graphene cap edge radii, a, for which dewetting is energetically favourable is shown in Fig. 5.1.2. Since the theoretical smallest CNT has chirality $(2, 2)$ [49], the smallest CNT that can be produced with a cap consisting only of pentagons or hexagons has chirality $(3, 3)$ [50]. Therefore we do not consider any values of a which would correspond to CNTs smaller than those with chirality $(3, 3)$. At small values of V_m the values of a for which dewetting is favourable are much lower. This is due to the catalyst particle restricting the maximum possible value of a to the catalyst particle's radius. For the majority of V_m values considered we find a range of a values between approximately 3.3 and 3.55 Å. This range widens for V_m values greater than approximately 28,000 $\mathrm{Å}^3$ with a new lower bound at $a \approx 3.25$ Å. This seemingly discontinuous change in the lower bound is a consequence of a changing rapidly at low values of A_g combined with the increased space

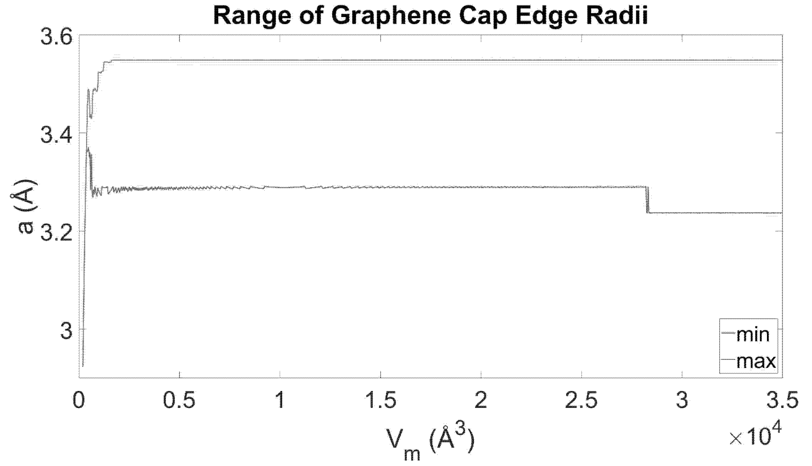

Figure 5.1.2: Range of a values for which dewetting is energetically favourable given iron carbide catalyst particles

between A_g values as the same number of points on the graph are spread between an increasing range of A_g values for larger values of V_m. According to our results, the variety of CNTs produced using iron carbide catalyst particles should have a uniform upper bound on radii, but a slowly decreasing lower bound on radii. This result is unintuitive given Fig. 5.1.1, which shows a wide range of potential A_g values for which dewetting could occur. As A_g increases the height of the cap becomes close to twice its radius, resulting in a nearly complete sphere with a relatively low value of a. Particles with volumes greater than 28,000 Å^3 are expected to additionally produce some CNTs with smaller radii. Amama et al. reports that iron particles with a diameter between 30–40 Å will grow single-walled CNTs with an average diameter of 30 Å [51]. This contradicts our model which predicts maximum CNT diameters of approximately 7 Å for catalyst particles with diameters up to 40 Å, over 4 times smaller than the observed average of 30 Å.

We can also use the data in Fig. 5.1.1 to produce a range of potential CNT chiralities for each value of V_m, and hence produce an unweighted proportion of those CNTs with metallic properties. This is shown in Fig. 5.1.3. We see that for very low values of V_m the proportion of metallic CNTs varies significantly, ranging from 0 to 100%. However, for all other values of V_m the proportion of metallic CNTs

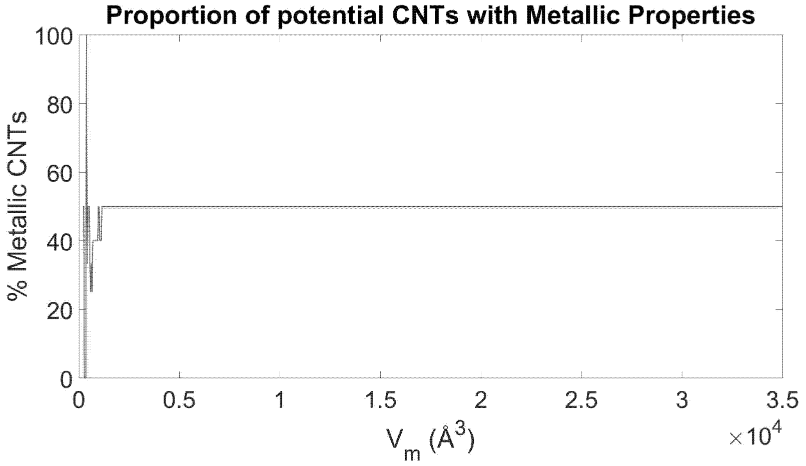

Figure 5.1.3: The proportion of CNTs in the dewetted state which are metallic for iron carbide catalyst particles

is constant and equal to 50% despite the larger range of a values for large V_{m} seen in Fig. 5.1.2.

5.2 Nickel carbide catalyst particles

To observe the predictions of our model for different catalyst compositions, we now consider a nickel carbide catalyst particle at 1550°C. Using (2.6.1) we calculate the value of γ_{ga} at 1550°C to be 0.058 eV/Å^2. Nizhenko et al. found that the surface free energy coefficient of nickel carbide in atmosphere is given by

$$\gamma_{\mathrm{ma}} = 2316 - 0.66\tau \ \mathrm{mJ/m}^2$$

where τ is the temperature in degrees Celsius [52]. Using this gives us a value of 0.081 eV/Å^2 for γ_{ma}. For the contact angle of nickel carbide with carbon, Naidich et al. reported a value of 115° [53]. Using Young's equation, (2.2.2), we calculate γ_{gm} to be 0.092 eV/Å^2. To calculate the last of our changed parameters, $\gamma^{\star}_{\ell t}$, we first need the bond dissociation energy between nickel and carbon. Sambasiva Rao et al. reported a value of 3.454 eV, which gives us a value of 1.39 eV/Å along the graphene cap edge [54]. Since nickel carbide consists of three nickel atoms for each

Parameter	Value	Citations
γ_{ga}	0.058 eV/Å^2	[39]
γ_{gm}	0.092 eV/Å^2	[39, 52, 53]
γ_{ma}	0.081 eV/Å^2	[52]
$\gamma_{\ell t}$	1.96 eV/Å	[41]
$\gamma_{\ell t}^{\star}$	1.45 eV/Å	[43, 54]
E_{55}	2.62 eV	[41]
E_{56}	4.54 eV	[41]
E_{66}	4.86 eV	[41]

Table 5.2.1: Parameter values for nickel carbide catalyst particles at 1550°C

carbon atom, we use a weighted average as performed in § 3.3 to calculate $\gamma_{\ell t}^{\star}$,

$$\gamma_{\ell t}^{\star} = \frac{3(1.39) + 1.63}{4} = 1.45 \text{ eV/Å}.$$

Table 5.2.1 summarises all our parameter values for nickel carbide catalyst particles.

Fig. 5.2.1 shows the region in which dewetting is energetically favourable for nickel carbide catalyst particles. Compared to iron carbide catalyst particles, our model predicts a much smaller region for which dewetting is energetically favourable for nickel carbide catalyst particles. The left side of this region appears to be bounded by a constant value of the graphene cap area A_{g} whereas the right side ends at a value of A_{g} which increases as V_{m} increases. Although our model predicts that dewetting will occur, the optimal dewetted state in the dewetting region has a very small value for the radius of the graphene cap edge, a, and therefore would not result in the growth of any CNTs. Given that nickel is a common catalyst material for the production of CNTs via CVD, this prediction does not match what is seen in practice, and the reason for this disparity is not clear. Additionally, Sivakumar et al. reported that under similar conditions multi-walled CNTs grown from nickel catalysts had a greater internal diameter than those grown on iron catalysts [55]. This is not reflected by our model, which predicts greater values of a where dewetting is energetically favourable for iron catalysts.

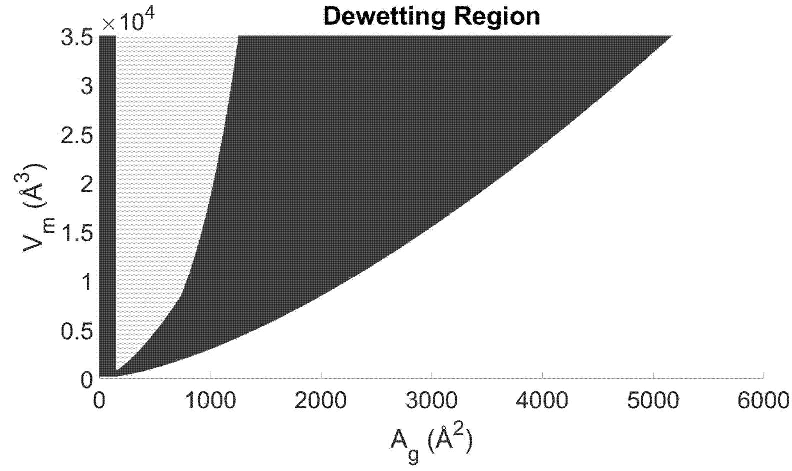

Figure 5.2.1: The region in which the dewetted state is energetically favourable, coloured yellow, for nickel carbide catalyst particles

5.3 Gold catalyst particles

Another catalyst material used for the production of CNTs via CVD is gold. In this case we consider gold catalyst particles at 1100°C. The value of γ_{ga} at 1100°C, as given by (2.6.1), is 0.062 eV/Å^2. Lee et al. report the surface free energy coefficient of liquid gold at 1100°C to be 1138 mN/m, so we have that $\gamma_{\text{ma}} = 0.071$ eV/Å^2 [56]. The contact angle of gold on graphite was reported by Joonho et al. to be 129° [57]. Using Young's equation, (2.2.2), we calculate the value of γ_{gm} to be 0.107 eV/Å^2. Liu et al. reported that the bond dissociation energy for double bonds between gold and carbon is 4.08 eV, which gives us 1.65 eV/Å along the graphene cap edge [58]. Since the metallic catalyst particle is purely gold atoms, we don't use a weighted average to calculate $\gamma^{\star}_{\ell t}$ and we simply use $\gamma^{\star}_{\ell t} = 1.65$ eV/Å. Our parameter values are summarised in Table 5.3.1.

The dewetting region for gold catalyst particles shown in Fig. 5.3.1 is larger than that for nickel carbide catalyst particles, but smaller than that for iron carbide catalyst particles. The left side of the dewetting region has an approximately constant minimum value of A_{g} for $V_{\text{m}} > 600$ Å^3. However, for smaller values of V_{m} the minimum value of A_{g} required for dewetting increases. As with the other catalyst materials, the maximum value of A_{g} for which dewetting occurs increases as V_m is

Parameter	Value	Citations
γ_{ga}	0.062 eV/Å^2	[39]
γ_{gm}	0.107 eV/Å^2	[39, 56, 57]
γ_{ma}	0.071 eV/Å^2	[56]
$\gamma_{\ell t}$	1.96 eV/Å	[41]
$\gamma_{\ell t}^*$	1.65 eV/Å	[43, 58]
E_{55}	2.62 eV	[41]
E_{56}	4.54 eV	[41]
E_{66}	4.86 eV	[41]

Table 5.3.1: Parameter values for gold catalyst particles at 1100°C

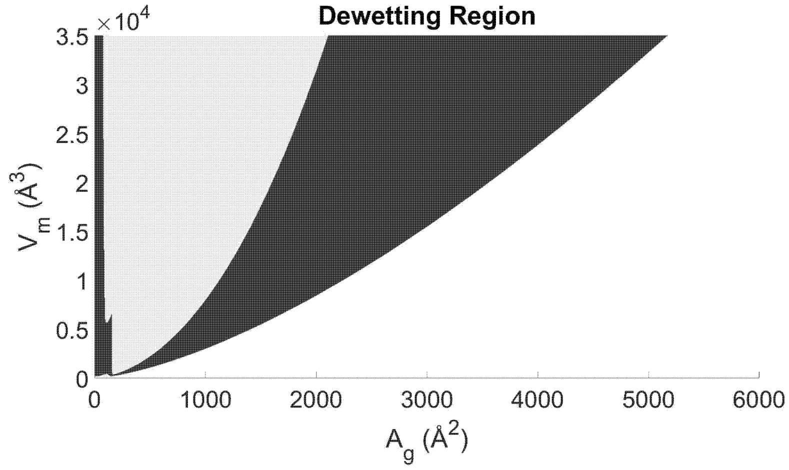

Figure 5.3.1: The region in which the dewetted state is energetically favourable, coloured yellow, for gold catalyst particles

increased.

Fig. 5.3.2 shows the range of graphene cap edge radii over which dewetting is favourable for gold particles of volume V_{m}. For $V_{\mathrm{m}} < 1,000$ Å^3 our model predicts edge radii ranging from approximately 2.6–3.4 Å. Between $V_{\mathrm{m}} = 1,000$ Å^3 and 5,000 Å^3 there is a gap in which there are no cap-particle systems for which dewetting is energetically favourable and the dewetted state has a sufficiently large value of a. For $V_{\mathrm{m}} > 5,000$ Å^3 the minimum graphene cap edge radius is approximately 3.5 Å regardless of V_{m}. The maximum graphene cap edge radius ranges between 4 and 4.5 Å, and increases as V_{m} is increased. Our model predicts that for gold catalyst particles we can expect the production of CNTs with radii of at least 3.5 Å with an upper bound determined by the maximum particle volume. Compared

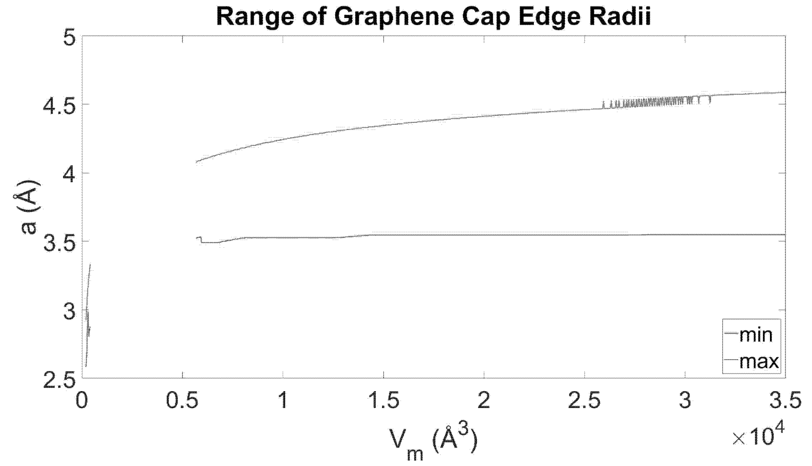

Figure 5.3.2: Range of a values for which dewetting is energetically favourable given gold catalyst particles

to iron carbide, gold catalyst particles produce significantly larger CNTs with the lower bound on radii being approximately equal to the upper bound for iron carbide catalyst particles. Takagi et al. report that the diameter of single-walled CNTs produced on gold particles of diameters in the range 10–30 Å had an approximate range of 9–18 Å [59]. However, our model predicts an upper bound on CNT diameter slightly less than 9 Å in the same region.

The percentage of CNTs with metallic properties expected to be produced from gold catalyst particles is shown in Fig. 5.3.3. For $V_{\mathrm{m}} > 5,000$ Å^3, the percentage of metallic CNTs initially increases from approximately 33% up to approximately 46% at $V_{\mathrm{m}} \approx 8,000$ Å^3. For larger values of V_{m} the percentage decreases to approximately 30% for $V_{\mathrm{m}} > 32,000$ Å^3. The yield of metallic CNTs produced by gold catalyst particles with volume greater than 5,000 Å^3 is always less than the 50% predicted for iron carbide catalyst particles. Therefore our model predicts that iron carbide is more efficient for the production of metallic CNTs while gold is more efficient for the production of semiconducting CNTs.

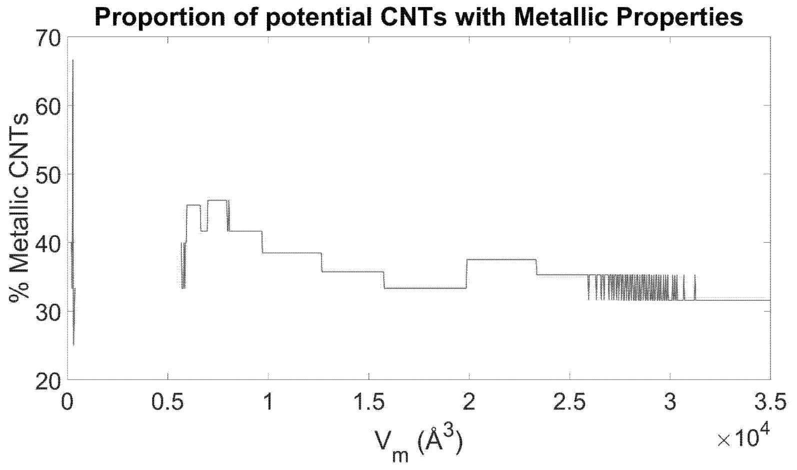

Figure 5.3.3: The proportion of CNTs in the dewetted state which are metallic for gold catalyst particles

Chapter 6

Conclusion

By applying the calculus of variations we were able to calculate the minimum energy state of a graphene cap wetted to a metallic catalyst particle, and hence develop a model for the growth of a graphene cap during chemical vapour deposition (CVD). Furthermore, by considering the minimum energy state of the dewetted system we could determine when dewetting is most likely to occur under a given set of parameters and estimate the chirality of the resulting carbon nanotubes (CNTs).

To develop this model we began with a surface free energy model for which the energy of the system was determined by the size of the graphene-metal, graphene-atmosphere, and metal-atmosphere interfaces. This initial model did not account for many of the energy costs within the system, and as such did not produce many interesting results. Instead the minimum energy state was always a spherical metallic catalyst particle with the graphene cap resting on its surface. To expand on the model we then added a line tension term to our initial model in order to account for the dangling chemical bonds at the edge of the graphene cap. The addition of the line tension term resulted in a model which favoured graphene caps with lower radii as these caps better balance the combined energy cost of the interfacial surfaces and the dangling chemical bonds at the edge of the graphene cap. As the final addition to our model we included a cap geometry term which would impose an energy cost on more curved graphene caps. This cap geometry term was calculated by subtracting the binding energy of each carbon-carbon bond in the graphene cap. Highly curved graphene caps have a significantly lower average binding energy between carbon atoms, so the addition of the cap geometry term made caps with very low radii unfavourable.

To test the model we used parameter values corresponding to three types of metallic catalyst particles with volumes of up to 35,000 Å^3. The types of metallic catalyst particles considered were iron carbide, nickel carbide, and gold. We then compared the results to experimental observations of nanotubes grown using the same type of metallic catalyst particle. For iron carbide and gold our model predicted CNTs with much smaller diameters than those observed in experiments. In the case of iron carbide, our model predicting diameters ranging from approximately 6.5 to 7.1 Å while experiments have shown that the average diameter of CNTs produced using similarly sized catalyst particles was 30 Å. For gold catalyst particles our model showed better agreement with experimental observations, but still predicted significantly lower CNT diameters. Our model predicted CNT diameters between 7 and 9 Å while a range of 9–18 Å was observed in experiments. For nickel carbide, despite being a commonly used catalyst material and our model predicting that dewetting would occur, no CNTs were predicted to be produced. This is because the CNT radii predicted were lower than that which is physically possible. In all cases our model predicted the production of CNTs with radii that are much smaller than those observed in experiments. Using our model we also predicted the percentage of metallic CNTs produced with each catalyst. For iron carbide our model predicted a consistent ratio of 50% metallic CNTs as the metallic catalyst particle volume was increased. For gold, however, our model predicted a lower percentage with an approximate range of 33–46% metallic as the catalyst particle volume was varied.

In order to improve our model and produce more accurate results we could remove or modify our assumptions about the graphene cap and metallic catalyst particle system. For example, we could assume that the graphene cap is shaped like a surface of revolution instead of a spherical cap and use variational calculus to determine which surface of revolution is optimal throughout graphene cap growth. We could also try generalising further and remove the assumption on the graphene cap's shape entirely. Alternatively we could model the accumulation of carbon atoms and interatomic bonds in the graphene cap discretely so that the exact structure of the graphene cap could be predicted. This would be a significant change to the model but should greatly improve our prediction of CNT chirality. Another potential extension to our model could be to improve how dewetting is handled by the model, which is currently based on the difference between the energies of the optimal wetted and dewetted states. Instead we could also compare the energies of several intermediate states between the wetted and dewetted states to determine if dewetting is likely to occur. This change may also allow us to determine suboptimal dewetted states which would be more likely to result from dewetting of a particular wetted system. This could significantly improve the prediction of CNT chirality by our model. Finally, we could expand on the model to consider multi-walled CNTs which are commonly produced via CVD. Although this would be significantly more complex than considering only single-walled CNTs, the expanded model would give a more complete prediction of the CNTs produced under specific conditions.

Bibliography

[1] H. W. Kroto, J. R. Heath, S. C. O'Brien, R. F. Curl, and R. E. Smalley. C60: Buckminsterfullerene. *Nature*, 318:162–163, 1985.

[2] H. Prinzbach, A. Weiler, P. Landenberger, F. Wahl, J. Wrth, L. T. Scott, M. Gelmont, D. Olevano, and B. v. Issendorff. Gas-phase production and photoelectron spectroscopy of the smallest fullerene, C_{20}. *Nature (London)*, 407(6800):60–63, 2000.

[3] P. Schwerdtfeger, L. N. Wirz, and J. Avery. The topology of fullerenes. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 5(1):96–145, 2015.

[4] A. T. Balaban, X. Liu, D. J. Klein, D. Babic, T. G. Schmalz, W. A. Seitz, and M. Randic. Graph invariants for fullerenes. *Journal of Chemical Information and Computer Sciences*, 35(3):396–404, 1995.

[5] A. M. Livshits and Y. E. Lozovik. Cut-and-unfold approach to fullerene enumeration. *Journal of Chemical Information and Computer Sciences*, 44(5):1517–1520, 2004.

[6] S. Iijima and T. Ichihashi. Single-shell carbon nanotubes of 1-nm diameter. *Nature*, 363:603–605, 1993.

[7] D. S. Bethune, C. H. Kiang, M. S. de Vries, G. Gorman, R. Savoy, J. Vazquez, and R. Beyers. Cobalt-catalysed growth of carbon nanotubes with single-atomic-layer walls. *Nature*, 363:605–607, 1993.

[8] M. S. Dresselhaus, G. Dresselhaus, and R. Saito. Physics of carbon nanotubes. *Carbon (New York)*, 33(7):883–891, 1995.

[9] J. Bernholc, D. Brenner, M. Buongiorno Nardelli, V. Meunier, and C. Roland. Mechanical and electrical properties of nanotubes. *Annual Review of Materials Research*, 32(1):347–375, 2002.

[10] M. S. Dresselhaus. Down the straight and narrow. *Nature (London)*, 358(6383):195–196, 1992.

[11] R. Saito, G. Dresselhaus, and M. S. Dresselhaus. *Physical Properties of Carbon Nanotubes*. World Scientific, 1998.

[12] B. C. Edwards. The space elevator: a new tool for space studies.(report). *Gravitational and Space Biology*, 16(2):101–105, 2003.

[13] M. Zhang, S. Fang, A. A. Zakhidov, S. B. Lee, A. E. Aliev, C. D. Williams, K. R. Atkinson, and R. H. Baughman. Strong, transparent, multifunctional, carbon nanotube sheets. *Science*, 309(5738):1215–1219, 2005.

[14] S. J. Tans, A. R. M. Verschueren, and C. Dekker. Room-temperature transistor based on a single carbon nanotube. *Nature*, 393(6680):49–52, 1998.

[15] P. G. Collins, M. S. Arnold, and P. Avouris. Engineering carbon nanotubes and nanotube circuits using electrical breakdown. *Science*, 292(5517):706–709, 2001.

[16] D. Chattopadhyay, I. Galeska, and F. Papadimitrakopoulos. A route for bulk separation of semiconducting from metallic single-wall carbon nanotubes. *Journal of the American Chemical Society*, 125(11):3370–3375, 2003.

[17] M. S. Arnold, A. A. Green, J. F. Hulvat, S. I. Stupp, and M. C. Hersam. Sorting carbon nanotubes by electronic structure using density differentiation. *Nature Nanotechnology*, 1(1):60–65, 2006.

[18] J. A. Rodrguez-Manzo, M. Terrones, H. Terrones, H. W. Kroto, L. Sun, and F. Banhart. In situ nucleation of carbon nanotubes by the injection of carbon atoms into metal particles. *Nature Nanotechnology*, 2(5):307–311, 2007.

[19] V. Jourdain and C. Bichara. Current understanding of the growth of carbon nanotubes in catalytic chemical vapour deposition. *Carbon*, 58:2–39, 2013.

[20] M. J. Bronikowski, P. A. Willis, D. T. Colbert, K. A. Smith, and R. E. Smalley. Gas-phase production of carbon single-walled nanotubes from carbon monoxide via the HiPco process: A parametric study. *Journal of Vacuum Science & Technology. A, Vacuum, Surfaces, and Films*, 19(4):1800–1805, 2001.

[21] T. Guo, P. Nikolaev, A. Thess, D. T. Colbert, and R. E. Smalley. Catalytic growth of single-walled manotubes by laser vaporization. *Chemical physics letters*, 243(1-2):49–54, 1995.

[22] F. W. A. H. Geurts and A. Sacco. The relative rates of the boudouard reaction and hydrogenation of CO over Fe and Co foils. *Carbon (New York)*, 30(3):415–418, 1992.

[23] J. Kong, A. M. Cassell, and H. Dai. Chemical vapor deposition of methane for single-walled carbon nanotubes. *Chemical physics letters*, 292(4-6):567–574, 1998.

[24] S. Maruyama, R. Kojima, Y. Miyauchi, S. Chiashi, and M. Kohno. Low-temperature synthesis of high-purity single-walled carbon nanotubes from alcohol. *Chemical physics letters*, 360(3-4):229–234, 2002.

[25] C. L. Cheung, A. Kurtz, H. Park, and C. M. Lieber. Diameter-controlled synthesis of carbon nanotubes. *Journal of Physical Chemistry. B*, 106(10):2429–2433, 2002.

[26] M.-F. C. Fiawoo, A.-M. Bonnot, H. Amara, C. Bichara, J. Thibault-Pénisson, and A. Loiseau. Evidence of correlation between catalyst particles and the single-wall carbon nanotube diameter: a first step towards chirality control. *Physical review letters*, 108(19):195503, 2012.

[27] F. Ding, P. Larsson, J. A. Larsson, R. Ahuja, H. Duan, A. Rosén, and K. Bolton. The importance of strong carbonmetal adhesion for catalytic nucleation of single-walled carbon nanotubes. *Nano letters*, 8(2):463–468, 2008.

[28] K. A. Shah and B. A. Tali. Synthesis of carbon nanotubes by catalytic chemical vapour deposition: A review on carbon sources, catalysts and substrates. *Materials Science in Semiconductor Processing*, 41:67–82, 2016.

[29] S. Helveg, C. López-Cartes, J. Sehested, P. L. Hansen, B. S. Clausen, J. R. Rostrup-Nielsen, F. Abild-Pedersen, and J. K. Nørskov. Atomic-scale imaging of carbon nanofibre growth. *Nature*, 427(6973):426–429, 2004.

[30] F. Yang, X. Wang, D. Zhang, J. Yang, D. Luo, Z. Xu, J. Wei, J.-Q. Wang, Z. Xu, F. Peng, X. Li, R. Li, Y. Li, M. Li, X. Bai, F. Ding, and Y. Li. Chirality-specific growth of single-walled carbon nanotubes on solid alloy catalysts. *Nature*, 510(7506):522, 2014.

[31] G. Lolli, L. Zhang, L. Balzano, N. Sakulchaicharoen, Y. Tan, and D. E. Resasco. Tailoring (n, m) structure of single-walled carbon nanotubes by modifying reaction conditions and the nature of the support of CoMo catalysts. *Journal of Physical Chemistry B*, 110(5):2108–2115, 2006.

[32] T. Hiraoka, S. Bandow, H. Shinohara, and S. Iijima. Control on the diameter of single-walled carbon nanotubes by changing the pressure in floating catalyst CVD. *Carbon*, 44(9):1853–1859, 2006.

[33] M. S. Dresselhaus, G. Dresselhaus, and P. Avouris. *Carbon Nanotubes Synthesis, Structure, Properties, and Applications.* Topics in Applied Physics, 80. 2001.

[34] H. Amara and C. Bichara. Modeling the growth of single-wall carbon nanotubes. *Topics in current chemistry (Cham)*, 375(3):55–55, 2017.

[35] K. Raji and C. B. Sobhan. Simulation and modeling of carbon nanotube synthesis: current trends and investigations. *Nanotechnology reviews (Berlin)*, 2(1):73–105, 2013.

[36] N. A. Morton. Modelling nucleation of carbon nanotube caps on floating molten metallic catalysts (unpublished honours thesis). 2017. The University of Auckland.

[37] T. Young. An essay on the cohesion of fluids. *Philosophical transactions of the Royal Society of London*, 95:65–87, 1805.

[38] P. G. de Gennes. Wetting: statics and dynamics. *Reviews of modern physics*, 57(3):827–863, 1985.

[39] S. K. Rhee. Critical surface energies of Al2O3 and graphite. *Journal of the American Ceramic Society*, 55(6):300–303, 1972.

[40] V. I. Nizhenko and L. I. Floka. Contact reaction of graphite with liquid iron and iron base melts. *Soviet Powder Metallurgy and Metal Ceramics*, 13(6):487–492, 1974.

[41] Z. Slanina and E. Osawa. Average bond dissociation energies of fullerene. *Fullerene Science and Technology*, 5(1):167–175, 1997.

[42] C. G. Riordan and J. Halpern. Kinetics, mechanism and thermodynamics of iron carbon bond dissociation in organoiron porphyrin complexes. *Inorganica Chimica Acta*, 243(1):19–24, 1996.

[43] T. L. Cottrell. *The strengths of chemical bonds.* Butterworth, London, 2^{nd} edition, 1958.

[44] H. W. Kroto. The stability of the fullerenes C_n, with $n = 24, 28, 32, 36, 50, 60,$ and 70. *Nature*, 329:529–531, 1987.

[45] N. Frederick and D. Tomanek. C_n fullerenes, 2018. Available at `http://www.nanotube.msu.edu/fullerene/fullerene-isomers.html`, accessed 25/3/2020.

[46] S. Weber. Mitsuho yoshidas fullerene library. Available at `http://www.jcrystal.com/steffenweber/gallery/Fullerenes/Fullerenes.html`, accessed 25/3/2021.

[47] N. Sano, H. Akazawa, T. Kikuchi, and T. Kanki. Separated synthesis of iron-included carbon nanocapsules and nanotubes by pyrolysis of ferrocene in pure hydrogen. *Carbon*, 41(11):2159–2162, 2003.

[48] L. Ci, Z. Rao, Z. Zhou, D. Tang, X. Yan, Y. Liang, D. Liu, H. Yuan, W. Zhou, G. Wang, W. Liu, and S. Xie. Double wall carbon nanotubes promoted by sulfur in a floating iron catalyst CVD system. *Chemical physics letters*, 359(1):63–67, 2002.

[49] X. Zhao, Y. Liu, S. Inoue, T. Suzuki, R. O. Jones, and Y. Ando. Smallest carbon nanotube is 3 Å in diameter. *Physical review letters*, 92(12):125502, 2004.

[50] L.-C. Qin, X. Zhao, K. Hirahara, Y. Miyamoto, Y. Ando, and S. Iijima. The smallest carbon nanotube. *Nature (London)*, 408(6808):50, 2000.

[51] P. B. Amama, C. L. Pint, L. McJilton, S. M. Kim, E. A. Stach, P. T. Murray, R. H. Hauge, and B. Maruyama. Role of water in super growth of single-walled carbon nanotube carpets. *Nano letters*, 9(1):44–49, 2009.

[52] V. I. Nizhenko and L. I. Floka. Surface parameters of carbon-saturated nickel-copper liquid alloys. *Powder metallurgy and metal ceramics*, 39(9):462–466, 2000.

[53] Y. V. Naidich, V. M. Perevertailo, and G. M. Nevodnik. Wetting of graphite by nickel as affected by the liquid-phase dissolution process of carbon. *Soviet Powder Metallurgy and Metal Ceramics*, 10(1):45–47, 1971.

[54] P. Sambasiva Rao, R. R. Reddy, and D. M. Potukuchi. Estimation of dissociation energy of NiC molecule. *Journal of quantitative spectroscopy and radiative transfer*, 98(1):81–84, 2006.

[55] V. M. Sivakumar, A. Z. Abdullah, A. R. Mohamed, and S. P. Chai. Optimized parameters for carbon nanotubes synthesis over Fe and Ni catalysts via methane CVD. *Reviews on Advanced Materials Science*, 27:25–30, 2011.

[56] J. Lee, M. Nakamoto, and T. Tanaka. Thermodynamic study on the melting of nanometer-sized gold particles on graphite substrate. *Journal of materials science*, 40(9):2167–2171, 2005.

[57] J. Lee, T. Tanaka, K. Seo, N. Hirai, J.-G. Lee, and H. Mori. Wetting of Au and Ag particles on monocrystalline graphite substrates. *Rare metals*, 25(5):469–472, 2006.

[58] H.-G Liu, X.-G. Xiong, P. D. Dau, Y.-L. Wang, D.-L Huang, J. Li, and L.-S. Wang. Probing the nature of goldcarbon bonding in goldalkynyl complexes. *Nature communications*, 4(1):2223–2223, 2013.

[59] D. Takagi, Y. Homma, H. Hibino, S. Suzuki, and Y. Kobayashi. Single-walled carbon nanotube growth from highly activated metal nanoparticles. *Nano letters*, 6(12):2642–2645, 2006.

Appendix A

Calculation of minimum energy state script

A.1 Explanation of script

In order to automate the generation of results from our model, and to find solutions that are too difficult to derive by analytic means, we implement the model in MATLAB. The figures in §2.4, §3.3, and §4.5 were generated using this implementation. This implementation of our model makes use of the golden section search algorithm to determine the location of energy minima to within a prescribed tolerance.

The golden section search algorithm is a numerical method to determine the location of an extremum for a given function inside a prescribed interval. In this case we are interested in finding the location of minima for our cap-particle system energy function, and we implement it in the functions at lines 534 and 538. We begin the golden section search algorithm with two energy values, E_1 and E_4, corresponding to the energy of our system at $h_1 = \eta_1$ and $h_1 = \eta_4$ where $\eta_1 < \eta_4$. To calculate the energy corresponding to a particular value of h_1 we use the functions at lines 402 and 460 for the wetted and dewetted states respectively. We then define two other points, η_2 and η_3 such that

$$\eta_2 = \eta_4 - \frac{\eta_4 - \eta_1}{\varphi}, \quad \eta_3 = \eta_1 + \frac{\eta_4 - \eta_1}{\varphi} \tag{A.1.1}$$

where φ is the golden ratio, $\varphi = (1 + \sqrt{5})/2$. We then calculate the value of E at $h_1 = \eta_2$ and η_3 and label these E_2 and E_3 respectively. Now we compare the energy at each point to determine if the minimum is located between points η_1 and η_3 or η_2 and η_4. If $E_1 > E_2$ and $E_2 < E_3$ then we know a minimum lies between η_1 and η_3. We can then redefine our points such that $\eta_3 = \eta_2$, $\eta_4 = \eta_3$, and define a new η_2 according to (A.1.1). Otherwise, if $E_2 > E_3$ and $E_3 < E_4$ then we know a minimum lies between η_2 and η_4. In this case we redefine our points such that $\eta_1 = \eta_2$, $\eta_2 = \eta_3$, and redefine η_3 according to (A.1.1). This process can be iterated to reduce the size of the interval $[\eta_1, \eta_4]$ to within the desired tolerance.

For a given set of parameter values, we start by calculating the energy of the cap particle system assuming that the cap is shaped like part of a fullerene C_n, with n being equal to 20, or as small as possible such that the area of the fullerene C_n is greater than the graphene cap area A_g. The value of n is determine in the script by the function at line 626. From our assumption we are able to determine the radius of the graphene cap given n, and hence calculate h_1 given A_g. We then increment n using an exponentially increasing step size and recalculate the energy of the system at each step using the appropriate function. Beginning from the third step, we compare the three most recently calculated values of the energy to determine whether a local minimum lies between these points. For example, at step k for some $k > 2$ a local minimum is detected when the energy at step $k - 1$ is less than that at steps $k - 2$ and k. If a local minimum is detected at this step then we perform a golden section search on the range bounded by the value of n at steps $k - 2$ and k. Once the golden section search reduces the range to within the prescribed tolerance, we set the location of the global minimum to be at the point calculated to have the lowest energy. This process repeats whenever additional local minima are detected, and the global minimum is updated if the energy at one of these minima is less than the recorded global minimum.

After the initial search is complete we then evaluate the energy at the endpoints of our search range. If the energy at either endpoint is less than the recorded global minimum then we update the global minimum to the value found at the endpoint. For both the wetted and dewetted states the location of the left hand endpoint is either determined by A_{g} or, for low A_{g}, is given by $n = 20$. The lower bound at $n = 20$ is due to C_{20} being the smallest possible fullerene, so smaller values of n are not physically relevant. The limit on n due to high values of A_{g} is to prevent the model predicting caps with a height greater than twice its radius as these caps are also unphysical. The right hand side endpoint has no maximum for the wetted state as caps with arbitrarily large radii, corresponding to large n, are valid in this model. However, for the dewetted state there is a limit on the right hand side endpoint as large values of n would result in a cap that deforms the metallic catalyst particle. This limit is applied in line 490, where the function that outputs the energy of the system returns Inf if the radius of the graphene cap is greater than or equal to that of the metallic catalyst particle while $h_1 < r_1$. A graphene cap with radius and height satisfying this condition would necessarily be in contact with the metallic catalyst particle and is therefore inconsistent with our definition of the dewetted state. Therefore we have a limit on values of n for the dewetted state that depends on V_{m}.

A.2 MATLAB implementation

```
1 clear
2 close all
3 Vm = 35000;                             %Volume of metal particle
4 Ag = (36*pi*Vm^2)^(1/3)/2;              %Area of the carbon cap
5 var1 = 1;                               %The variable to be varied (index of varlist)
6 min_var1 = 0.1;                         %Minimum value of the parameter being varied
7 max_var1 = (36*pi*Vm^2)^(1/3);          %Maximum value of the parameter being varied
```

```
var2 = 7;                               %The second variable to be varied
min_var2 = 200;
max_var2 = 35000;

if var1 == 1 && var2 == 7
    Ag_max = (36*pi*max_var2^2)^(1/3)/10;
end

n = 1000;                                %Number of iterations over range of var values
n2 = 150;                               %Number of initial iterations over fullerene size
step_scale = 1.08;                      %step size is multiplied by this each iteration
step_initial = 0.25;                     %Initial step size

line_tension_term = 1;                  %0 = no line tension energy, 1 = line tension energy
cap_energy_term = 1;                    %0 = no cap energy, 1 = cap energy

gamma_ma = 0.0918;                      %Surface tension coefficient of the metallic particle
gamma_ga = 0.0646;                      %graphene-atmosphere interface
gamma_gm = 0.137;                       %graphene-metal interface
gamma_lt = 1.96;                        %Line tension energy coefficient
gamma_lt_star = 0.8;                    %dangling bond energy coefficient
varlist = [Ag, gamma_ma, gamma_lt, gamma_lt_star, gamma_ga, gamma_gm, Vm];   %Vector of parameters

var1_vals = linspace(min_var1, max_var1, n);   %Initialising vectors
var2_vals = linspace(min_var2, max_var2, n);
%These matrices store the wetted state data
E = NaN(n,n);                         %Total energy
E_comp = cell(n,n);                     %Vector [E_ma, E_lt, E_c]
E_ma = NaN(n,n);                      %Energy of the metal atmosphere interface
E_lt = NaN(n,n);                      %Line tension energy
E_c = NaN(n,n);                       %Cap geometry energy
h1 = NaN(n,n);                        %Height of carbon cap
h2 = NaN(n,n);                        %Height of metal particle
r1 = NaN(n,n);                        %Radius of carbon cap
r2 = NaN(n,n);                        %Radius of metal particle
fullerene_size = NaN(n,n);            %Size of fullerene with radius r1
contact_angle = NaN(n,n);             %Contact angle between cap and particle
flag_mat = NaN(n,n);                  %Flag data, 1 = local minimum, 2 = minimum at lowest value, 3 = minimum at end of search range
%These matrices store the dewetted state data
E_dewetted = NaN(n,n);
E_comp_dewetted = cell(n,n);
E_ma_dewetted = NaN(n,n);
E_lt_dewetted = NaN(n,n);
E_c_dewetted = NaN(n,n);
h1_dewetted = NaN(n,n);
h2_dewetted = NaN(n,n);
r1_dewetted = NaN(n,n);
r2_dewetted = ((3*varlist(7))/(4*pi))^(1/3);
fullerene_size_dewetted = NaN(n,n);
contact_angle_dewetted = NaN(n,n);
flag_mat_dewetted = NaN(n,n);

V1 = NaN(n,n);                        %Matrix of var1 values
V2 = NaN(n,n);                        %Matrix of var2 values

for j = 1:n

```

```
    varlist(var2) = var2_vals(j);               %varlist is updated
    if var1 == 1 && var2 == 7
        %Sets appropriate limits if var2 is particle volume and var1 is cap
        %area
        max_var1 = (36*pi*varlist(var2)^2)^(1/3)/10;
        n_var1 = ceil(n*max_var1/Ag_max);
        if n_var1 == 1
            n_var1 = 2;
        end
        var1_vals = linspace(min_var1, max_var1, n_var1);
    end

    for i = 1:n_var1
        varlist(var1) = var1_vals(i);           %Sets the new value of the parameter being varied
        V1(i,j) = varlist(var1);                %Variable value matrices are updated
        V2(i,j) = varlist(var2);
        loop_counter = 1;                       %Tracks the progress of the while loop
        n_atoms = starting_point(varlist(1));   %Initial value of n_atoms is chosen based on cap area
        n_atoms_vals = zeros(1,2);              %Vector of n_atoms values
        n_atoms_vals(1) = n_atoms;              %First value in vector is updated
        loop_flag = 0;                          %Becomes 1 if a minimum is found
        energy = zeros(1,2);                    %Vector of energy values
        h1_vals = zeros(1,2);                   %Vector of h1 values
        h2_vals = zeros(1,2);                   %Vector of h2 values
        r1_vals = zeros(1,2);                   %Vector of r1 values
        r2_vals = zeros(1,2);                   %Vector of r2 values
        E_comp_vals = zeros(4,2);               %Vector of energy components

        %If we ignore the line tension term the corresponding variables are
        %set to 0
        varlist([3,4]) = varlist([3,4])*line_tension_term;

        while true
            %Energy and shape of the system is calculated assuming that the carbon
            %cap is shaped like a fullerene which has a total number of carbon
            %atoms equal to n_atoms. We use an exponentially increasing
            %step size to search a range of values for local minima.
            [energy(loop_counter),h1_vals(loop_counter), h2_vals(loop_counter), r1_vals(loop_counter), r2_vals(
     loop_counter), E_comp_vals(:, loop_counter)] = cap_energy(varlist, n_atoms, cap_energy_term);
            if loop_counter > 2
                %If a local energy minimum is discovered we perform a
                %golden search to determine the energy at this minimum
                if energy(loop_counter-2) ~= Inf && energy(loop_counter-2) > energy(loop_counter-1) && energy(
     loop_counter) > energy(loop_counter-1)
                    %If we have already found a local minimum for our
                    %current set of parameters we use temp variables to
                    %store data
                    if loop_flag == 1
                        [temp_E, temp_h1, temp_h2, temp_r1, temp_r2, temp_size, temp_E_comp] = golden_search(
     varlist, n_atoms_vals(end-2), n_atoms_vals(end), cap_energy_term);
                        %If the energy of this minimum is less than the
                        %lowest energy minimum found so far, we update the
                        %matrices. Otherwise we keep the other minimum's
                        %values
                        if temp_E < E(i,j)
                            E(i,j) = temp_E;
                            E_comp{i,j} = temp_E_comp;
                            E_ma(i,j) = E_comp{i,j}(1);
```

```
                        E_lt(i,j) = E_comp{i,j}(3);
                        E_c(i,j) = E_comp{i,j}(4);
                        h1(i,j) = temp_h1;
                        h2(i,j) = temp_h2;
                        r1(i,j) = temp_r1;
                        r2(i,j) = temp_r2;
                        fullerene_size(i,j) = temp_size;
                    end
                %If we haven't found a local minimum yet then we update
                %the matrices to the values at the minimum using a
                %golden search.
                else
                    [E(i,j), h1(i,j), h2(i,j), r1(i,j), r2(i,j), fullerene_size(i,j), E_comp{i,j}] = golden_search(varlist, n_atoms_vals(end-2), n_atoms_vals(end), cap_energy_term);
                    E_ma(i,j) = E_comp{i,j}(1);
                    E_lt(i,j) = E_comp{i,j}(3);
                    E_c(i,j) = E_comp{i,j}(4);
                    loop_flag = 1;
                end
            end
        end
        %When we reach the end of the search range we check for global
        %minima at the endpoints
        if loop_counter == n2
            index = find(energy ~= Inf, 1);

            if index ~= 1
                [temp_E, temp_h1, temp_h2, temp_r1, temp_r2, temp_size, temp_E_comp] = golden_search(varlist, n_atoms_vals(index-1), n_atoms_vals(index), cap_energy_term);
            else
                temp_E = energy(1);
                temp_r1 = r1_vals(1);
                temp_r2 = r2_vals(1);
                temp_h1 = h1_vals(1);
                temp_h2 = h2_vals(1);
                temp_size = n_atoms_vals(1);
                temp_E_comp = E_comp_vals(:,1);
            end
            if temp_E < E(i,j) || loop_flag == 0
                E(i,j) = temp_E;
                E_comp{i,j} = temp_E_comp;
                E_ma(i,j) = E_comp{i,j}(1);
                E_lt(i,j) = E_comp{i,j}(3);
                E_c(i,j) = E_comp{i,j}(4);
                h1(i,j) = temp_h1;
                h2(i,j) = temp_h2;
                r1(i,j) = temp_r1;
                r2(i,j) = temp_r2;
                fullerene_size(i,j) = temp_size;
                loop_flag = 2;
            end

            [temp_E, temp_h1, temp_h2, temp_r1, temp_r2, temp_size, temp_E_comp] = golden_search(varlist, n_atoms_vals(end-1), n_atoms_vals(end), cap_energy_term);
            if temp_E < E(i,j)
                E(i,j) = temp_E;
                E_comp{i,j} = temp_E_comp;
                E_ma(i,j) = E_comp{i,j}(1);
```

```
                    E_lt(i,j) = E_comp{i,j}(3);
                    E_c(i,j) = E_comp{i,j}(4);
                    h1(i,j) = temp_h1;
                    h2(i,j) = temp_h2;
                    r1(i,j) = temp_r1;
                    r2(i,j) = temp_r2;
                    fullerene_size(i,j) = temp_size;
                    loop_flag = 3;
                end

                break
            else
                %n_atoms is increased by exponentially increasing
                %increments each iteration
                n_atoms = n_atoms + step_initial*step_scale^(loop_counter-1);
                loop_counter = loop_counter + 1;
                n_atoms_vals(loop_counter) = n_atoms;
            end
        end
        %The contact angle is calculated according to the dimensions of the
        %system
        sin_theta = -sqrt(varlist(1)/pi - h1(i,j)^2)*(h1(i,j) - r1(i,j) + h2(i,j) - r2(i,j))/(r1(i,j)*r2(i,j));
        cos_theta = ((varlist(1)/pi - h1(i,j)^2) - (r1(i,j) - h1(i,j))*(r2(i,j)-h2(i,j)))/(r1(i,j)*r2(i,j));
        if cos_theta > 0
            contact_angle(i,j) = real(asin(sin_theta));
        elseif cos_theta < 0 && sin_theta > 0
            contact_angle(i,j) = pi - real(asin(sin_theta));
        else
            contact_angle(i,j) = -pi - real(asin(sin_theta));
        end
        flag_mat(i,j) = loop_flag;              %The matrix of flag values is updated

        loop_counter = 1;                       %Tracks the progress of the while loop
        n_atoms = n_atoms_vals(1);              %We reuse the value from the previous section
        n_atoms_vals = zeros(1,2);              %Vector of n_atoms values
        n_atoms_vals(1) = n_atoms;
        loop_flag = 0;                          %Becomes 1 if a minimum is found
        energy = zeros(1,2);                    %Vector of energy values
        h1_vals = zeros(1,2);                   %Vector of h1 values
        h2_vals = zeros(1,2);                   %Vector of h2 values
        r1_vals = zeros(1,2);                   %Vector of r1 values
        r2_vals = zeros(1,2);                   %Vector of r2 values
        E_comp_vals = zeros(4,2);               %Vector of energy components

        while true
            %Except where commented, this is the same process as in the
            %previous while loop.
            [energy(loop_counter), r1_vals(loop_counter), h2_vals(loop_counter), E_comp_vals(:, loop_counter)]
     = dewetted_energy(varlist, n_atoms, cap_energy_term);
            if loop_counter > 2
                %If the value of n_atoms becomes too large we fail to
                %obtain and energy and Inf is output instead. When this
                %occurs we skip to the end of the loop. Otherwise the loop
                %continues as in the previous loop.
                if energy(loop_counter) == Inf && energy(loop_counter - 1) ~= Inf
                    loop_counter = n2;
                elseif energy(loop_counter-2) ~= Inf && energy(loop_counter-2) > energy(loop_counter-1) &&
     energy(loop_counter) > energy(loop_counter-1)
```

```
                if loop_flag == 1
                    [temp_E, temp_r1, temp_h2, temp_size, temp_E_comp] = golden_search_dewetted(varlist, n_atoms_vals(end-2), n_atoms_vals(end), cap_energy_term);
                    if temp_E < E_dewetted(i,j)
                        E_dewetted(i,j) = temp_E;
                        E_comp_dewetted{i,j} = temp_E_comp;
                        E_ma_dewetted(i,j) = E_comp_dewetted{i,j}(1);
                        E_lt_dewetted(i,j) = E_comp_dewetted{i,j}(3);
                        E_c_dewetted(i,j) = E_comp_dewetted{i,j}(4);
                        r1_dewetted(i,j) = temp_r1;
                        h1_dewetted(i,j) = varlist(1)/(2*pi*r1_dewetted(i,j));
                        h2_dewetted(i,j) = temp_h2;
                        fullerene_size_dewetted(i,j) = temp_size;
                    end
                else
                    [E_dewetted(i,j), r1_dewetted(i,j), h2_dewetted(i,j), fullerene_size_dewetted(i,j), E_comp_dewetted{i,j}] = golden_search_dewetted(varlist, n_atoms_vals(end-2), n_atoms_vals(end), cap_energy_term);
                    E_ma_dewetted(i,j) = E_comp_dewetted{i,j}(1);
                    E_lt_dewetted(i,j) = E_comp_dewetted{i,j}(3);
                    E_c_dewetted(i,j) = E_comp_dewetted{i,j}(4);
                    h1_dewetted(i,j) = varlist(1)/(2*pi*r1_dewetted(i,j));
                    loop_flag = 1;
                end
            end
        end
        if loop_counter == n2

            index = find(energy ~= Inf, 1);

            if index ~= 1
                [temp_E, temp_r1, temp_h2, temp_size, temp_E_comp] = golden_search_dewetted(varlist, n_atoms_vals(index-1), n_atoms_vals(index), cap_energy_term);
            else
                temp_E = energy(1);
                temp_r1 = r1_vals(1);
                temp_h2 = h2_vals(1);
                temp_size = n_atoms_vals(1);
                temp_E_comp = E_comp_vals(:,1);
            end
            if temp_E <= E_dewetted(i,j) || loop_flag == 0
                E_dewetted(i,j) = temp_E;
                E_comp_dewetted{i,j} = temp_E_comp;
                E_ma_dewetted(i,j) = E_comp_dewetted{i,j}(1);
                E_lt_dewetted(i,j) = E_comp_dewetted{i,j}(3);
                E_c_dewetted(i,j) = E_comp_dewetted{i,j}(4);
                r1_dewetted(i,j) = temp_r1;
                h1_dewetted(i,j) = varlist(1)/(2*pi*r1_dewetted(i,j));
                h2_dewetted(i,j) = temp_h2;
                fullerene_size_dewetted(i,j) = temp_size;
                loop_flag = 2;
            end

            [temp_E, temp_r1, temp_h2, temp_size, temp_E_comp] = golden_search_dewetted(varlist, n_atoms_vals(end-1), n_atoms_vals(end), cap_energy_term);
            if temp_E < E_dewetted(i,j)
                E_dewetted(i,j) = temp_E;
                E_comp_dewetted{i,j} = temp_E_comp;
```

```
                    E_ma_dewetted(i,j) = E_comp_dewetted{i,j}(1);
                    E_lt_dewetted(i,j) = E_comp_dewetted{i,j}(3);
                    E_c_dewetted(i,j) = E_comp_dewetted{i,j}(4);
                    r1_dewetted(i,j) = temp_r1;
                    h1_dewetted(i,j) = varlist(1)/(2*pi*r1_dewetted(i,j));
                    h2_dewetted(i,j) = temp_h2;
                    fullerene_size_dewetted(i,j) = temp_size;
                    loop_flag = 3;
                end
                break
            else
                n_atoms = n_atoms + step_initial*step_scale^(loop_counter-1);              %step size increases each loop to reduce total number of steps when optimal number of atoms is large
                loop_counter = loop_counter + 1;
                n_atoms_vals(loop_counter) = n_atoms;
            end
        end

        sin_theta = -sqrt(varlist(1)/pi - h1_dewetted(i,j)^2)*(h1_dewetted(i,j) - r1_dewetted(i,j) + h2_dewetted(i,j) - r2_dewetted)/(r1_dewetted(i,j)*r2_dewetted);
        cos_theta = ((varlist(1)/pi - h1_dewetted(i,j)^2) - (r1_dewetted(i,j) - h1_dewetted(i,j))*(r2_dewetted-h2_dewetted(i,j)))/(r1_dewetted(i,j)*r2_dewetted);
        if cos_theta > 0
            contact_angle_dewetted(i,j) = real(asin(sin_theta));
        elseif cos_theta < 0 && sin_theta > 0
            contact_angle_dewetted(i,j) = pi - real(asin(sin_theta));
        else
            contact_angle_dewetted(i,j) = -pi - real(asin(sin_theta));
        end
        flag_mat_dewetted(i,j) = loop_flag;
    end
end

n_figs = 5;

%Figure of where dewetting occurs signified by a value of 1
surf(V1,V2, double((E - E_dewetted) > 0), 'edgecolor', 'none')
title('Dewetting Region')

%Figure of wetted state energy
figure(2)
surf(V1, V2, E, 'edgecolor', 'none')
hold on
contour3(V1,V2,E,'k','linewidth',2)
title('Wetted State Energy')

%Figure of dewetted state energy
figure(3)
surf(V1, V2, E_dewetted, 'edgecolor', 'none')
hold on
contour3(V1,V2,E_dewetted,'k','linewidth',2)
title('Dewetted State Energy')

%Figure of wetted state cap radius
figure(4)
surf(V1, V2, r1, 'edgecolor', 'none')
hold on
contour3(V1,V2,r1,'k','linewidth',2)
```

```
title('Wetted Cap Radius')

%Figure of dewetted state cap radius
figure(5)
surf(V1, V2, r1_dewetted, 'edgecolor', 'none')
hold on
contour3(V1,V2,r1_dewetted,'k','linewidth',2)
title('Dewetted Cap Radius')

%All the figures are updated to an appropriate font size and labels are
%applied
for i = 1:n_figs

    figure(i)
    ax = gca;
    ax.FontSize = 50;
    h = colorbar;
    %Dewetting plots have the colour range limited to between 0 and 1
    if i == 1
        caxis([0,1])
    elseif i > 3
        ylabel(h, ['r_1 (' char(197) ')'])
    else
        ylabel(h, ['E (eV)'])
    end
    grid off
    view(0,90)

    switch var1
        case 1
            xlabel(['A_g (' char(197) '^2)'])
        case 2
            xlabel(['\gamma_{ma} (eV/' char(197) '^2)'])
        case 3
            xlabel(['\gamma_{lt} (eV/' char(197) ')'])
        case 4
            xlabel('\gamma_{lt}*')
        case 5
            xlabel(['\gamma_{ga} (eV/' char(197) '^2)'])
        case 6
            xlabel(['\gamma_{gm} (eV/' char(197) '^2)'])
        case 7
            xlabel(['V_m (' char(197) '^3)'])
    end

    switch var2
        case 1
            ylabel(['A_g (' char(197) '^2)'])
        case 2
            ylabel(['\gamma_{ma} (eV/' char(197) '^2)'])
        case 3
            ylabel(['\gamma_{lt} (eV/' char(197) ')'])
        case 4
            ylabel('\gamma_{lt}*')
        case 5
            ylabel(['\gamma_{ga} (eV/' char(197) '^2)'])
        case 6
            ylabel(['\gamma_{gm} (eV/' char(197) '^2)'])
```

```
        case 7
            ylabel(['V_m (' char(197) '^3)'])
    end
end

function [E, h1, h2, r1, r2, E_comp] = cap_energy(varlist, atoms, cap_energy_term)
length_66 = 1.425;                      %Length of 6-6 bonds
if atoms >= 60
    bonds = [0, 60, atoms*3/2-60];              %[5-5 bonds, 5-6 bonds, 6-6 bonds]
elseif atoms >= 28
    bonds = [135/4 - 9*atoms/16, 9*atoms/8 - 15/2, 30*atoms/32 - 105/4];
else
    bonds = [60 - 3*atoms/2, 3*atoms - 60, 0];
end

%Total area of fullerene approximated by assuming each bond is a side
%shared by two isosceles triangles.
A = 0.75*atoms*sqrt(3)*length_66^2;
if A < varlist(1)
    E = Inf;                            %Return infinity if the cap area is larger than the total area of the chosen fullerene
    h1 = Inf;
    h2 = Inf;
    r1 = Inf;
    r2 = Inf;
    E_comp = [Inf; Inf; Inf; Inf];
else
    r1 = 0.5*sqrt(A/pi);                    %Radius of carbon cap
    h1 = varlist(1)/(2*pi*r1);                     %Height of carbon cap
    n_bonds = 1.5*atoms*varlist(1)/A;    %Number of bonds within region of area Ag
    p = 3*h1*(2*r1-h1);
    q = 2*(3*r1 - h1)*h1^2 - 6*varlist(7)/pi;
    C = (-q*0.5 + sqrt(0.25*q^2 + p^3/27))^(1/3);
    h2 = C - p/(3*C);
    r2 = varlist(7)/(pi*h2^2) - (3*varlist(1)*h1/(2*pi) - h1^3 - h2^3)/(3*h2^2);    %Radius of metal particle
    a = sqrt(varlist(1)/pi - h1^2);              %The radius of the interface between the metal particle and the carbon cap
    if real(a) == 0 && imag(a) ~= 0
        a = 0;
    end

    %The bonds in the carbon cap are assumed to be chosen randomly without
    %replacement
    N = atoms*3/2;                   %Total number of carbon-carbon bonds in fullerene
    mean_55 = n_bonds*bonds(1)/N;   %Mean number of 5-5 bonds in cap
    mean_56 = n_bonds*bonds(2)/N;   %Mean number of 5-6 bonds in cap (hypergeometric)
    mean_66 = n_bonds*bonds(3)/N;   %Mean number of 6-6 bonds in cap
    if cap_energy_term == 0
        E_55 = 0;
        E_56 = 0;
        E_66 = 0;
    else
        E_55 = 2.62;                     %Energy of 5-5 bonds
        E_56 = 4.54;                    %Energy of 5-6 bonds
        E_66 = 4.86;                    %Energy of 6-6 bonds
    end

    E_comp = [varlist(2)*(6*varlist(7) - 3*varlist(1)*h1 + 2*pi*(h1^3 + h2^3))/(3*h2); (varlist(6) + varlist(5)
```

```
     )*varlist(1); 2*pi*a*(varlist(3) - varlist(4)*(-a)*(r1 - h1 + r2 - h2)/(r1*r2)); -(E_55*mean_55 + E_56*
     mean_56 + E_66*mean_66)];
    %1 = metal-atmosphere interface, 2 = cap interfaces, 3 = line
    %tension, 4 = cap geometry
    E = E_comp(1)+E_comp(2)+E_comp(3)+E_comp(4);   %Total energy of the system

end
end

function [E, r1, h2, E_comp] = dewetted_energy(varlist, atoms, cap_energy_term)
length_66 = 1.425;                      %Length of 6-6 bonds
if atoms >= 60
    bonds = [0, 60, atoms*3/2-60];             %[5-5 bonds, 5-6 bonds, 6-6 bonds]
elseif atoms >= 28
    bonds = [135/4 - 9*atoms/16, 9*atoms/8 - 15/2, 30*atoms/32 - 105/4];
else
    bonds = [60 - 3*atoms/2, 3*atoms - 60, 0];
end

%Total area of fullerene approximated by assuming each bond is a side
%shared by two isosceles triangles.
A = 0.75*atoms*sqrt(3)*length_66^2;
if A < varlist(1)
    E = Inf;                            %Return infinity if the cap area is larger than the total area of the
     chosen fullerene
    h2 = Inf;
    r1 = Inf;
    E_comp = [Inf; Inf; Inf; Inf];
else
    r1 = sqrt(A/pi)/2;                   %Radius of carbon cap
    h1 = varlist(1)/(2*pi*r1);                    %Height of carbon cap
    n_bonds = 1.5*atoms*varlist(1)/A;    %Number of bonds within region of area Ag
    r2 = (3*varlist(7)/(4*pi))^(1/3);    %Radius of metal particle
    h2_plus = r2 + sqrt(r2^2 - 2*r1*h1 + h1^2);
    h2_minus = r2 - sqrt(r2^2 - 2*r1*h1 + h1^2);
    a = sqrt(varlist(1)/pi - h1^2);             %The radius of the interface between the metal particle and the
      carbon cap
    if ~isreal(a) && real(a) == 0 && isreal(h2_plus)
        a = 0;
    end

    if (r1 >= r2 && h1 < r1) || ~isreal(h2_plus)
        E = Inf;
        E_comp = [Inf; Inf; Inf; Inf];
        h2 = h2_plus;
    else
        %The bonds in the carbon cap are assumed to be chosen randomly without
        %replacement
        N = atoms*3/2;                  %Total number of carbon-carbon bonds in fullerene
        mean_55 = n_bonds*bonds(1)/N;   %Mean number of 5-5 bonds in cap
        mean_56 = n_bonds*bonds(2)/N;   %Mean number of 5-6 bonds in cap (hypergeometric)
        mean_66 = n_bonds*bonds(3)/N;   %Mean number of 6-6 bonds in cap
        if cap_energy_term == 0
            E_55 = 0;
            E_56 = 0;
            E_66 = 0;
        else
            E_55 = 2.62;                     %Energy of 5-5 bonds
```

```
            E_56 = 4.54;                    %Energy of 5-6 bonds
            E_66 = 4.86;                    %Energy of 6-6 bonds
        end

        E_comp = [4*pi*varlist(2)*r2^2; 2*varlist(5)*varlist(1); 2*pi*a*(varlist(3) - varlist(4)*(-a)*(r1 - h1 + r2 - h2_plus)/(r1*r2)); -(E_55*mean_55 + E_56*mean_56 + E_66*mean_66)];
        %1 = metal-atmosphere interface, 2 = cap interfaces, 3 = line
        %tension, 4 = cap geometry
        E = E_comp(1)+E_comp(2)+E_comp(3)+E_comp(4);   %Total energy of the system

        if r1 < r2 && h2_minus <= r2
            E_temp = Inf;
        else
            E_comp_temp = [4*pi*varlist(2)*r2^2; 2*varlist(5)*varlist(1); 2*pi*a*(varlist(3) - varlist(4)*(-a)*(r1 - h1 + r2 - h2_minus)/(r1*r2)); -(E_55*mean_55 + E_56*mean_56 + E_66*mean_66)];
            E_temp = E_comp_temp(1)+E_comp_temp(2)+E_comp_temp(3)+E_comp_temp(4);
        end

        if E_temp < E
            E_comp = E_comp_temp;
            E = E_temp;
            h2 = h2_minus;
        else
            h2 = h2_plus;
        end
    end
end
end

function [E, h1, h2, r1, r2, size, E_comp] = golden_search(varlist, min_val, max_val, cap_energy_term)
E_vals = zeros(1,4);
h1_vals = zeros(1,4);
h2_vals = zeros(1,4);
r1_vals = zeros(1,4);
r2_vals = zeros(1,4);
E_comp_vals = zeros(4,4);
golden_ratio = 0.5*(sqrt(5)+1);
inv_gr = 1/golden_ratio;
atoms = [min_val, max_val - inv_gr*(max_val-min_val), min_val + inv_gr*(max_val-min_val), max_val];
[E_vals(1),h1_vals(1), h2_vals(1), r1_vals(1), r2_vals(1), E_comp_vals(:,1)] = cap_energy(varlist, atoms(1), cap_energy_term);
[E_vals(2),h1_vals(2), h2_vals(2), r1_vals(2), r2_vals(2), E_comp_vals(:,2)] = cap_energy(varlist, atoms(2), cap_energy_term);
[E_vals(3),h1_vals(3), h2_vals(3), r1_vals(3), r2_vals(3), E_comp_vals(:,3)] = cap_energy(varlist, atoms(3), cap_energy_term);
[E_vals(4),h1_vals(4), h2_vals(4), r1_vals(4), r2_vals(4), E_comp_vals(:,4)] = cap_energy(varlist, atoms(4), cap_energy_term);
range = max_val-min_val;
while range > 0.001
    if (E_vals(1) > E_vals(2) && E_vals(3) >= E_vals(2)) || (E_vals(1) <= E_vals(2) && E_vals(2) <= E_vals(3) && E_vals(3) <= E_vals(4))
        atoms = [atoms(1), atoms(3) - inv_gr*(atoms(3) - atoms(1)), atoms(2), atoms(3)];
        E_vals = [E_vals(1), 0, E_vals(2), E_vals(3)];
        h1_vals = [h1_vals(1), 0, h1_vals(2), h1_vals(3)];
        h2_vals = [h2_vals(1), 0, h2_vals(2), h2_vals(3)];
        r1_vals = [r1_vals(1), 0, r1_vals(2), r1_vals(3)];
        r2_vals = [r2_vals(1), 0, r2_vals(2), r2_vals(3)];
        E_comp_vals = [E_comp_vals(:,1), zeros(4,1), E_comp_vals(:,2), E_comp_vals(:,3)];
```

```
        [E_vals(2),h1_vals(2), h2_vals(2), r1_vals(2), r2_vals(2), E_comp_vals(:,2)] = cap_energy(varlist, atoms(2), cap_energy_term);
    else
        atoms = [atoms(2), atoms(3), atoms(2) + inv_gr*(atoms(4)-atoms(2)), atoms(4)];
        E_vals = [E_vals(2), E_vals(3), 0, E_vals(4)];
        h1_vals = [h1_vals(2), h1_vals(3), 0, h1_vals(4)];
        h2_vals = [h2_vals(2), h2_vals(3), 0, h2_vals(4)];
        r1_vals = [r1_vals(2), r1_vals(3), 0, r1_vals(4)];
        r2_vals = [r2_vals(2), r2_vals(3), 0, r2_vals(4)];
        E_comp_vals = [E_comp_vals(:,2), E_comp_vals(:,3), zeros(4,1), E_comp_vals(:,4)];
        [E_vals(3),h1_vals(3), h2_vals(3), r1_vals(3), r2_vals(3), E_comp_vals(:,3)] = cap_energy(varlist, atoms(3), cap_energy_term);
    end
    range = atoms(4) - atoms(1);
end
E = min(E_vals);
index = find(E_vals == E);
if length(index) > 1
    index = index(1);
end
h1 = h1_vals(index);
h2 = h2_vals(index);
r1 = r1_vals(index);
r2 = r2_vals(index);
E_comp = E_comp_vals(:,index);
size = atoms(index);
end

function [E, r1, h2, size, E_comp] = golden_search_dewetted(varlist, min_val, max_val, cap_energy_term)
E_vals = zeros(1,4);
h2_vals = zeros(1,4);
r1_vals = zeros(1,4);
E_comp_vals = zeros(4,4);
golden_ratio = 0.5*(sqrt(5)+1);
inv_gr = 1/golden_ratio;
atoms = [min_val, max_val - inv_gr*(max_val-min_val), min_val + inv_gr*(max_val-min_val), max_val];
[E_vals(1), r1_vals(1), h2_vals(1), E_comp_vals(:,1)] = dewetted_energy(varlist, atoms(1), cap_energy_term);
[E_vals(2), r1_vals(2), h2_vals(2), E_comp_vals(:,2)] = dewetted_energy(varlist, atoms(2), cap_energy_term);
[E_vals(3), r1_vals(3), h2_vals(3), E_comp_vals(:,3)] = dewetted_energy(varlist, atoms(3), cap_energy_term);
[E_vals(4), r1_vals(4), h2_vals(4), E_comp_vals(:,4)] = dewetted_energy(varlist, atoms(4), cap_energy_term);
range = max_val-min_val;
while range > 0.001
    if (E_vals(2) > E_vals(3) && E_vals(4) >= E_vals(3)) || (E_vals(1) >= E_vals(2) && E_vals(2) >= E_vals(3) && E_vals(3) >= E_vals(4))
        atoms = [atoms(2), atoms(3), atoms(2) + inv_gr*(atoms(4)-atoms(2)), atoms(4)];
        E_vals = [E_vals(2), E_vals(3), 0, E_vals(4)];
        h2_vals = [h2_vals(2), h2_vals(3), 0, h2_vals(4)];
        r1_vals = [r1_vals(2), r1_vals(3), 0, r1_vals(4)];
        E_comp_vals = [E_comp_vals(:,2), E_comp_vals(:,3), zeros(4,1), E_comp_vals(:,4)];
        [E_vals(3), r1_vals(3), h2_vals(3), E_comp_vals(:,3)] = dewetted_energy(varlist, atoms(3), cap_energy_term);
    else
        atoms = [atoms(1), atoms(3) - inv_gr*(atoms(3) - atoms(1)), atoms(2), atoms(3)];
        E_vals = [E_vals(1), 0, E_vals(2), E_vals(3)];
        h2_vals = [h2_vals(1), 0, h2_vals(2), h2_vals(3)];
        r1_vals = [r1_vals(1), 0, r1_vals(2), r1_vals(3)];
        E_comp_vals = [E_comp_vals(:,1), zeros(4,1), E_comp_vals(:,2), E_comp_vals(:,3)];
        [E_vals(2), r1_vals(2), h2_vals(2), E_comp_vals(:,2)] = dewetted_energy(varlist, atoms(2),
```

```
         cap_energy_term);
    end
    range = atoms(4) - atoms(1);
end
E = min(E_vals);
index = find(E_vals == E);
if length(index) > 1
    index = index(1);
end
h2 = h2_vals(index);
r1 = r1_vals(index);
E_comp = E_comp_vals(:,index);
size = atoms(index);
end

function n_atoms = starting_point(Ag)
length_66 = 1.425;                       %Length of 6-6 bonds
n_atoms = 4*Ag/(3*sqrt(3)*length_66^2)+0.00000001;
if n_atoms < 20
    n_atoms = 20;
end
end
```

Appendix B

Bond type analysis script

```
clear
close all

sizes = 'all';                          %Size of fullerenes to be analysed
if strcmp(sizes, 'all')
    dinfo = dir('*.xyz');
else
    dinfo = dir(['*C', sizes, '*']);
end

nearby_faces = 1;
if nearby_faces
    length_55_0_hex = [];
    length_55_1_hex = [];
    length_55_2_hex = [];
    length_56_0_hex = [];
    length_56_1_hex = [];
    length_56_2_hex = [];
    length_66_0_hex = [];
    length_66_1_hex = [];
    length_66_2_hex = [];
end


nfiles = length(dinfo);                 %Number of files being read
atoms = zeros(1,nfiles);                %Vector to store the number of atoms in each fullerene
bond_numbers = zeros(3, nfiles);        %[5-5, 5-6, 6-6]
bond_lengths_55 = [];
bond_lengths_56 = [];
bond_lengths_66 = [];
for i = 1:nfiles
    filename = dinfo(i).name;
    fID = fopen(filename, 'r');                                 %A file is opened
    n_atoms = textscan(fID, ' %u', 1);                          %First line contains the number of atoms
    atoms(i) = n_atoms{1};
    A = textscan(fID, '%s %f %f %f %u %u %u %u', 'HeaderLines', 2); %Lines 3 onward contain position and bond
      data
    bonds = [A{5}, A{6}; A{5}, A{7}; A{5}, A{8}];               %List of bonds is generated
```

```
    L = zeros(3*atoms(i)/2);                                        %Line graph of fullerene is initialised
    for j = 1:3*atoms(i)/2                                          %This for loop removes duplicate edges
        index1 = find(bonds(:,1) == bonds(j,2));
        index2 = index1(bonds(index1,2) == bonds(j,1));
        bonds(index2,:) = [];
    end

    bond_lengths = sqrt((A{2}(bonds(:,1)) - A{2}(bonds(:,2))).^2 + (A{3}(bonds(:,1)) - A{3}(bonds(:,2))).^2 + (A{4}(bonds(:,1)) - A{4}(bonds(:,2))).^2);

    for j = 1:3*atoms(i)/2                                          %This for loop adds edges to line graph
        indices = [find(bonds(:,1) == bonds(j,1)); find(bonds(:,2) == bonds(j,1)); find(bonds(:,1) == bonds(j,2)); find(bonds(:,2) == bonds(j,2))];
        indices = indices(indices ~= j);
        for k = 1:length(indices)
            L(j, indices(k)) = 1;
            L(indices(k), j) = 1;
        end
    end

    for j = 1:3*atoms(i)/2                                          %Number of pentagons which share each edge is calculated using a depth first search
        n_pentagons = DFS2([j], L, 0, 5);
        n_pentagons = n_pentagons/2;
        if nearby_faces
            n_hexagons = DFS2([j], L, 0, 6)/2 - 5*n_pentagons;
        end
        if n_pentagons == 0
            bond_numbers(3, i) = bond_numbers(3, i)+1;
            bond_lengths_66(end+1) = bond_lengths(j);
            if nearby_faces
                if n_hexagons == 4
                    length_66_0_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 3
                    length_66_1_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 2
                    length_66_2_hex(end+1) = bond_lengths(j);
                else
                    n_hexagons
                end
            end
        elseif n_pentagons == 1
            bond_numbers(2, i) = bond_numbers(2, i)+1;
            bond_lengths_56(end+1) = bond_lengths(j);
            if nearby_faces
                if n_hexagons == 3
                    length_56_0_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 2
                    length_56_1_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 1
                    length_56_2_hex(end+1) = bond_lengths(j);
                else
                    n_hexagons
                end
            end
        elseif n_pentagons == 2
            bond_numbers(1, i) = bond_numbers(1, i)+1;
            bond_lengths_55(end+1) = bond_lengths(j);
```

```
            if nearby_faces
                if n_hexagons == 2
                    length_55_0_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 1
                    length_55_1_hex(end+1) = bond_lengths(j);
                elseif n_hexagons == 0
                    length_55_2_hex(end+1) = bond_lengths(j);
                else
                    n_hexagons
                end
            end
        end
    end

    fclose(fID);
end

fullerene_sizes = unique(atoms);
bond_averages = zeros(3, length(fullerene_sizes));
bond_length_averages = [mean(bond_lengths_55); mean(bond_lengths_56); mean(bond_lengths_66)];
for i = 1:length(fullerene_sizes)
    bond_averages(:, i) = mean(bond_numbers(:,find(atoms == fullerene_sizes(i))), 2);
end

if strcmp(sizes, 'all')
    scatter(atoms, bond_numbers(1,:), 100)
    hold on
    plot(fullerene_sizes, bond_averages(1,:), 'LineWidth', 3)
    hold off
    ax = gca;
    ax.FontSize = 50;
    title('5-5 Bonds')
    xlabel('Atoms in Fullerene')
    ylabel('Number of 5-5 Bonds')
    axis([min(fullerene_sizes), max(fullerene_sizes), 0, max(bond_averages(1,:))])

    figure(2)

    scatter(atoms, bond_numbers(2,:), 100)
    hold on
    plot(fullerene_sizes, bond_averages(2,:), 'LineWidth', 3)
    hold off
    ax = gca;
    ax.FontSize = 50;
    title('5-6 Bonds')
    xlabel('Atoms in Fullerene')
    ylabel('Number of 5-6 Bonds')
    axis([min(fullerene_sizes), max(fullerene_sizes), 0, max(bond_averages(2,:))])

    figure(3)

    scatter(atoms, bond_numbers(3,:), 100)
    hold on
    plot(fullerene_sizes, bond_averages(3,:), 'LineWidth', 3)
    hold off
    ax = gca;
    ax.FontSize = 50;
    title('6-6 Bonds')
```

```
    xlabel(’Atoms in Fullerene’)
    ylabel(’Number of 6-6 Bonds’)
    axis([min(fullerene_sizes), max(fullerene_sizes), 0, max(bond_averages(3,:))])

else

    hist(bond_numbers(1,:), ceil(2*sqrt(length(bond_numbers(1,:)))))
    ax = gca;
    ax.FontSize = 50;
    title(’5-5 Bonds’)
    xlabel(’Number of 5-5 Bonds’)
    ylabel(’Frequency’)
    axis tight

    figure(2)

    hist(bond_numbers(2,:), ceil(2*sqrt(length(bond_numbers(2,:)))))
    ax = gca;
    ax.FontSize = 50;
    title(’5-6 Bonds’)
    xlabel(’Number of 5-6 Bonds’)
    ylabel(’Frequency’)
    axis tight

    figure(3)

    hist(bond_numbers(3,:), ceil(2*sqrt(length(bond_numbers(3,:)))))
    ax = gca;
    ax.FontSize = 50;
    title(’6-6 Bonds’)
    xlabel(’Number of 6-6 Bonds’)
    ylabel(’Frequency’)
    axis tight

end

n = 19;          %must be odd
m = (n-1)/2;

figure(4)

h = histogram(bond_lengths_55, ceil(2*sqrt(length(bond_lengths_55))));
x = h.BinEdges;
x = x(1:end-1);
y = h.Values;
temp_y = zeros(1,length(y));
y = [zeros(1,m), y, zeros(1,m)];
for i = 1:n
    temp_y = temp_y + y(i:end-n+i);
end
y = temp_y/n;
hist(bond_lengths_55, ceil(2*sqrt(length(bond_lengths_55))));
hold on
plot(x,y,’k’,’LineWidth’,5)
ax = gca;
ax.FontSize = 50;
title(’5-5 Bond Lengths’)
xlabel([’Length (’ char(197) ’)’])
```

```
ylabel('Frequency')
axis tight
hold off

figure(5)

h = histogram(bond_lengths_56, ceil(2*sqrt(length(bond_lengths_56))));
x = h.BinEdges;
x = x(1:end-1);
y = h.Values;
temp_y = zeros(1,length(y));
y = [zeros(1,m), y, zeros(1,m)];
for i = 1:n
    temp_y = temp_y + y(i:end-n+i);
end
y = temp_y/n;
hist(bond_lengths_56, ceil(2*sqrt(length(bond_lengths_56))))
hold on
plot(x,y,'k','LineWidth',5)
ax = gca;
ax.FontSize = 50;
title('5-6 Bond Lengths')
xlabel(['Length (' char(197) ')'])
ylabel('Frequency')
axis tight
hold off

figure(6)

h = histogram(bond_lengths_66, ceil(2*sqrt(length(bond_lengths_66))));
x = h.BinEdges;
x = x(1:end-1);
y = h.Values;
temp_y = zeros(1,length(y));
y = [zeros(1,m), y, zeros(1,m)];
for i = 1:n
    temp_y = temp_y + y(i:end-n+i);
end
y = temp_y/n;
hist(bond_lengths_66, ceil(2*sqrt(length(bond_lengths_66))))
hold on
plot(x,y,'k','LineWidth',5)
ax = gca;
ax.FontSize = 50;
title('6-6 Bond Lengths')
xlabel(['Length (' char(197) ')'])
ylabel('Frequency')
axis tight
hold off

save(['Fullerene Bond Type Analysis ', sizes], 'atoms', 'bond_numbers', 'fullerene_sizes', 'bond_averages', '
     bond_lengths_55', 'bond_lengths_56', 'bond_lengths_66', 'bond_length_averages')

function n_pentagons = DFS2(explored, A, n_pentagons, path_length)
    if length(explored) == path_length
        if A(explored(end), explored(1))
            n_pentagons = n_pentagons + 1;
        end
```

```
    else
        nodes = find(A(explored(end),:) == 1);
        for i = nodes
            if ~any(explored == i)
                n_pentagons = DFS2([explored, i], A, n_pentagons, path_length);
            end
        end
    end
end
```

Appendix C

Script for analysing results

```
function [values, chiralities, percent_metallic] = dewetting_chirality(E,E_dewetted,V1,V2,h1_dewetted)
%Initialise the matrix of possible chiralities
max_nm = 100;   %Largest nanotube by radius is of chirality (max_nm, max_nm)
%c is the matrix of chiralities
[N,M] = meshgrid(0:max_nm, 0:max_nm);
c = cat(2,N’,M’);
c = reshape(c,[],2);
c = unique(sort(c,2),’rows’);
c(c(:,1)==c(:,2),:) = [];
c = [c(:,1), c(:,2); (1:max_nm)’, (1:max_nm)’];
%radius in angstroms is calculated for each chirality and added to the matrix
radii = (sqrt(3)*1.42/(2*pi))*sqrt(c(:,1).^2 + c(:,1).*c(:,2) + c(:,2).^2);
c = [radii, c];
c = sortrows(c,1);
%the chiral angle in radians is also calculated and added to c
c_angle = atan(sqrt(3).*c(:,2)./(2*c(:,3) + c(:,2)));
c = [c, c_angle];
%smallest possible nanotube is (2,2), smaller nanotubes are discarded
c = c(6:end,:);
%dewetting_matrix has a value of 1 at points where the dewetted state is
%energetically favourable
dewetting_matrix = double(E>E_dewetted);
n = length(dewetting_matrix);
%each column in values corresponds to a different vector of values
%1:   particle volume
%2-4: a, n, and m for minimum radius tube
%5-7: a, n, and m for maximum radius tube
%8-9: min and max chiral angles
%10:  chiral angle for earliest dewetting
%11:  chiral angle for latest dewetting
values = NaN(n,11);
%range of chiralities is stored in chiralities cell array
chiralities = cell(1,n);
for i = 1:n
    index = find(dewetting_matrix(:,i) == 1,1);
    index_end = find(dewetting_matrix(:,i) == 1,1,’last’);
    if ˜isempty(index) && ˜isnan(V2(index,i))
        values(i,1) = V2(index,i);
```

```
        r_range = V1(index:index_end,i)./(2*pi*h1_dewetted(index:index_end,i));
        a_range = sqrt(2*r_range.*h1_dewetted(index:index_end,i) - h1_dewetted(index:index_end,i).^2);
%values of a corresponding to nanotubes smaller than (3,3) are discarded
        a_range = a_range(a_range > 1.71);
        a_min = min(a_range);
        if ~isempty(a_range)
            values(i,2) = a_min;
            a_max = max(a_range);
            values(i,5) = a_max;
            index2 = find(a_min < c(:,1),1);
            index2_end = find(a_max < c(:,1),1);
            values(i,3:4) = c(index2, 2:3);
            values(i,6:7) = c(index2_end, 2:3);
            if index2 < index2_end
                values(i,8) = min(c(index2:index2_end, 4));
                values(i,9) = max(c(index2:index2_end, 4));
                chiralities{i} = c(index2:index2_end, 2:3);
            else
                values(i,8) = min(c(index2_end:index2, 4));
                values(i,9) = max(c(index2_end:index2, 4));
                chiralities{i} = c(index2_end:index2, 2:3);
            end
            values(i,10) = c(index2, 4);
            values(i,11) = c(index2_end, 4);
        end
    end
end
%The percentage of metallic nanotubes for each particle volume is calculated here
percent_metallic = zeros(n,1);
for i = 1:n
    if ~isempty(chiralities{i})
        diff = chiralities{i}(:,2) - chiralities{i}(:,1);
        percent_metallic(i) = sum(mod(diff,3)==0)/length(diff);
    else
        percent_metallic(i) = NaN;
    end
end

%Plot for range of a values
plot(values(:,1),values(:,2),'LineWidth',2)
hold on
plot(values(:,1),values(:,5),'LineWidth',2)
title('Range of Graphene Cap Edge Radii')
xlabel(['V_m (' char(197) '^3)'])
ylabel(['a (' char(197) ')'])
legend('min','max','Location','southeast')
ax = gca;
ax.FontSize = 50;
hold off
%Plot for percentage of metallic nanotubes
figure(2)
plot(values(:,1),100*percent_metallic,'LineWidth',2)
title('Proportion of potential CNTs with Metallic Properties')
xlabel(['V_m (' char(197) '^3)'])
ylabel('% Metallic CNTs')
ax = gca;
ax.FontSize = 50;
end
```