Department of Mathematics and Applied Mathematics

Dynamical Behavior of Graphene Models

Arnold Ngapasare

NGPARN001

Dissertation presented for the degree of Master of Science in the Department of Mathematics and Applied Mathematics

University of Cape Town

September 2017

Supervisor: Prof. Ch. Skokos

Plagiarism declaration

I know the meaning of plagiarism and declare that all of the work in the dissertation, save for that which is properly acknowledged, is my own.

Arnold Ngapasare

Signature

Date

Contents

Abstract

Since the discovery of a method to obtain graphene in 2004, there has been intensive research on this material by several researchers ranging from investigations of its physical and chemical properties to some novel applications of graphene. The discovery of graphene came about despite the fact that some leading researchers such as Landau and Peierls had predicted that two-dimensional (2D) crystals were thermodynamically unstable. Andre Geim and Konstantin Novoselov managed to obtain graphene using a rather surprising technique called the scotch tape method. The scotch tape method involves carefully peeling off layer after layer in graphite which is a three-dimensional (3D) material without making any distortions to the subsequent layers.

Due to the many applications of graphene, understanding the dynamical behavior of the material is a very important problem. In order to investigate the chaoticity in graphene, empirical force fields appropriate for modeling its dynamics are required. In this study we use such models which have been established to accurately describe bond stretching and bond deformation in graphene. Based on the corresponding Hamiltonian formalism we derive the system's Hamilton equations of motion, whose numerical solution determine the dynamical behavior of graphene, as well as the so-called variational equations needed for the numerical computation of several chaos indicators like the maximum Lyapunov Characteristic Exponent.

To my parents

List of Acronyms

CPU: Central Processing Unit
DFT: Density Functional Theory
GCC: Gnu Compiler Collection
mLCE: maximum Lyapunov Characteristic Exponent
PSS: Poincaré Surface of Section
SI : Symplectic Integrator

Acknowledgements

It is with a deep sense of gratitude and pleasure that I thank those who made this thesis possible. I wish to thank, first and foremost, my supervisor A/Prof. Ch. Skokos and Dr. G. Kalosakas for giving me the opportunity to work on this topic. I would like to express my deepest gratitude for their encouragement, guidance and support. In particular, I would like to thank Dr. Kalosakas for making available to me his code for graphene's evolution, as this greatly helped me to develop my codes for this system. I also take this opportunity to thank all members of the 'Nonlinear Dynamics and Chaos' group at the University of Cape Town for the fruitful conversations and ideas. I would also like to acknowledge the technical support, as well as the computational resources provided by the Centre for High Performance Computing in South Africa. To Kevin Colville, I say thank you so much for the useful discussions. Finally, I would like to thank my beloved parents for their financial and moral support and Patience for her patience and support.

Chapter 1

Hamiltonian Systems

1.1 Dynamical Systems

A dynamical system can be described by a set of differential equations or difference equations governing the evolution of some variables, the values of which determine the present state of the system. If a dynamical system has say k state variables its dimension is $k \in \mathbb{N}$. The k state variables can be represented by a vector $\mathbf{x} = (x_1\, x_2\, x_3 \ldots x_k)$ or a transposed matrix $\mathbf{x} = [x_1\, x_2\, x_3 \ldots x_k]^T$. The space defined by $\mathbf{x}$ is called the system's phase space. For a phase space of k dimensions a state $\mathbf{x}$ gives a particular point in that space. The set of all the parameterized points $\mathbf{x}(t)$, where t is the discrete or continuous time, are termed an orbit of the dynamical system. There are two types of dynamical systems termed continuous and discrete systems. A continuous dynamical system is described by a set of continuous time differential equations of the form

$$\dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \vec{f}(\vec{x}). \tag{1.1}$$

Difference equations describe the so-called discrete dynamical systems or maps. In that case the dynamics of the system is defined by a set of equations of the form

$$x(t_{n+1}) = f(x(t_n)), \qquad n = 0, 1, 2, 3, \ldots \tag{1.2}$$

A Hamiltonian system is an example of a continuous dynamical system whose dynamics is determined by the so called Hamiltonian function, which is a function of the system's state variables. If the Hamiltonian does not explicitly depend on time t the system is called autonomous, while the explicit appearance of t in the functional form of the Hamiltonian makes the system non-autonomous.

1.2 The Hénon-Heiles model

In order to illustrate some basic concepts of Hamiltonian dynamics let us consider a particular prototypical example case. The Hénon-Heiles system is an autonomous Hamiltonian model which describes the motion of stars in a galaxy (Hénon and Heiles, 1964). This system was used to address the question of the existence of the third integral of galactic motion in the 1960s by Michel Hénon and Carl Heiles.

The dynamical evolution of Hamiltonian systems is governed by the Hamilton equations of motion

$$\begin{aligned} \dot{q}_i &= \frac{\partial H(q,p)}{\partial p_i}, \\ \dot{p}_i &= -\frac{\partial H(q,p)}{\partial q_i}, \end{aligned} \qquad i = 1, 2, \ldots, N, \tag{1.3}$$

where $H(q,p)$ is the system's Hamiltonian function with q denoting the set of generalized coordinates q_i and p the conjugate generalized momenta p_i. The number N of q or p coordinates defines the systems degrees of freedom. The Hamiltonian of the Hénon-Heiles system is

$$H(x, y, p_x, p_y) = \frac{p_x^2}{2} + \frac{p_y^2}{2} + \frac{1}{2}\left(x^2 + y^2 + 2x^2 y - \frac{2}{3}y^3\right), \tag{1.4}$$

where x, y are the cartesian coordinates of a star on the galaxy plane and p_x, p_y the corresponding conjugate momenta. The corresponding equations of motion are

$$\begin{aligned} \dot{x} &= p_x, \\ \dot{y} &= p_y, \\ \dot{p_x} &= -(x + 2xy), \\ \dot{p_y} &= -(y + x^2 - y^2). \end{aligned} \tag{1.5}$$

Since H in (1.4) does not depend explicitly on time t the Hénon-Heiles system is an autonomous one. In this case the value of the Hamiltonian function in (1.4) is an integral of motion, i.e. it remains constant during the time evolution of any orbit defined by some initial conditions $x(0)$,$y(0)$, $p_x(0)$ or $p_y(0)$. For this reason an autonomous Hamiltonian is said to be conservative.

1.3 Integration techniques

In general the solution of the equations of motion (1.3) can be found by using some numerical integration technique. Numerical integration methods for Hamiltonian systems can be divided into two basic categories: symplectic and non symplectic integrators.

Let us discuss in more detail the notion of symplectic integrators. The equations of motion (1.3) of an N degrees of freedom Hamiltonian system can be written as $\frac{d\vec{X}}{dt} = \{H, \vec{X}\} = D_H$ where $\vec{X} = (\vec{q}, \vec{p}) = (q_1, q_2, \ldots, q_N, p_1, p_2, \ldots, p_N)$ is the system's state vector, $\{\cdot, \cdot\}$ denotes the usual Poisson bracket i.e.

$$\{f, g\} = \sum_{i=1}^{N} \left\{ \frac{\partial f}{\partial q_i}\frac{\partial g}{\partial p_i} - \frac{\partial f}{\partial p_i}\frac{\partial g}{\partial q_i} \right\} \tag{1.6}$$

and D_H is the corresponding differential operator. Then the solution from $t = 0$ to time $t = \tau$ of the Hamilton equations of motion (1.3) is formally given by

$$\vec{X}(t) = \Big[exp(\tau D_H)\Big]\vec{X}(0). \tag{1.7}$$

Considering a Hamiltonian system which can be split into two integrable parts, like for example the kinetic energy $T(p)$ and the potential energy $V(q)$, the differential operator D_H can be written as $D_H = D_T + D_V$. Then (1.7) becomes

$$\vec{X}(\tau) = \Big[exp(\tau(D_T + D_V))\Big]\vec{X}(0). \tag{1.8}$$

By denoting $A = D_T$ and $B = D_V$ we can, in general, find appropriate real numbers c_i, d_i for $i = 1, 2, 3 \ldots, n$ satisfying the equality

$$exp(\tau(A+B)) = \prod_{i=1}^{n} exp(\tau c_i A) exp(\tau d_i B) + \mathcal{O}(\tau^{n+1}) \tag{1.9}$$

for some given integer n. We note that the operators $exp(\tau c_i A)$ and $exp(\tau d_i B)$ correspond to the propagation of an initial condition for times τc_i and τd_i respectively under the influence of the Hamiltonian functions $T(p)$ and $V(q)$ respectively. The mapping

$$\vec{X}(\tau) = \Big[\prod_{i=1}^{n} exp(\tau c_i A) exp(\tau d_i B)\Big] \vec{X}(0) \tag{1.10}$$

approximates the exact solution of (1.3) to an accuracy of order n $\mathcal{O}(\tau^n)$ and can be explicitly evaluated through the successive applications of the following mappings

$$q' = q + \tau c_i \frac{\partial T(p)}{\partial p} \tag{1.11}$$

and

$$p' = p - \tau d_i \frac{\partial V(q)}{\partial q}. \tag{1.12}$$

The mapping (1.10) yields an n^{th} order symplectic integrator (SI) which propagates $z = (q(t), p(t)) \equiv \vec{X}(t)$ to $z' = (q(t+\tau), p(t+\tau) \equiv \vec{X}(t+\tau)$. In order to apply SI schemes for the Hénon-Heiles system we split the Hamiltonian (1.4) in two parts

$$T(p) = T(p_x, p_y) = \frac{p_x^2}{2} + \frac{p_y^2}{2}, \qquad V(q) = V(x, y) = \left(x^2 + y^2 + 2x^2 y - \frac{2}{3} y^3\right), \tag{1.13}$$

so that $H(q,p) = T(p) + V(q)$. Then the operators (1.11) and (1.12) take the form

$$exp(\tau D_T) = \begin{cases} x' = x + p\tau \\ y' = y + p_y \tau \\ p_x' = p_x \\ p_y' = p_y \end{cases} \tag{1.14}$$

and

$$exp(\tau D_V) = \begin{cases} x' = x \\ y' = y \\ p_x' = p_x - (x + 2xy)\tau \\ p_y' = p_y - (y + x^2 - y^2)\tau. \end{cases} \tag{1.15}$$

A way of quantifying the efficiency of an integration scheme in the case of autonomous Hamiltonian systems is by checking the constancy of the system's integral of motion, i.e. the Hamiltonian itself throughout the evolution of the system's relative energy error. The relative energy error $\Delta H(t)$ at any given time t is computed as

$$\Delta H(t) = \frac{|H(t) - H(0)|}{H(0)}. \tag{1.16}$$

A salient feature of symplectic integrators is that the relative energy error remains bounded for conservative Hamiltonians, while for non symplectic schemes in general it increases as the integration

time grows. This behavior can be easily seen in Figure 1.1 where the evolution of $\Delta H(t)$ for the same orbit of the Hénon-Heiles system is plotted when the non-symplectic 4^{th} order Runge-Kutta integrator (Kutta, 1901) and the Forest and Ruth 4^{th} order symplectic scheme are used (Forest and Ruth, 1990). It is clearly seen that the error remains bounded when the SI is used. This property makes symplectic integration ideal for following the time evolution of Hamiltonian systems.

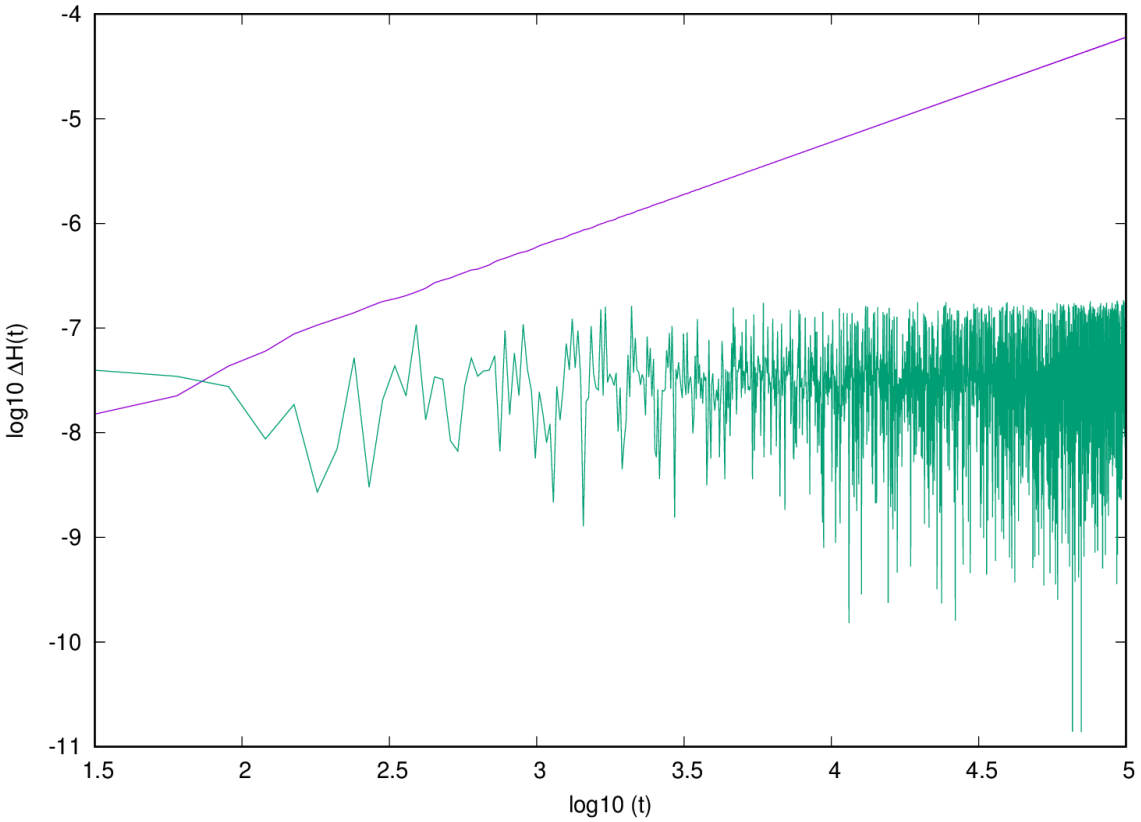

Figure 1.1: The time evolution of the relative energy error $\Delta H(t)$ (1.16) of the Hénon-Heiles system (1.4) for the orbit with initial conditions $x(0) = 0$, $y(0) = 0.3$, $p_x(0) = 0.30$, $p_y(0) = 0$. The non-symplectic Runge-Kutta 4^{th} order scheme (purple curve) with time step $\tau = 0.03$ and the Forest and Ruth 4^{th} order SI (green curve) with time step $\tau = 0.03$ were used.

1.4 Chaos

We present here the notion of chaos following the formal definition provided by Devaney (1989). According to that definition there are three properties which should be fulfilled in order for a dynamical system to be considered as chaotic. The properties are

1. the dynamical system must be sensitive to initial conditions,
2. the dynamical system must be topologically mixing, and
3. the dynamical system must have dense set of periodic orbits.

This definition has been shown to be redundant as the last two properties imply sensitivity to initial conditions.

1.5 Poincaré Surface of Section

The Poincaré surface of section (PSS) method provides a qualitative way of distinguishing between chaotic and regular regions. The PSS can be defined as a section cutting through the dynamical system's phase space. For an autonomous Hamiltonian system with two degrees of freedom, like the Hénon-Heiles model, the PSS can be a two-dimensional surface in the system's four-dimensional phase space, which separate the phase space in two regions, let's say A and B. Whenever an orbit intersects this surface following a particular direction, e.g. from A to B, the intersection point is marked on the PSS. In this way phase space portraits are created, which visualize the system's dynamics.

The PSS method has some drawbacks in that it can only be used for low dimensional Hamiltonian systems. For higher dimensional systems, quantitative chaos indicators such as the maximum Lyapunov Characteristic Exponent (mLCE) should be used. Hénon (1982) outlined a numerical algorithm for computing Poincaré maps and we use this method to get the PSS for the Hénon-Heiles system. There are some obvious advantages of using the PSS such as

- The number of the coordinates is reduced by at least one
- The amount of computed data is therefore decreased.
- The computational resources required are consequently reduced.

It is important to note that the PSS can be taken along any plane in the phase space but some choices are better. For example taking a plane in the phase space such that one state variable remains constant can simplify the analysis. The procedure for taking the PSS as given by Williams (1997) is as follows

1. Plot the raw data in phase space.
2. Decide on the orientation of the Poincaré section.
3. Get the parameters for the equation of the section.
4. Using the equation of the section, the coordinates of each datum point are inserted and the appropriate equation from 2 is solved. If the result is positive the datum point is on one side if negative the datum point is on the other side.
5. A decision on whether to take points entering from the positive side or negative side is made.
6. If the section is to represent the positive to negative direction then all pairs of points from step four with a $+-$ signs are found.
7. For each pair of such points we interpolate to get the coordinates where the trajectory intersects the Poincaré section. This is done using the method proposed by Hénon (1982).
8. Plot the Poincaré section as a separate figure.

From the PSS plot the regions populated by scattered points represent chaotic regions whilst the regions where smooth curves are formed represent regular regions.

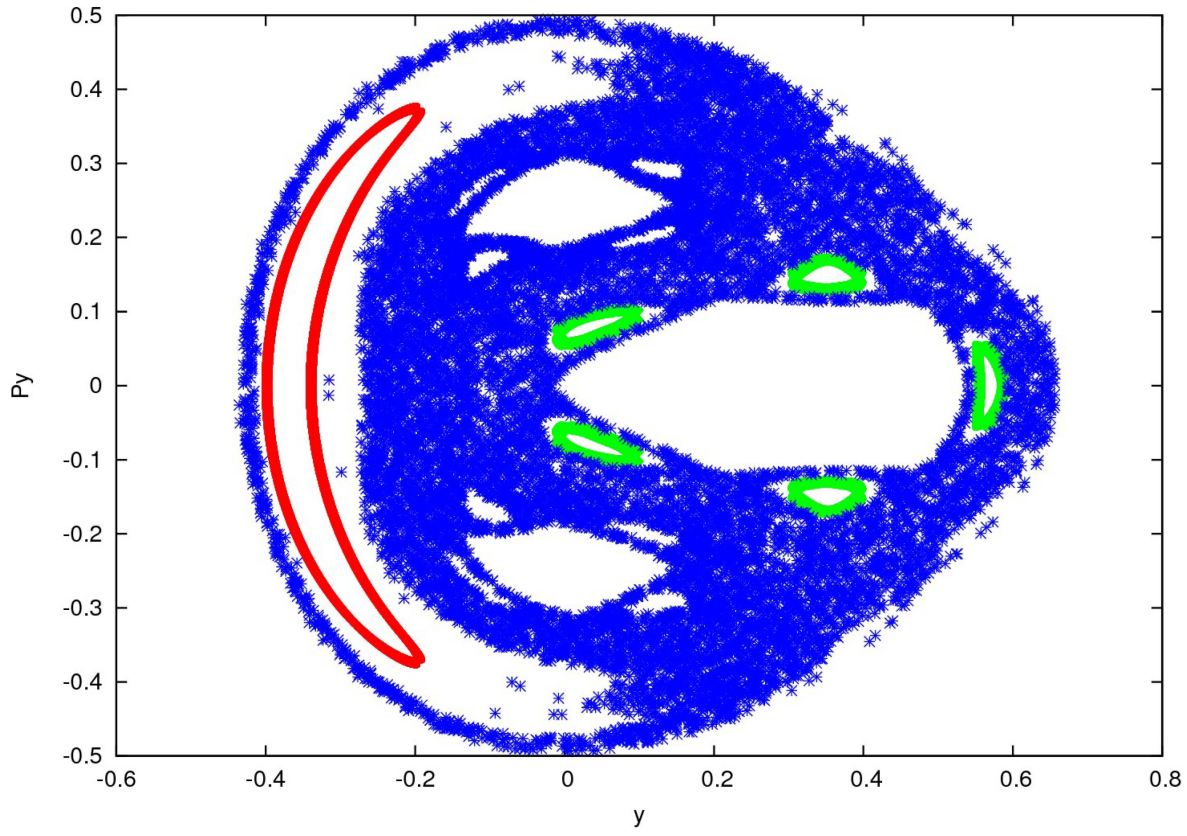

Figure 1.2: The PSS for a Hénon-Heiles Hamiltonian for orbits with initial conditions $x = 0$, $y = -0.24852$, $p_x = 0.3015641$, $p_y = 0.295064$ (regular orbit), $x = 0$, $y = -0.20962$, $p_x = 0.44371$, $p_y = 0.055140$ (chaotic orbit) and $x = 0$, $y = 0.555277$, $p_x = 0.23598$, $p_y = 0.0101094$ (regular orbit).

For our test case of the Hénon-Heiles Hamiltonian model (1.4), we plot in Figure 1.2 the corresponding PSS to show the islands of stability (smooth curves) and the chaotic seas (scattered points) for different initial conditions. For higher dimensional Hamiltonians, the PSS method may not be appropriate hence we need to determine chaoticity quantitatively.

1.6 Variational Equations and Tangent Map Method

One way of quantifying the chaoticity of a particular orbit is through the evolution of a small deviation vector about it. So for an orbit with initial conditions $\vec{X(0)}$ in the phase space we have to see how an infinitesimal deviation $\vec{\delta x(0)}$ evolves in time. The time evolution of $\vec{\delta x}$ is governed (at first order approximation) by the so-called variational equations having the form

$$\dot{\vec{w}} = [J_{2N} \cdot D_H^2(\vec{X}(t))] \cdot \vec{w}, \tag{1.17}$$

where $\vec{w} = (\delta q, \delta p)$ and

$$J_{2N} = \begin{bmatrix} 0_N & I_N \\ -I_N & 0_N \end{bmatrix}, \tag{1.18}$$

with I_N being the $N \times N$ identity matrix, 0_N the $N \times N$ matrix having its elements equal to zero and $D_H^2(\vec{X(t)})$ being the Hessian matrix of Hamiltonian $H(q,p)$ calculated on the reference orbit $\vec{X(t)}$ i.e.

$$D_H^2\left(\vec{X}(t)\right)_{(i,j)} = \left.\frac{\partial^2 H}{\partial x_i \partial x_j}\right|_{\vec{X}(t)}, \tag{1.19}$$

for $i,j = 1,2,3,\ldots N$. The variational equations of the Hénon-Heiles system are

$$\dot{\delta x} = \delta p_x, \qquad \dot{\delta y} = \delta p_y,$$
$$\dot{\delta p_x} = -\frac{\partial^2 H}{\partial x \partial x}\delta x - \frac{\partial^2 H}{\partial x \partial y}\delta y \Rightarrow \dot{\delta p_x} = (-1-2y)\delta x + (-2x)\delta y,$$
$$\dot{\delta p_y} = -\frac{\partial^2 H}{\partial y \partial y}\delta y - \frac{\partial^2 H}{\partial y \partial x}\delta x \Rightarrow \dot{\delta p_y} = (-1+2y)\delta y + (-2x)\delta x.$$

Several numerical integration schemes can be used to integrate the equations of motion simultaneously with the variational equations. In our study we consider the so-called "Tangent Map method" which is based on the use of SI (Gerlach and Skokos, 2011). This method has been proven to be a fast and accurate technique when compared to other numerical integration techniques both symplectic and non symplectic such as the Taylor series method (Gerlach and Skokos, 2011). Applying the tangent map method for the Hénon-Heiles system we consider the combined vector of the phase space variables x, y, p_x, p_y and the deviation vector coordinates $\delta x, \delta y, \delta p_x, \delta p_y$ at time t to be propagated at time $t+\tau$ to $(x', y', p_x', p_y', \delta x', \delta y', \delta p_x', \delta p_y')$ through the successive applications of the following extensions of operators (1.14) and (1.15)

$$exp(\tau D_{TS}) = \begin{cases} x' = x + p_x\tau \\ y' = y + p_y\tau \\ p_x' = p_x \\ p_y' = p_y \\ \delta x' = \delta x + \delta p_x\tau \\ \delta y' = \delta y + \delta p_y\tau \\ \delta p_x' = \delta p_x \\ \delta p_y' = \delta p_y \end{cases}, \tag{1.20}$$

$$exp(\tau D_{VS}) = \begin{cases} x' = x \\ y' = y \\ p_x' = p_x - (x+2xy)\tau \\ p_y = p_y - (y+x^2-y^2)\tau \\ \delta x' = \delta x \\ \delta y' = \delta y \\ \delta p_x' = \delta p_x + [(-1-2y)\delta x + (-2x)\delta y]\tau \\ \delta p_y' = \delta p_y + [(-1+2y)\delta y + (-2x)\delta x]\tau. \end{cases} \tag{1.21}$$

1.7 The maximum Lyapunov Characteristic Exponent

A main measure of chaoticity one can use is the mLCE. The theory of mLCE was developed by Oseledec (1968) who first used LCEs to characterize chaotic orbits, while later the numerical

algorithm of Benettin et al. (1980a,b) made possible the computation of the mLCE along with the whole set of the so-called spectrum of LCEs. We know that for a chaotic orbit of an N degrees of freedom Hamiltonian system the mLCE is positive, signifying the divergence of nearby orbits.

In order to evaluate the mLCE we start with a unitary deviation vector which is evolved according to the variational equations (1.17). In order to avoid overflow problems the evolved vector $\vec{\mathbf{w}}(\mathrm{t})$ is then replaced by a unitary vector $\hat{\mathbf{w}}(\mathrm{t})$ having the same direction with $\vec{\mathbf{w}}(\mathrm{t})$. This is done for successive steps up to the final integration time (see Figure 1.3).

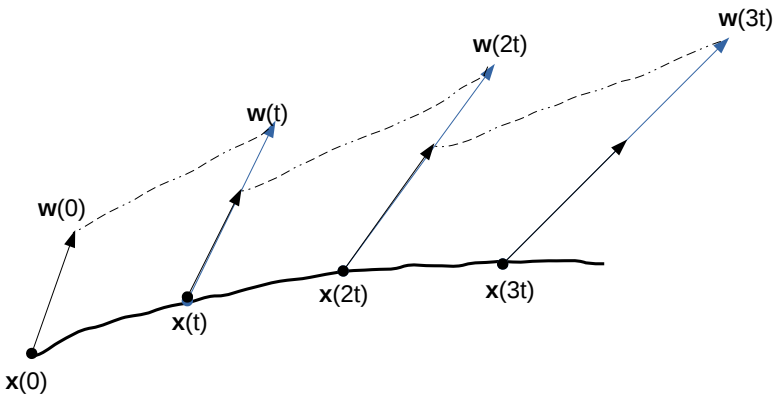

Figure 1.3: A schematic diagram on how to compute the mLCE where $\mathbf{x}$ is the vector for the phase space coordinates and $\mathbf{w}$ is the deviation vector.

Then the mLCE is evaluated as

$$\mathrm{mLCE} = \lim_{t\to\infty} \frac{1}{t} \ln \frac{\|\vec{\mathbf{w}}(t)\|}{\|\vec{\mathbf{w}}(0)\|}, \tag{1.22}$$

where $\|\cdot\|$ denotes the usual Euclidean norm of a vector.

In the case of chaotic orbits the mLCE is positive, whereas in the case of regular orbits the mLCE tends to zero following a power law of the form $\mathrm{mLCE} \propto t^{-1}$ (Skokos, 2010).

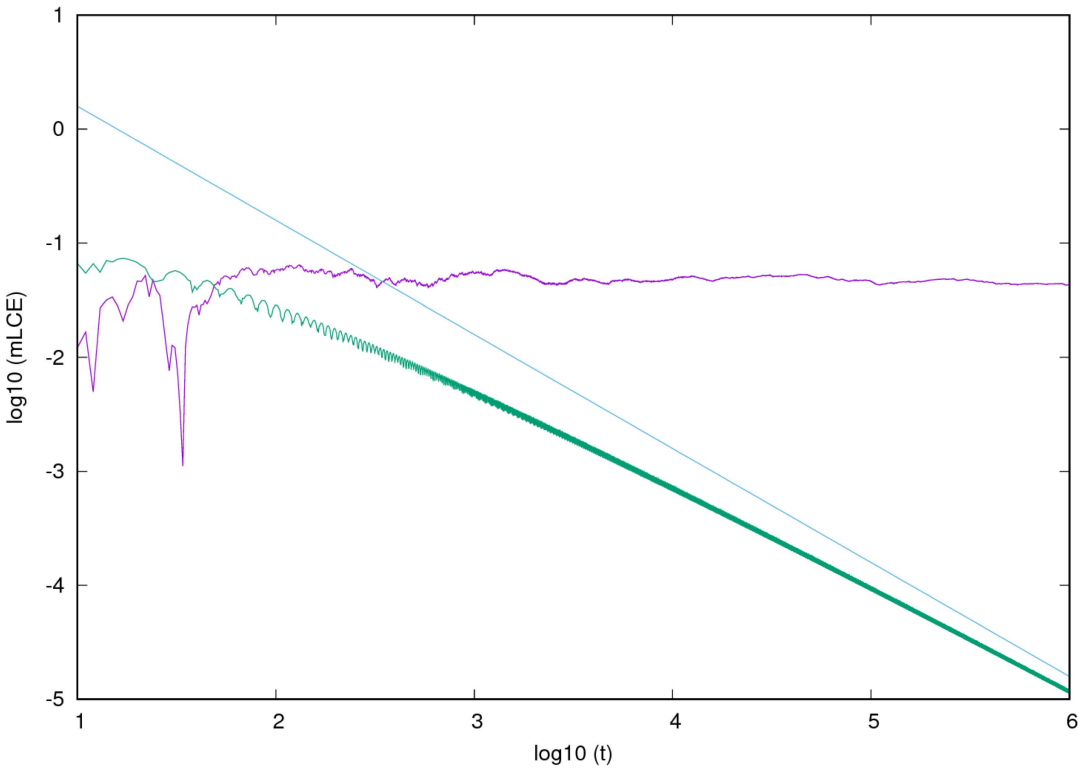

Figure 1.4: The time evolution of the mLCE for a regular (green curve) and a chaotic (purple curve) orbit of the Hénon-Heiles model. The initial conditions of the orbits are $x = 0$, $y = -0.24852$, $p_x = 0.3015641$, $p_y = 0.295064$ (regular orbit), and $x = 0$, $y = -0.25$, $p_x = 0.42081$, $p_y = 0$ (chaotic orbit). The blue straight line represents a function $\propto t^{-1}$.

In Figure 1.4 we show the time evolution of the estimated mLCE for a chaotic orbit (purple curve) and a regular orbit (green curve) of the Hénon-Heiles system.

Chapter 2

Dynamics of a Hamiltonian model of graphene

2.1 Graphene

Graphene is a two-dimensional (2D) hexagonal one atom thick carbon macromolecule which is derived from the three-dimensional (3D) carbon macromolecule graphite. Graphite macromolecules are made up of layer upon layer of 2D honeycomb carbon lattices interconnected by van der Waals' forces to give a 3D structure. When the 2D layers in graphite are "peeled off" one after another without affecting the top and bottom layers we end up with just a single 2D carbon lattice which is one atom thick and this is called graphene (Figure 2.1). Graphene is the building block for many other carbon macromolecules for example it can be rolled to give zero-dimensional (0D) buckballs, can be rolled into one-dimensional carbon nanotubes or stacked up as 3D graphite. 0D Buckyballs are one atom thick spherical carbon macromolecules with about 60 carbon atoms derived from graphene. When a flat layer of graphene is rolled to give a cylindrical macromolecule of carbon atoms the resultant macromolecule is called a 1D carbon nanotube.

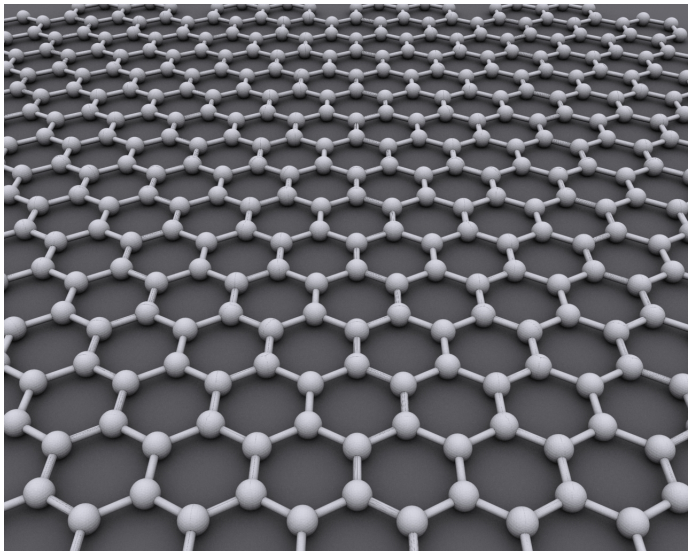

Figure 2.1: The ideal graphene crystalline structure. Source: Alexander Alus (2010).

The "road to graphene" was complicated as some leading researchers such as Landau (1937) had

predicted that 2D crystals were thermodynamically unstable, hence a good number of researchers thought graphene was just another wild goose chase or a pie in the sky. However, increasing numbers of scientists had started working on graphite with fewer and fewer layers and this gave hope that one day a single layer of graphite could be obtained and characterized. The breakthrough came in 2004 when Konstantin Novoselov and Andre Geim (Figure 2.2) at Manchester University managed to obtain single layer graphene using the so called scotch tape method (Novoselov et al., 2004). The scotch tape method entails that a cellophane tape is used to painstakingly remove layer after

Figure 2.2: The winners of the 2010 Nobel Prize for Physics Konstantin Novoselov (left) and Andre Geim (right). Photos: copyright ©The Nobel Foundation

layer from a graphite molecule until only a single layer of graphene remains. The major drawback of this method is that it can not produce graphene in quantities large enough to allow commercial production although it still yields graphene of the highest quality as compared to the rest of the techniques that were developed from 2005 onwards (Novoselov et al., 2005b, 2012). Since 2004, several methods have been developed to produce good quality graphene on a commercial scale and many other materials such as molybdenum disulphide, niobium diselenide, tungsten selenium etc (Novoselov et al., 2005a). It is worth noting that for their work on graphene, Konstantin Novoselov and Andre Geim were awarded the Nobel Prize in Physics in 2010, further highlighting the importance of their contribution.

Graphene has unique properties in that it can be used either as an oxidizing agent (electron accepting material) or a reducing agent (electron giving material). This rather unique property can be attributed to graphene's electron affinity as well as its ionizing potential of about $4.6eV$ (Allen, 2009). The electron affinity value for graphene means that the carbon atoms in graphene molecules can be easily ionized as compared to other non metallic elements such as oxygen and nitrogen. In comparison to conventional materials such as metals, graphene exhibits excellent thermal properties (Ghosh et al., 2008; Cai et al., 2010; Chen et al., 2011) with a thermal conductivity in the range of $\sim 3080 - 5150W/mK$ which makes graphene an integral part of future thermal management materials. Graphene also has an intrinsic tensile strength that is much greater than that of diamond that is within the range 0.5 TPa to 4 TPa (Zhao et al., 2002), which makes it a possible candidate for applications that require great strength. The excellent electronic properties of graphene have made it a strong candidate for use as a major component of supercapacitors (Novoselov et al., 2004; Castro Neto et al., 2007).

The list of possible applications of graphene seems to be ever growing with the some possible applications being in polymer matrix composites and targeted drug delivery. Ramanathan et al.

(2008) reported the so-called polystyrene-graphene composite that showed improved percolation threshold properties. Functionalized graphene was reported by Stankovich et al. (2006) to have glass transition temperatures of over $40°C$, which were unprecedented at that time. Frank et al. (2010) reported that graphene embedded in plastic beams showed remarkable compression buckling strain. Another application of graphene has been in the field of targeted drug delivery. Recent studies on how water soluble polyethyleneimine functionalized single wall carbon nanotubes were loaded with the therapeutic agents acetic acid and the gemcitabine analogue, deoxycytidine to control the release of agents from the single wall carbon nanotube conjugate were reported by Mendes et al. (2013). Graphene oxide has also been shown to be a simple synthetic route for loading and targeted delivery of anti cancer drugs (Zhang et al., 2010). Reduced graphene oxide has also been used to electrochemically trigger the release of anticancer drug doxorubicin from flexible electrodes modified electrophoretically (Zhang et al., 2011). Several other applications of graphene compounds are being vigourously explored across all fields. Some selected examples worth mentioning include the work of Bao et al. (2011) who reported chitosan functionalized graphene oxide being used to load and deliver the water soluble anti-cancer drug camptothecin via the covalent bonds formed when the lobes of nearby atoms overlap. Graphene paper which is fragile, porous, opaque and metallic has also been reported by Geim (2009); this is also another application worth mentioning.

2.2 Theoretical background

The vast array of the many possible applications in several areas make graphene such an important material that is worth studying in depth and understanding in the context of nonlinearities and chaos. Nonlinearities occur in almost all materials and this topic has been studied for decades since the publication of some seminal paper on the subject by Fermi, Pasta and Ulam (1955) and Anderson (1958) in the 1950s. The Fermi group got some interesting results, as they saw that in a nonlinear lattice model energy equipartition does not always occur in the presence of nonlinearities. Anderson showed that in the presence of disorder (i.e. a randomness in the system's parameters) linear systems can exhibit energy localization. This means that energy excitations can remain confined within a small region of the lattice and consequently the material exhibits some insulating properties.

Over the years many researchers have studied chaoticity and energy transport in several 1D lattice models such as the Klein-Gordon system (Skokos et al., 2013; Mulansky and Pikovsky, 2013), the Discrete Nonlinear Schrödinger Equation (Laptyeva et al., 2010), the Fermi-Pasta-Ulam (FPU) model (Palleari and Penati, 2008) to mention a few, while considerably less work has been done on 2D lattices. In this study we intend to fill part of that gap by studying the dynamics of a 2D lattice of graphene. Graphene is an exciting new hotbed for research as its behavior is still not fully understood and a number of important questions remain open related for example to its chaoticity.

It is worth noting that disorder can be encountered also in graphene models. For example a sample of graphene which contains its naturally occurring isotopes of carbon-12 , carbon-13 and carbon-14 is actually a disordered system. Nowadays it is generally believed that thermalization of disordered lattices is a chaotic process. The characteristics of such phenomena are not yet well understood, although the first attempt to systematically study chaos in a 1D disordered nonlinear lattice was made by Skokos et al. (2013). Some researchers have shown that energy localization in general is destroyed by nonlinearities (Lahini et al., 2008; Flach et al., 2009; Vicencio and Flach, 2009; Laptyeva et al., 2010; Bodyfelt et al., 2011; Mulansky and Pikovsky, 2013). In graphene models

nonlinearity exists intrinsically hence it is expected for such systems to be chaotic although the chaotic nature of graphene has not been studied in detail. In the remaining chapter a Hamiltonian model of graphene is presented along with a detailed derivation of the system's equations of motion.

2.3 A Hamiltonian model of graphene

In recent years there has been a lot of research interest into the dynamical properties of graphene (Xu and Buehler, 2010; Paulatto et al., 2013; Da Silva et al., 2014; Koukaras et al., 2015; Michel et al., 2015). To numerically study the dynamical behavior of graphene a force field model is required. Several such models have been considered in the literature especially for carbon macromolecules with the one presented by Tersoff (1988) being an example. Stuart et al. (2000), Los and Fasolino (2003) and Los et al. (2005) give other types of force fields which have been used to describe bond stretching and structure deformation in graphene.

In Figure 2.3 we schematically present the arrangement of the carbon atoms in the case of equilibrium. We use a cartesian coordinate system (x, y) in such a way that the position of each carbon atom on the plane can be obtained by a set of two indices $i = 1, 2, 3, \ldots, M$ and $j = 1, 2, 3, \ldots, N$ with the point $(1, 1)$ i.e. $i = 1, j = 1$, being at the origin. The red solid dots in Figure 2.3 show the positions of the carbon atoms in the system's equilibrium state and the red open circles show the nearest neighbors at the borders of our lattice. Since in our model we consider periodic boundary conditions, the atoms denoted by the circles are duplicates of atoms existing at the opposite edge. For example points $(i, N + 1)$ for $i = 1, 2, 3, \ldots, M$ correspond to points $(i, 1)$. In a similar way $(i, 0) \equiv (i, N)$, $(M+1, j) \equiv (1, j)$ and $(0, j) \equiv (M, j)$ for $i = 1, 2, 3, \ldots M$ and $j = 1, 2, 3, \ldots, N$. The inter atomic bonds between neighboring atoms are shown by blue segments, while green segments denote the bonds at the boundaries of the lattice.

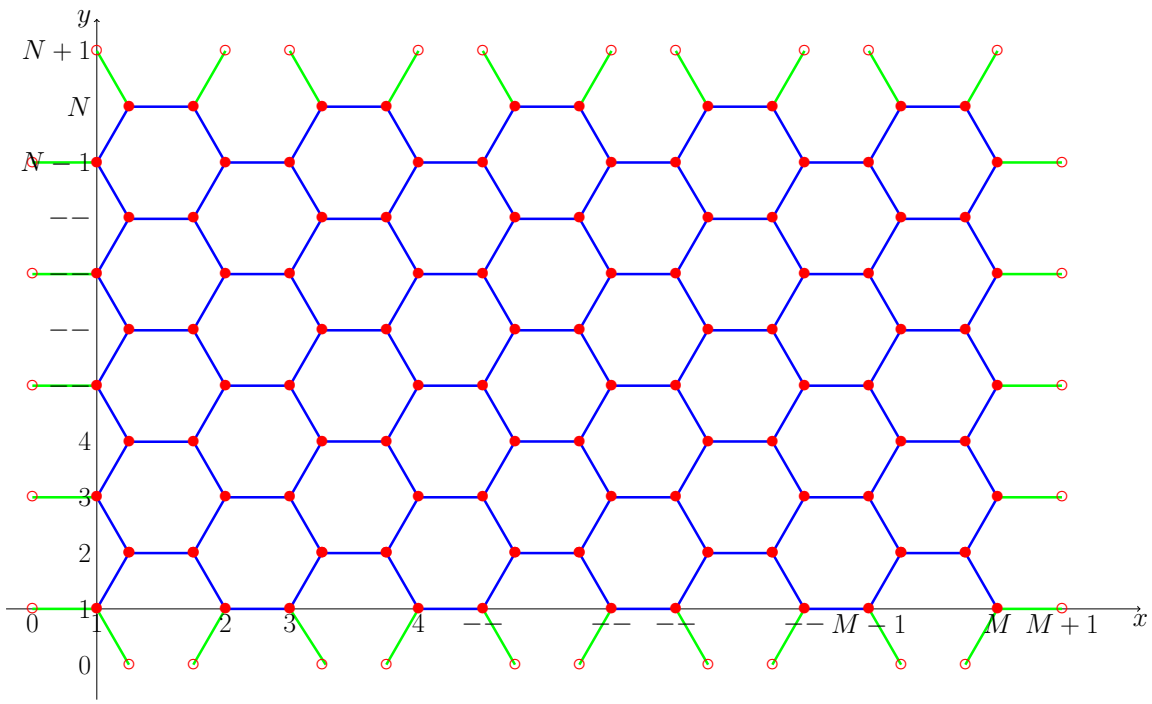

Figure 2.3: A graphical representation of the arrangement of carbon atoms in the equilibrium configuration of graphene.

In our study we use the empirical force fields described by explicit analytical functional expressions derived from first principles (Kalosakas et al., 2013). These functions proved to accurately describe the bond stretching and the deformation of the graphene hexagonal structure. The results obtained using these force fields by Kalosakas et al. (2013) confirm the accuracy and appropriateness of the used potentials as the molecular dynamics simulations presented there agree quite well with the reported density functional theory (DFT) calculations.

Let us now formulate the explicit Hamiltonian function of the considered graphene model of $M \times N$ carbon atoms with periodic boundary conditions. The nearest neighbors for each arbitrary carbon atom at position (i, j) depend on whether the sum of i and j is odd or even. If $i + j = odd$, the nearest neighbors are $(i, j + 1)$, $(i, j - 1)$ and $(i + 1, j)$ [Figure 2.4(a)]. For the case $i + j = even$, only the indices of atoms at $(i + 1, j)$ change to $(i - 1, j)$ whilst the other neighbors' indices remain unchanged [Figure 2.4(b)].

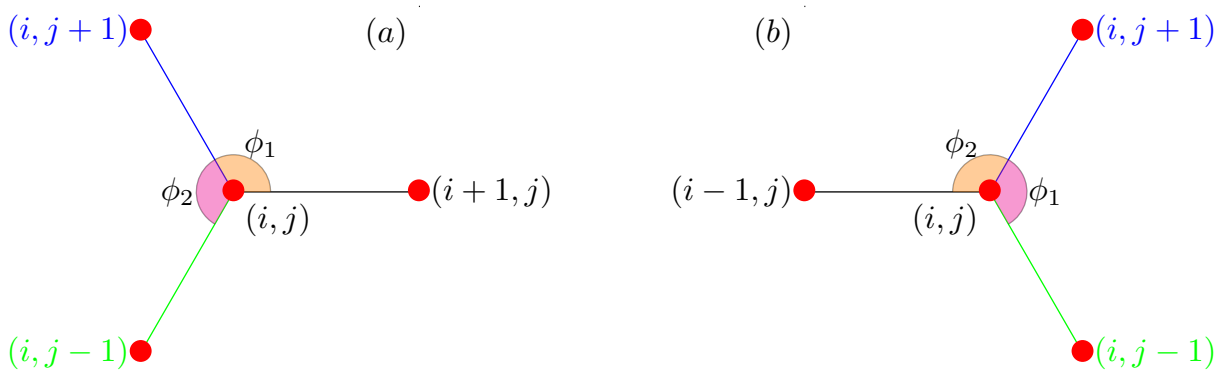

Figure 2.4: Nearest neighbors for an arbitrary atom at position (i, j) when the sum of i and j is odd (a) or even (b).

For a carbon atom at position (i, j) the angle which has as vertex point (i, j) and is formed by its neighboring atoms at positions (k, l) and (m, n) (Figure 2.5) is denoted by ${}_{(i,j)}\phi^{(k,l)}_{(m,n)}$. This notation describes the anticlockwise angle from atom (m, n) to atom (k, l).

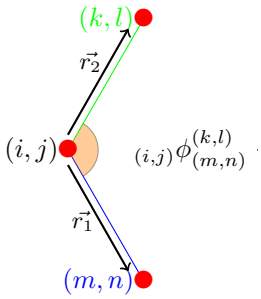

Figure 2.5: A diagram showing the notation used to denote the angle between any three neighboring carbon atoms.

The radial distance between any two carbon atoms at positions (i, j) and (k, l) in vector notation is

$$\vec{r}^{(k,l)}_{(i,j)} = (x_{(k,l)} - x_{(i,j)})\hat{\mathbf{i}} + (y_{(k,l)} - y_{(i,j)})\hat{\mathbf{j}}, \tag{2.1}$$

where $\hat{\mathbf{i}}$ and $\hat{\mathbf{j}}$ are the unit vectors along the x and y axis respectively. The length of vector (2.1) is

$$r_{(i,j)}^{(k,l)} = \sqrt{(x_{(k,l)} - x_{(i,j)})^2 + (y_{(k,l)} - y_{(i,j)})^2}. \tag{2.2}$$

In our study we use the Morse potential to describe the bond stretching of atoms at positions (i, j) and (k, l):

$$V_{(i,j)}^{M}(k,l) = D\left[e^{-a\left(r_{(i,j)}^{(k,l)} - r_0\right)} - 1\right]^2. \tag{2.3}$$

The values of the parameters D, a and r_0 in (2.3) have been determined through appropriate fittings of the numerical density functional theory results and are $5.7eV$, $1.96\mathring{A}^{-1}$ and $1.42\mathring{A}$ respectively (Kalosakas et al., 2013).

The bending potential V^A related to an angle ${}_{(i,j)}\phi_{(m,n)}^{(k,l)}$ is given by

$$V_{(i,j)}^{A}\left[(k,l),(m,n)\right] = \frac{d}{2}\Big[{}_{(i,j)}\phi_{(m,n)}^{(k,l)} - \phi_0\Big]^2 - \frac{d'}{3}\Big[{}_{(i,j)}\phi_{(m,n)}^{(k,l)} - \phi_0\Big]^3, \tag{2.4}$$

with parameters d, d' and ϕ_0 assuming the values $7.0eV/rad^2$, $4.0eV/rad^3$ and $\frac{2\pi}{3}$ respectively (Kalosakas et al., 2013).

In Figure 2.6 we note the arrangement of the atoms around the (i, j) atom for $i + j = odd$ [Figure 2.6(a)] and $i + j = even$ [Figure 2.6(b)]. We note that we included all the angles which are affected by the position of the (i, j) atom. For example, in Figure 2.6(a) atom (i, j) is the vertex of three angles ${}_{(i,j)}\phi_{(i,j+1)}^{(i,j-1)}$, ${}_{(i,j)}\phi_{(i,j-1)}^{(i+1,j)}$ and ${}_{(i,j)}\phi_{(i+1,j)}^{(i,j+1)}\Big)$, and at the edge of six other angles $\Big[{}_{(i,j+1)}\phi_{(i,j)}^{(i,j+2)}$, ${}_{(i,j+1)}\phi_{(i-1,j+1)}^{(i,j)}$, ${}_{(i,j-1)}\phi_{(i,j)}^{(i-1,j-1)}$, ${}_{(i,j-1)}\phi_{(i,j-2)}^{(i,j)}$, ${}_{(i+1,j)}\phi_{(i,j)}^{(i+1,j-1)}$ and ${}_{(i+1,j)}\phi_{(i+1,j-1)}^{(i,j)}\Big]$. A similar arrangement is seen for $i + j = even$ [Figure 2.6(b)].

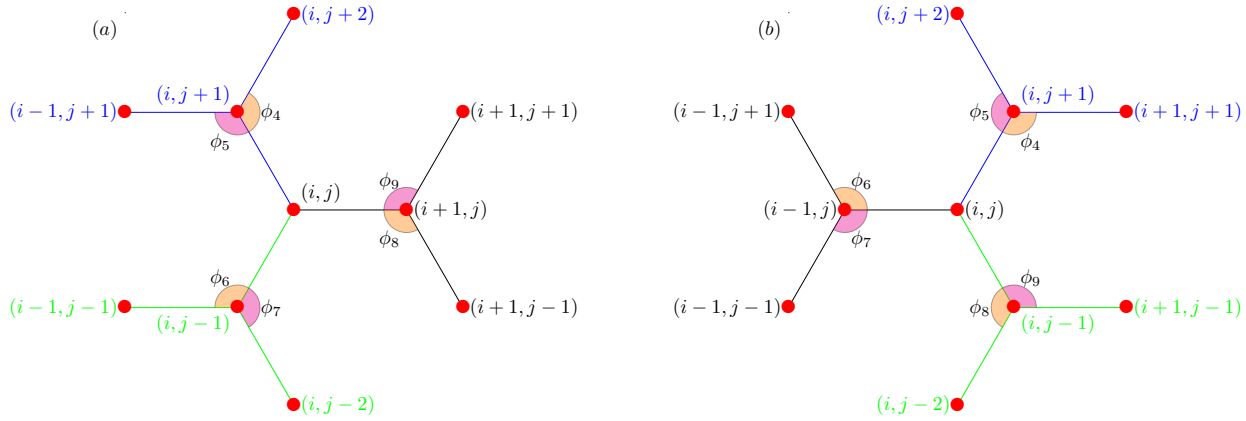

Figure 2.6: The arrangement of neighbouring atoms for an arbitrary point at (i, j) when the sum of i and j is odd(a) and even(b).

The Hamiltonian H for the graphene model for the cases $i + j = even$, $(i + j = odd)$ is the sum of the kinetic and the potential energy for all the carbon atoms at (i, j) , $i = 1, 2, 3, \ldots, M$, $j = 1, 2, 3, \ldots, N$

$$H=\sum_{i=1}^{M}\sum_{j=1}^{N}\left[\frac{p_{x_{(i,j)}}^2}{2m_c}+\frac{p_{y_{(i,j)}}^2}{2m_c}\right]+\frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{N}\left[V_{(i,j)}^{M}(i,j+1)+V_{(i,j)}^{M}(i,j-1)+V_{(i,j)}^{M}(i-1,j)\left[V_{(i,j)}^{M}(i+1,j)\right]\right]$$
$$+\sum_{i=1}^{M}\sum_{j=1}^{N}\left[V^{A}\left(\phi_{(i,j+1)}^{(i,j-1)}\right)_{(i,j)}+V^{A}\left(\phi_{(i,j-1)}^{(i-1,j)}\right)_{(i,j)}\left[V^{A}\left(\phi_{(i,j-1)}^{(i+1,j)}\right)_{(i,j)}\right]+V^{A}\left(\phi_{(i-1,j)}^{(i,j+1)}\right)_{(i,j)}\left[V^{A}\left(\phi_{(i+1,j)}^{(i,j+1)}\right)_{(i,j)}\right]\right] \tag{2.5}$$

We note that throughout this work, the terms written in black in equations are present for both cases $i+j=even$ and $i+j=odd$, while terms written in blue appear only when $i+j=even$ and terms in red exist only for $i+j=odd$. We consider the studied graphene model to be homogeneous, i.e. consisting of carbon atoms of the same atomic mass so that m_c is 12 atomic mass units.

2.4 Equations of motion

For a general Hamiltonian $H(q_i,p_i,t)$ with $i=1,2,3,\ldots,K$, (K being the number of degrees of freedom) where q_i, p_i are respectively the generalized positions and conjugate momenta, the Hamilton equations of motion are given by

$$\dot{q}_i=\frac{\partial H}{\partial p_i},\quad \dot{p}_i=-\frac{\partial H}{\partial q_i}\quad i=1,2,3,\ldots,K. \tag{2.6}$$

Thus for the graphene Hamiltonian in (2.5) the differential equations governing the time evolution of the position and momentum of atom (i,j) are

$$\dot{x}_{(i,j)}=\frac{p_{x_{(i,j)}}}{m_c},\quad \dot{y}_{(i,j)}=\frac{p_{y_{(i,j)}}}{m_c}\quad i=1,2,3,\ldots,M,\quad j=1,2,3,\ldots,N, \tag{2.7}$$

$$\dot{p}_{x_{(i,j)}}=-\frac{\partial H}{\partial x_{(i,j)}},\quad \dot{p}_{y_{(i,j)}}=-\frac{\partial H}{\partial y_{(i,j)}}\qquad i=1,2,3,\ldots,M,\quad j=1,2,3,\ldots,N. \tag{2.8}$$

We reemphasize that for the graphene model under study, we use periodic boundary conditions so that for example points $(i,N+1)$ for $i=1,2,3,\ldots,M$ correspond to points $(i,1)$. In a similar way $(i,0)\equiv(i,N)$, $(M+1,j)\equiv(1,j)$ and $(0,j)\equiv(M,j)$ for $i=1,2,3,\ldots M$ and $j=1,2,3,\ldots,N$. As the equations governing the time evolution of the momenta $p_{x_{(i,j)}}$ and $p_{y_{(i,j)}}$ are complicated, we will treat separately the terms related to the bond stretching, V^M, and bond deformation, V^A, potentials keeping as before the coloring convention of the equations parts depending on $i+j=odd$ or $i+j=even$.

To get the equations of motion due to the bond stretching between atoms (i,j) and atom (m,n) with respect to the position $x_{(i,j)}$ we consider the derivative of the radial distances $r_{(i,j)}^{(m,n)}$ with respect to $x_{(i,j)}$:

$$\frac{\partial r_{(i,j)}^{(m,n)}}{x_{(i,j)}}=\frac{x_{(i,j)}-x_{(m,n)}}{r_{(m,n)}^{(i,j)}}. \tag{2.9}$$

The equation of motion of $p_{x_{(i,j)}}^{M}$ due to bond strecthing between two atoms (i,j) and (m,n) is given by

$$\dot{p}_{x_{(i,j)}}^{M}=-\frac{\partial V^M}{\partial r_{(i,j)}^{(m,n)}}\frac{\partial r_{(i,j)}^{(m,n)}}{\partial x_{(i,j)}}. \tag{2.10}$$

For $V^M_{(i,j)}(m,n) = D\left[e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}-1\right]^2$ the derivative of $V^M_{(i,j)}(m,n)$ with respect to $r^{(m,n)}_{(i,j)}$ is

$$\frac{\partial V^M}{\partial r^{(m,n)}_{(i,j)}} = 2D\left[e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}. \tag{2.11}$$

Then using equations (2.9) and (2.11) equation (2.10) takes the form

$$\dot{p}^M_{x_{(i,j)}} = -2D\left[e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}\frac{2[x_{(i,j)}-x_{(m,n)}]}{r^{(i,j)}_{(m,n)}}. \tag{2.12}$$

In a similar way, we find

$$\dot{p}^M_{y_{(i,j)}} = -2D\left[e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(m,n)}_{(i,j)}-r_0\right)}\frac{[y_{(i,j)}-y_{(m,n)}]}{r^{(i,j)}_{(m,n)}}. \tag{2.13}$$

So, the contribution of the Morse potential, i.e. the bond stretching part of the Hamiltonian, to the equations of motion for the momenta is

$$\begin{aligned}
\dot{p}^M_{x_{(i,j)}} &= 2aDe^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]\frac{[x_{(i,j)}-x_{(i,j+1)}]}{r^{(i,j+1)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]\frac{[x_{(i,j)}-x_{(i,j-1)}]}{r^{(i,j-1)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]\frac{[x_{(i,j)}-x_{(i-1,j)}]}{r^{(i-1,j)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]\frac{[x_{(i,j)}-x_{(i+1,j)}]}{r^{(i+1,j)}_{(i,j)}}
\end{aligned} \tag{2.14}$$

and

$$\begin{aligned}
\dot{p}^M_{y_{(i,j)}} &= 2aDe^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]\frac{[y_{(i,j)}-y_{(i,j+1)}]}{r^{(i,j+1)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]\frac{[y_{(i,j)}-y_{(i,j-1)}]}{r^{(i,j-1)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]\frac{[y_{(i,j)}-y_{(i-1,j)}]}{r^{(i-1,j)}_{(i,j)}}\\
&+2aDe^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]\frac{[y_{(i,j)}-y_{(i+1,j)}]}{r^{(i+1,j)}_{(i,j)}}.
\end{aligned} \tag{2.15}$$

In order to obtain the part of the equations of motion related to the bending or in other words the part induced by the deformation of the angle ${}_{(i,j)}\phi^{(k,l)}_{(m,n)}$, we evaluate the equations of motion for $p^A_{x_{(i,j)}}$ as

$$\dot{p}^A_{x_{(i,j)}} = -\frac{\partial V^A}{\partial_{(i,j)}\phi^{(m,n)}_{(k,l)}}\frac{\partial_{(i,j)}\phi^{(m,n)}_{(k,l)}}{\partial\cos\left({}_{(i,j)}\phi^{(m,n)}_{(k,l)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(m,n)}_{(k,l)}\right)}{x_{(i,j)}}. \tag{2.16}$$

We note that $x_{(i,j)}$ appears also in the cases where atom (i,j) is at the edge of an angle and not at its vertex. We will refer to such cases later on. For the case of an angle ${}_{(i,j)}\phi^{(k,l)}_{(m,n)}$ whose vertex is at (i,j) we can evaluate $\cos\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)$ based on the inner product of vectors $\vec{r}^{\,(m,n)}_{(i,j)}$ and $\vec{r}^{\,(k,l)}_{(i,j)}$ which we denote by $\mathbf{r}_1$ and $\mathbf{r}_2$ respectively (Figure 2.5).

$$\cos\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right) = \frac{(x_{(k,l)}-x_{(i,j)})(x_{(m,n)}-x_{(i,j)})+(y_{(k,l)}-y_{(i,j)})(y_{(m,n)}-y_{(i,j)})}{\sqrt{(x_{(m,n)}-x_{(i,j)})^2+(y_{(m,n)}-y_{(i,j)})^2}\sqrt{(x_{(k,l)}-x_{(i,j)})^2+(y_{(k,l)}-y_{(i,j)})^2}}. \tag{2.17}$$

The $\sin\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)$ can be obtained from the cross product of vectors $\mathbf{r}_1$ and $\mathbf{r}_2$.

$$\|\mathbf{r}_1\times\mathbf{r}_2\| = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ x_{(i,j)}-x_{(m,n)} & y_{(i,j)}-y_{(m,n)} & 0 \\ x_{(i,j)}-x_{(k,l)} & y_{(i,j)}-y_{(k,l)} & 0 \end{vmatrix} = \|\mathbf{r}_1\|\|\mathbf{r}_2\|\sin\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right), \tag{2.18}$$

so that

$$\frac{1}{\sin\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)} = \frac{\sqrt{\left[x_{(i,j)}-x_{(m,n)}\right]^2+\left[y_{(i,j)}-y_{(m,n)}\right]^2}\sqrt{\left[x_{(i,j)}-x_{(k,l)}\right]^2+\left[y_{(i,j)}-y_{(k,l)}\right]^2}}{\sqrt{\left(\left[x_{(i,j)}-x_{(m,n)}\right]\left[y_{(i,j)}-y_{(k,l)}\right]-\left[x_{(i,j)}-x_{(k,l)}\right]\left[y_{(i,j)}-y_{(m,n)}\right]\right)^2}}. \tag{2.19}$$

The partial derivatives appearing in (2.16) can be evaluated as follows

$$\frac{\partial V^A}{\partial_{(i,j)}\phi^{(k,l)}_{(m,n)}} = d\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}-\frac{2\pi}{3}\right)-d'\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}-\frac{2\pi}{3}\right)^2, \tag{2.20}$$

$$\frac{\partial_{(i,j)}\phi^{(k,l)}_{(m,n)}}{\partial\cos_{(i,j)}\phi^{(k,l)}_{(m,n)}} = \frac{-1}{\sin\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)}, \tag{2.21}$$

$$\begin{aligned}\frac{\partial\cos\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)}{\partial x_{(i,j)}} &= \frac{\left[x_{(i,j)}-x_{(k,l)}\right]+\left[x_{(i,j)}-x_{(m,n)}\right]}{r^{(k,l)}_{(i,j)}r^{(m,n)}_{(i,j)}} \\ &-\frac{\mathbf{r}^{(k,l)}_{(i,j)}\cdot\mathbf{r}^{(m,n)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(m,n)}\right]\left(r^{(k,l)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(k,l)}\right]\left(r^{(m,n)}_{(i,j)}\right)^2\right]}{\left[r^{(k,l)}_{(i,j)}r^{(m,n)}_{(i,j)}\right]^3}.\end{aligned} \tag{2.22}$$

In the same spirit the partial derivative of $\cos\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)$ with respect to $y_{(i,j)}$ is

$$\begin{aligned}\frac{\partial\cos\left({}_{(i,j)}\phi^{(k,l)}_{(m,n)}\right)}{\partial y_{(i,j)}} &= \frac{\left[y_{(i,j)}-y_{(k,l)}\right]+\left[y_{(i,j)}-y_{(m,n)}\right]}{r^{(k,l)}_{(i,j)}r^{(m,n)}_{(i,j)}} \\ &-\frac{\mathbf{r}^{(k,l)}_{(i,j)}\cdot\mathbf{r}^{(m,n)}_{(i,j)}\left[\left[y_{(i,j)}-y_{(m,n)}\right]\left(r^{(k,l)}_{(i,j)}\right)^2+\left[y_{(i,j)}-y_{(k,l)}\right]\left(r^{(m,n)}_{(i,j)}\right)^2\right]}{\left[r^{(k,l)}_{(i,j)}r^{(m,n)}_{(i,j)}\right]^3}.\end{aligned} \tag{2.23}$$

Thus, the parts of the equation of motions for the momenta $p_{x(i,j)}$ and $p_{x(i,j)}$ related to the bending potential V^A (2.4) are given by the following equations

$$
\begin{aligned}
\dot{p}^{A}_{x_{(i,j)}} = &-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}{\partial\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}{\partial\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}\frac{\partial\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}{\partial\cos\left({}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right)}\frac{\partial\cos\left({}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}{\partial\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}\frac{\partial\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}{\partial\cos\left({}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right)}\frac{\partial\cos\left({}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right)}{\partial x_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}{\partial\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right)}{\partial x_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right)}{\partial x_{(i,j)}}.
\end{aligned}
\tag{2.24}
$$

Similarly we get,

$$
\begin{aligned}
\dot{p}^{A}_{y_{(i,j)}} = &-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i-1,j)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}{\partial\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}{\partial\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}\frac{\partial\left[{}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right]}{\partial\cos\left({}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right)}\frac{\partial\cos\left({}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}{\partial\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}\frac{\partial\left[{}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right]}{\partial\cos\left({}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right)}\frac{\partial\cos\left({}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}{\partial\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}\frac{\partial\left[{}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right]}{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right)}\frac{\partial\cos\left({}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}\right)}{\partial y_{(i,j)}}\\
&-\frac{\partial V^{A}\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}{\partial\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}\frac{\partial\left[{}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right]}{\partial\cos\left({}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right)}\frac{\partial\cos\left({}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}\right)}{\partial y_{(i,j)}}-\frac{\partial V^{A}\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}\frac{\partial\left[{}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right]}{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right)}\frac{\partial\cos\left({}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}\right)}{\partial y_{(i,j)}}.
\end{aligned}
\tag{2.25}
$$

In conclusion, based on equations (2.14), (2.15), (2.24) and (2.25) the equations of motion of the Hamiltonian (2.5) take the form

$$\dot{x}_{(i,j)} = \frac{p^M_{x_{(i,j)}}}{m_c} \tag{2.26}$$

$$\dot{y}_{(i,j)} = \frac{p^M_{y_{(i,j)}}}{m_c} \tag{2.27}$$

$$\dot{p}_{x_{(i,j)}} = \dot{p}^M_{x_{(i,j)}} + \dot{p}^A_{x_{(i,j)}} \tag{2.28}$$

$$\dot{p}_{x_{(i,j)}} = \dot{p}^M_{y_{(i,j)}} + \dot{p}^A_{y_{(i,j)}} \tag{2.29}$$

2.5 Numerical Results

2.5.1 Initial lattice setup

The model used has M particles in the x direction and N particles in the y direction. In our simulation, we chose an even number of atoms in both the x and y directions as this choice makes the model easier to manipulate and manage. In particular we consider $M = 12$ and $N = 16$ particles. Periodic boundary conditions are employed at all edges.

The initial positions of the atoms used in our simulations were the equilibrium positions, whilst for the momenta in both the x and y directions the initial values followed a standard normal distribution with mean 0 and variance 1 nm s^{-1}. These initial momenta were created using the Box and Muller (1958) algorithm. After getting the set of the initial momenta, the center of mass of their distributions were obtained in order to eventually set their mean values to zero so that the lattice does not drift. This procedure was performed as follows (Frenkel and Smit, 2002);

- Sum the momenta in the x and y directions for $M \times N$ number of particles. $\sum_{i=1}^{M}\sum_{j=1}^{N} P_{x_{(i,j)}}$ and $\sum_{i=1}^{M}\sum_{j=1}^{N} P_{y_{(i,j)}}$
- Divide the sums by $M \times N$ to get the residuals which must be removed from each component so that the sums are set to zero.
- For each component of momentum we remove the residuals to put the center of mass to zero: $p_{x_{(i,j)}} = P_{x_{(i,j)}} - \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} P_{x_{(i,j)}}$ and $p_{y_{(i,j)}} = P_{y_{(i,j)}} - \frac{1}{M \times N}\sum_{i=1}^{M}\sum_{j=1}^{N} P_{y_{(i,j)}}$
- Scale these momenta to match the desired total energy value. The total kinetic energy in both directions is also summed up and divided by $2m_c$. The new value of the kinetic energy E_{rand} is either greater or smaller than the required value of the total lattice energy E_{tot}. The scaling factor ld is therefore $\sqrt{\frac{E_{rand}}{E_{tot}}}$. To scale the energy to the required value we multiply the momenta by ld.

The sums of the new momenta $p_{x_{(i,j)}}$ and $p_{y_{(i,j)}}$ is zero. A Fortran90 code that performs all these steps has been included in the appendix section.

In this study, the performances, i.e. accuracy and Central Processing Unit (CPU) time, of several symplectic integrators were analyzed. In order to make this evaluation a good balance between accuracy and required CPU time had to be obtained hence a number of 4^{th} order symplectic integrators were tested. The integrators considered were the Forest and Ruth (1990) SI and the ABA fourth order class of symplectic integrators: ABA104, ABA864 and ABA1064 (Blanes et al., 2013).

Table 2.1: Coefficients of the 4^{th} order Forest and Ruth SI.

SI	**Coefficients**
Forest & Ruth	$c_1 = 0.675603595979828817023843904485$
	$d_1 = 1.351207191959657634047687808971$
	$c_2 = -0.175603595979828817023843904485$7
	$d_2 = -1.702414383919315268095375617942$
	$c_3 = -0.175603595979828817023843904485$
	$d_3 = 1.351207191959657634047687808971$
	$c_4 = 0.675603595979828817023843904485$
	$d_4 = 0.000000000000000000000000000000$

The coefficients of the 4^{th} order symplectic integrator by Forest and Ruth (1990) are given in Table 2.1. The structure of the integrator is given as by the sequence $c_1d_2c_2d_2c_3d_3c_4d_4$. This SI has seven steps due to the fact that the last step is unity and can be written as

$$e^{\tau c_1 L_A} e^{\tau d_1 L_B} e^{\tau c_2 L_A} e^{\tau d_2 L_B} e^{\tau c_3 L_A} e^{\tau d_3 L_B} e^{\tau c_4 L_A} e^{\tau d_4 L_B}$$

The ABA class of SI has some fourth order symplectic integrators in which the first and last steps of the integrator correspond to the A a part of the Hamiltonian H. The sequences of the coefficients for the ABA104 and ABA864 is $c_1d_1c_2d_2c_3d_3c_4d_4c_4d_3c_3d_2c_2d_1c_1$ whilst that of the ABA1064 is $c_1d_1c_2d_2c_3d_3c_4d_4c_5d_4c_4d_3c_3d_2c_2d_1c_1$. The structure of the ABA104 and ABA864 SI is

$$e^{\tau c_1 L_A} e^{\tau d_1 L_B} e^{\tau c_2 L_A} e^{\tau d_2 L_B} e^{\tau c_3 L_A} e^{\tau d_3 L_B} e^{\tau c_4 L_A} e^{\tau d_4 L_B} e^{\tau c_4 L_A} e^{\tau d_3 L_B} e^{\tau c_3 L_A} e^{\tau d_2 L_B} e^{\tau c_2 L_A} e^{\tau d_1 L_B} e^{\tau c_1 L_A}$$

and the appropriate coefficients of these SIs are given in Table 2.2. The sequence of the ABA1064 integrator is $c_1d_1c_2d_2c_3d_3c_4d_4c_5d_4c_4d_3c_3d_2c_2d_1c_1$. The ABA1064 SI is then composed using the coefficients in Table 2.2 in a similar manner.

Table 2.2: Coefficients of the 4^{th} order ABA class of SIs.

SI	**Coefficients**
ABA104	$c_1 = 0.047067100645972506129478876372$
	$d_1 = 0.11888191736819701994535039508 5$
	$c_2 = 0.184756935417088106924737619370$
	$d_2 = 0.241050460551501565744166786590$
	$c_3 = 0.282706005679836205324361656554$
	$d_3 = -0.273286666705323806054311398166$
	$c_4 = -0.014530041742896818378578152296$
	$d_4 = 0.826708577571250440729588432981$
ABA864	$c_1 = 0.071133426498223117777938730006$
	$d_1 = 0.183083687472197221961703757166$
	$c_2 = 0.241153427956640098736487795326$
	$d_2 = 0.310782859898574869507522291054$
	$c_3 = 0.521411761772814789212136078067$
	$d_3 = -0.026564618511958800697212137916$
	$c_4 = -0.333698616227678005726562603400$
	$d_4 = 0.065396142282373418455972179391$
ABA1064	$c_1 = 0.038094497422412195456975322308$
	$d_1 = 0.095858880837075210610771503771$
	$c_2 = 0.145298716116913749294020072660$
	$d_2 = 0.204446153142998780680507783916$
	$c_3 = 0.207627695725541250716205611324$
	$d_3 = 0.217070347978991101714338592430$
	$c_4 = 0.435909703651526159223154862401$
	$d_4 = -0.017375381959065093005617880118$
	$c_5 = -0.653861225832786709380711737390$

These SIs were analyzed for a graphene lattice of 192 atoms i.e $M = 12$, $N = 16$ at an energy of $2eV$ and the ABA1064 performed better than the rest of the integrators. However what we require is not just accuracy. We have to achieve a certain level of accuracy along with computational speed. We decided that a relative energy error $\Delta H(t)$ (1.16) of the order of 10^{-6} would be sufficient for our simulations. The results in Figure 2.7 were obtained with all the four SIs under consideration

utilizing a time step of $\tau = 0.05$.

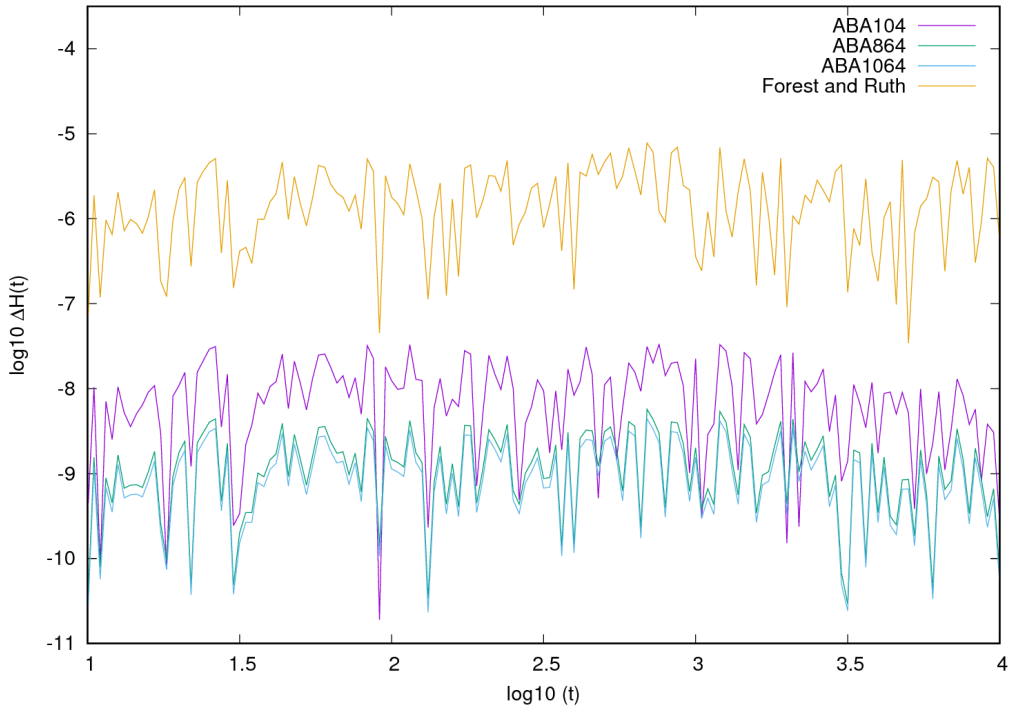

Figure 2.7: A comparison of the accuracy of some 4^{th} order symplectic integrators.

A comparison of the required CPU times (Figure 2.8) shows that the ABA864 is the best SI. The relative energy error, $\Delta H(t)$ was maintained at about 10^{-6}. This $\Delta H(t)$ was achived when the time steps used for the ABA104, ABA864, ABA1064 and the Forest and Ruth SIs were $\tau = 0.15$, $\tau = 0.20$, $\tau = 0.20$ and $\tau = 0.05$ respectively.

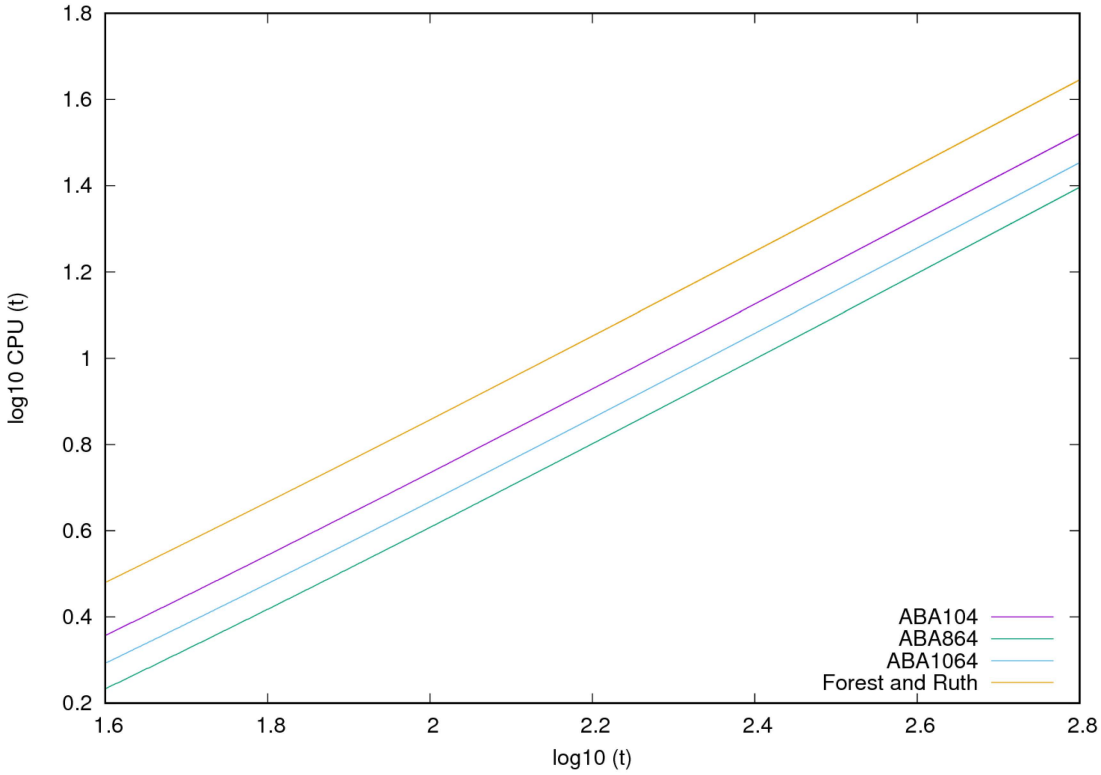

Figure 2.8: A comparison of the CPU times of some 4^{th} order SIs.

We show the results of the analysis of the ABA 4^{th} order class of SI in Figure 2.9 where we plot the time evolution of the relative energy for all tested SIs. The results show that the ABA864 SI exhibits the best performance.

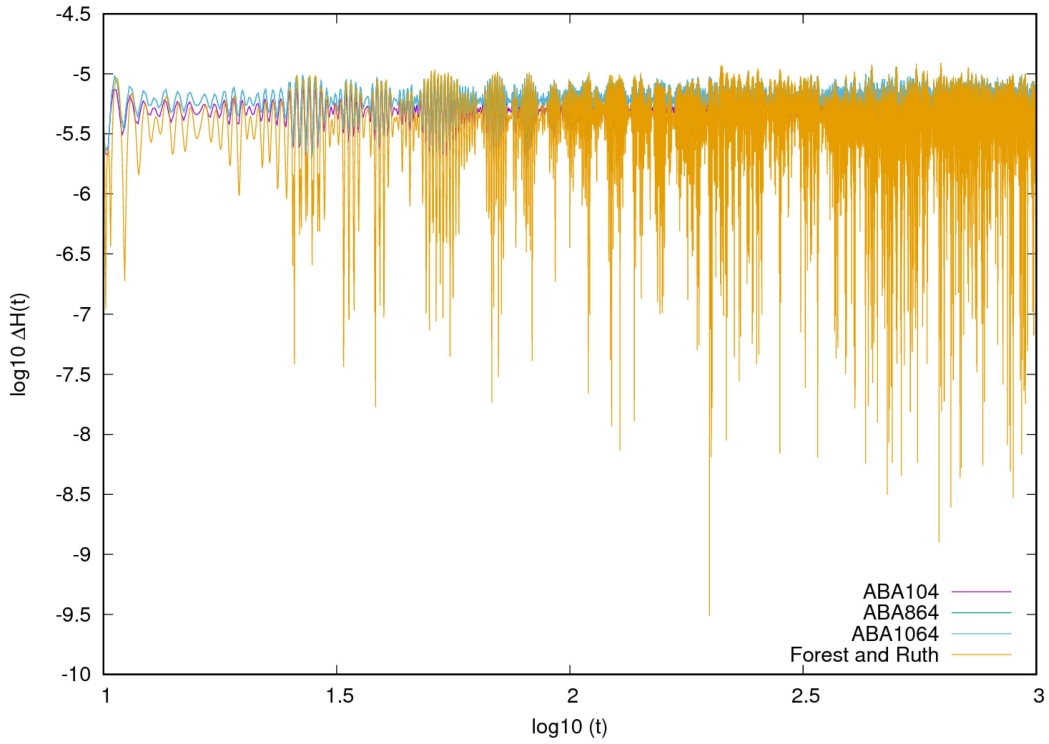

Figure 2.9: The time evolution of $\Delta H(t)$ for time steps $\tau = 0.15$ (ABA104), $\tau = 0.20$ (ABA864), $\tau = 0.20$ (ABA1064) and $\tau = 0.05$ (Forest and Ruth).

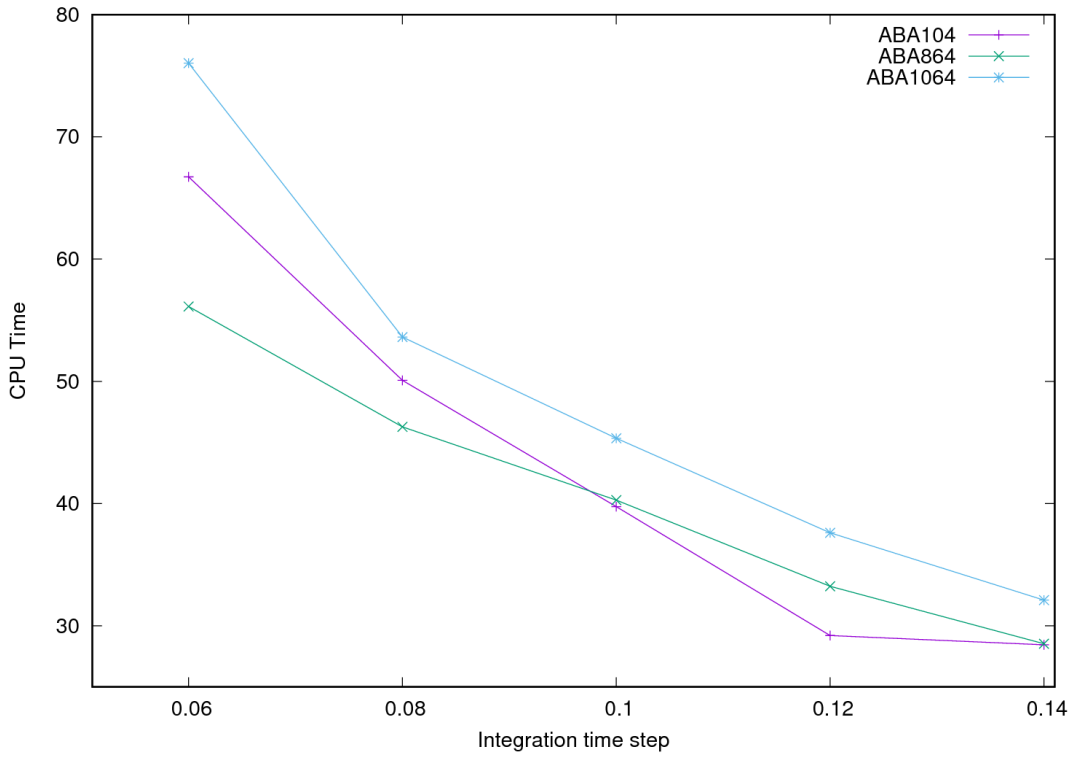

Figure 2.10: A comparison of the CPU time against time step for the 4^{th} order ABA104, ABA864 and ABA1064 symplectic integrators.

2.5.2 System's relaxation

The average distance covered by the atoms from their equilibrium positions $x_{eq_{(i,j)}}$ and $y_{eq_{(i,j)}}$ or the so called root mean square displacement (RMSD) is another quantity which was computed for a lattice of 12×16 carbon atoms with a total lattice energy of $2eV$. The RMSD is given by

$$RMSD = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\sqrt{[x_{eq_{(i,j)}} - x_{(i,j)}]^2 + [y_{eq_{(i,j)}} - y_{(i,j)}]^2} \tag{2.30}$$

In our computation of the RMSD, we used the ABA1064 SI with an integration time step of $\tau = 0.2$. The RMSD allows us to get an idea of when the system has reached relaxation. The RMSD tends to be more or less constant when such a state of relaxation has been reached. Figure 2.11 shows that such a state is reached almost immediately for graphene. When such a state is reached samples for computation of observables such as temperature can be taken. The RMSD was computed using the equilibrium positions of an unperturbed lattice where all the angles in the lattice are exactly $120°$ as the reference points.

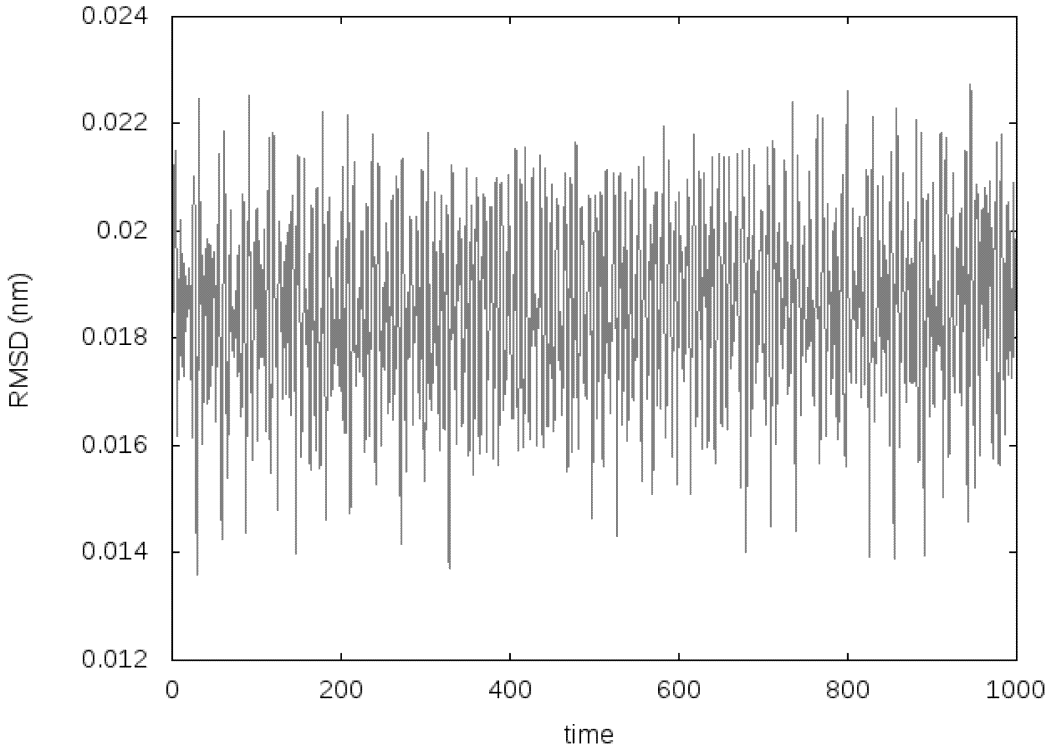

Figure 2.11: The time evolution of the root mean square displacement (RMSD) of atoms in a graphene lattice of 192 atoms with total energy 2 eV.

Plotting in Figure 2.12 the time evolution of the system's kinetic and the potential energy we see that these quantities reach quite fast (after about 10-20 time units) an equilibrium state, fluctuating around their mean value, which is equal to the half of the total energy.

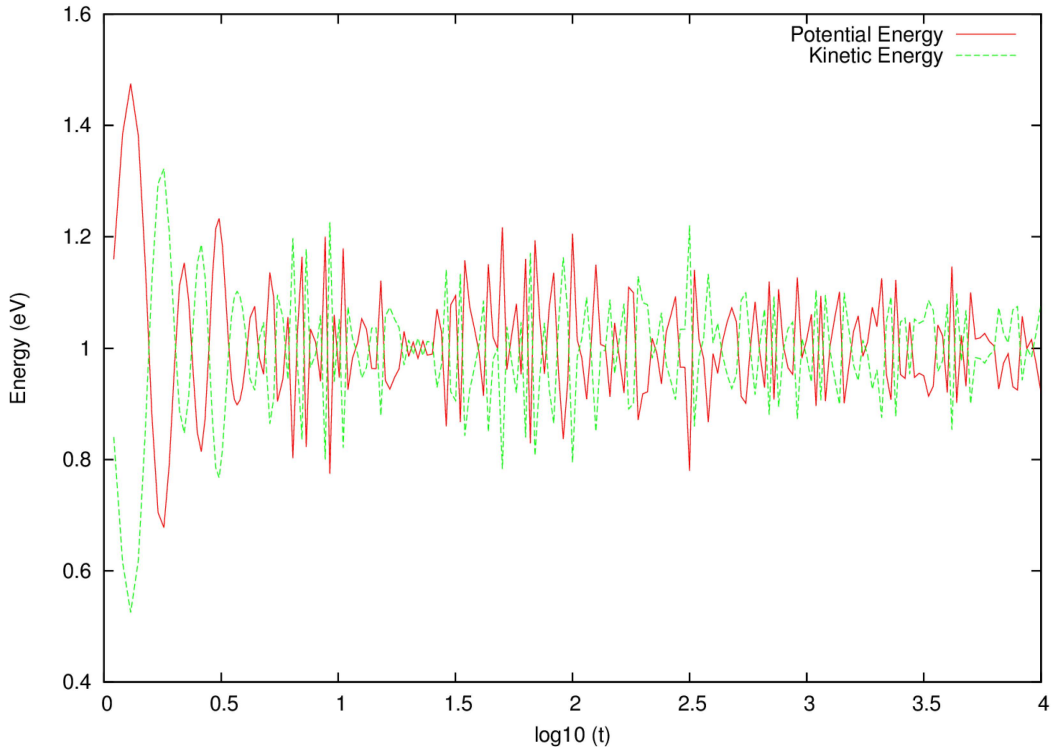

Figure 2.12: The time evolution of the kinetic (green curve) and the potential energy (red curve) of a graphene lattice of 192 atoms with total energy 2 eV.

Performing in Figures 2.13 and 2.14 a statistical analysis for different values of the system's total energy (in particular for 2 eV, 10 eV and 20 eV) and several different initial conditions, we get again clear evidences of fast relaxation. We note that in Figure 2.13 we plot the time evolution of the averaged value of RMDS (2.30) over 50 different initial conditions, while in Figure 2.14 the time evolution of the system's averaged kinetic and the potential energy is shown. In both figures the plotted error bars correspond to one standard deviation. It is worth noting that the length of these error bars increase as the system's total energy grows.

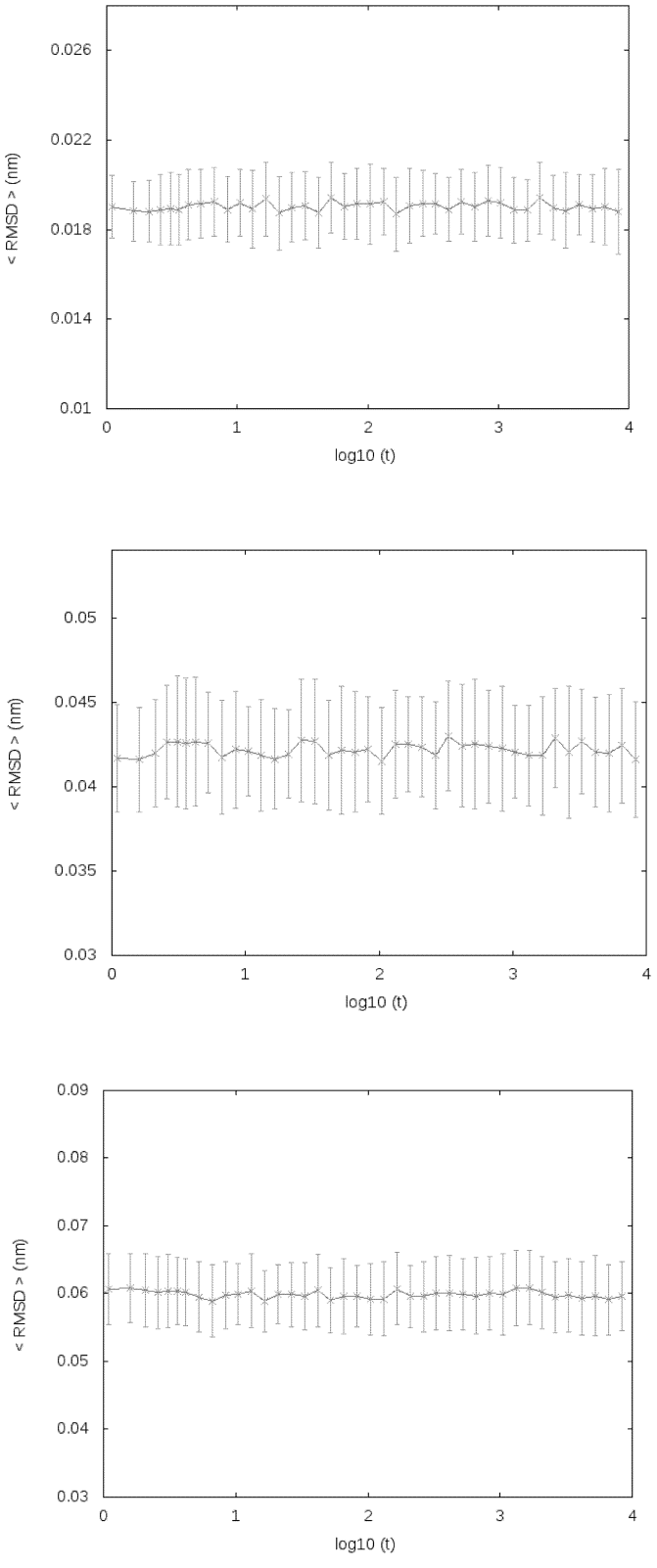

Figure 2.13: The time evolution of the averaged value of RMSD (2.30), $\langle \text{RMSD} \rangle$, over 50 different initial conditions of a graphene lattice of 192 atoms with total energy 2 eV (upper panel), 10 eV (middle panel) and 20 eV (lower panel). The error bars correspond to one standard deviation.

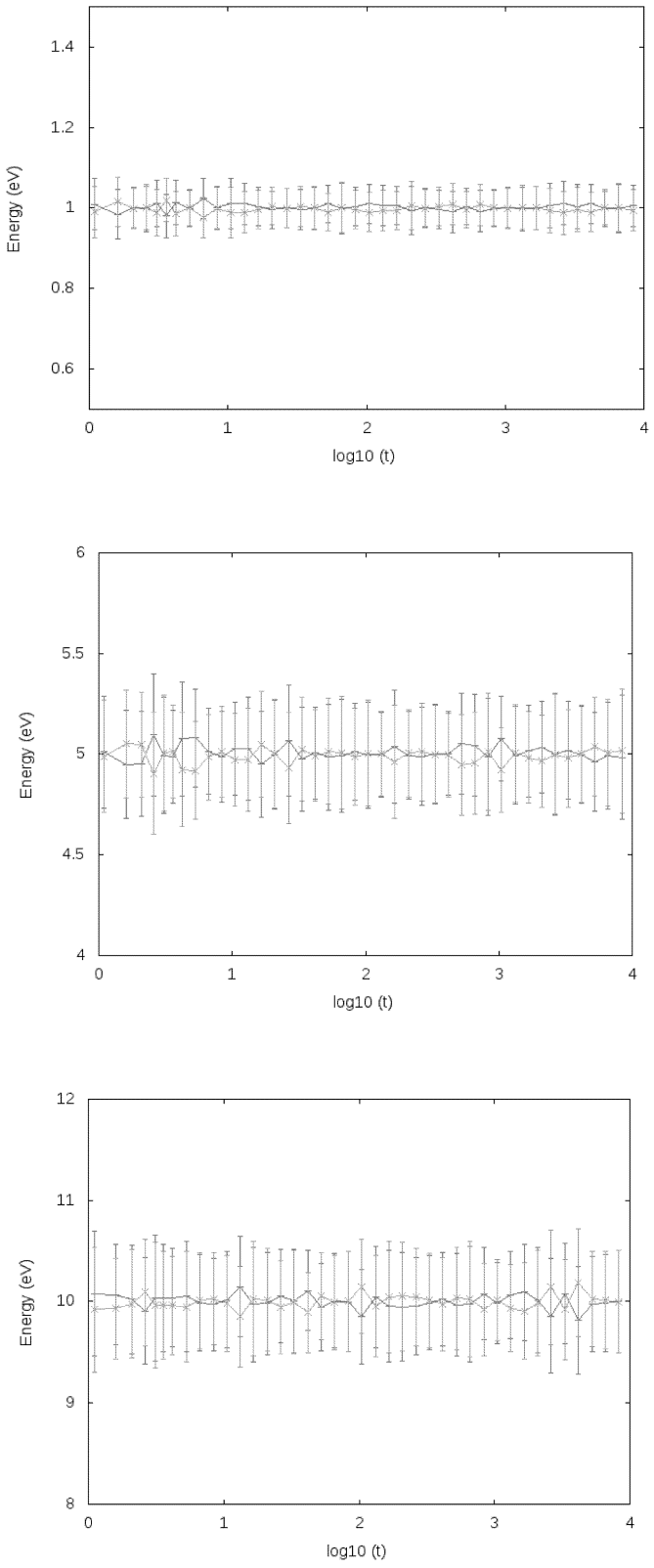

Figure 2.14: The time evolution of the averaged kinetic (green curves) and the potential energy (red curves) over 50 different initial conditions of a graphene lattice of 192 atoms with total energy 2 eV (upper panel), 10 eV (middle panel) and 20 eV (lower panel). The error bars correspond to one standard deviation.

2.5.3 System's temperature

The system's temperature for the graphene lattice was computed by applying the equipartition theorem bearing in mind that each lattice point (atom) has 2 degrees of freedom in the x and y direction. The lattice points were given an initial excitation such that the contribution of the potential energy was zero as explained in section 2.5.1. In other words the initial energy of the whole system was purely kinetic. The instantaneous system temperature T is given from the relationship

$$T = \frac{\langle E_T \rangle}{K_b} \tag{2.31}$$

where $\langle E_T \rangle$ is the mean kinetic energy which is given as

$$\langle E_T \rangle = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} \left(\frac{p_{x_{(i,j)}}^2}{2m_c} + \frac{p_{y_{(i,j)}}^2}{2m_c} \right) \tag{2.32}$$

while K_b is the Boltzmann constant whose value is $0.00008617 eV K^{-1}$.

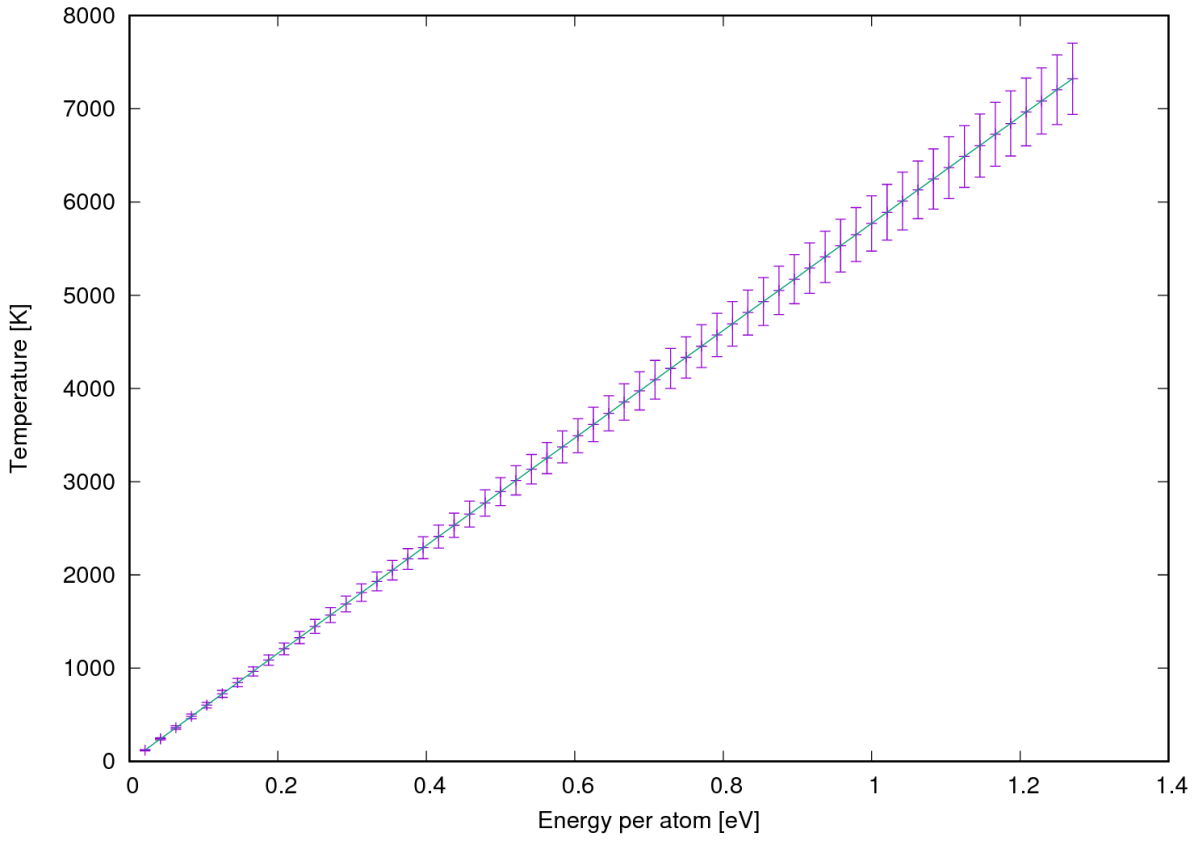

Figure 2.15: The system's temperature (with error bars) for different energies for graphene.

By performing simulations for several initial excitations we numerically obtained the temperatures of the system as a function of the total energy per atom in Figure 2.15. The average temperature

$\bar{T}$ of the lattice over time for different values of the total energy is computed as

$$\bar{T} = \frac{\sum_{i=1}^{K} T_i}{K} \tag{2.33}$$

after allowing for the relaxation of the lattice for a time of at least $t = 20$ time units. The results of section 2.5.2 indicate that this time is sufficient for the system to reach relaxation. We note that K in (2.33) is the sample size or number of iterations. To obtain the error bars for each energy excitation we have an ensemble of different initial values of momenta obtained using the procedure already outlined in section 2.5.1. Each of these ensembles or realizations is evolved in time up to a time $t = 50$. The last 500 computed values of the instantaneous temperature are stored and the same is done for all the other realizations. If we have say N realizations this is done N times. The standard deviation (or the length of the error bars) for the system's temperature is computed as the standard deviation of the $500N$ temperature values.

Chapter 3

Chaotic Behavior of Graphene

3.1 Variational equations

In order to investigate the chaotic behavior of the graphene Hamiltonian (2.5) through the computation of a chaos indicator like the mLCE, we have to follow the time evolution of a deviation vector from an orbit along with the evolution of the orbit itself. The evolution of the deviation vector is governed by the variational equations (1.17), which take the form

$$\begin{aligned} \dot{\delta x}_{(i,j)} &= \frac{\delta p_{x_{(i,j)}}}{m_c} \\ \dot{\delta y}_{(i,j)} &= \frac{\delta p_{y_{(i,j)}}}{m_c} \\ \dot{\delta p}_{x_{(i,j)}} &= \dot{\delta p}^{M}_{x_{(i,j)}} + \dot{\delta p}^{A}_{x_{(i,j)}} \\ \dot{\delta p}_{y_{(i,j)}} &= \dot{\delta p}^{M}_{y_{(i,j)}} + \dot{\delta p}^{A}_{y_{(i,j)}}, \end{aligned} \tag{3.1}$$

for Hamiltonian (2.5), with $i = 1, 2, \ldots, M$, $j = 1, 2, \ldots, N$.

Due to the complexity in writing explicitly the equations for $\dot{\delta p}_{x_{(i,j)}}$ and $\dot{\delta p}_{y_{(i,j)}}$ in (3.1), we separate the contribution to these equations from the Morse potential (2.3) and the bending potential (2.4) and denote them $\dot{\delta p}^{M}$ and $\dot{\delta p}^{A}$ respectively.

In what follows we will keep the already used convention of writing in blue the parts of equations related to the $i + j$ =even case and in red the ones corresponding to $i + j$ =odd. We note that in the subsequent sections we present in great detail the derivation of the variational equation related to $\dot{\delta p}_{x_{(i,j)}}$. Analogous results hold for the $\dot{\delta p}_{y_{(i,j)}}$equation.

3.1.1 The part of the variational equations related to the Morse potential

The part of the variational equation of $\dot{\delta p}_{x_{(i,j)}}$ related to the Morse potential is given by the following equation

$$\begin{aligned}
\dot{\delta p}^{M}_{x_{(i,j)}} = & \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{M_A}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{M_A}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} \\
& + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{M_A}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{M_A}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} \\
& + \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{M_B}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{M_B}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} \\
& + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{M_B}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{M_B}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} \\
& + \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i-1,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta x_{(i-1,j)} \\
& + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i-1,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta y_{(i-1,j)} \\
& + \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i+1,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta x_{(i+1,j)} \\
& + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i+1,j)}}\left(\dot{p}^{M_C}_{x_{(i,j)}}\right)\delta y_{(i+1,j)}.
\end{aligned} \tag{3.2}$$

In order to obtain this equation we expressed the corresponding equation of motion (2.14) as

$$\dot{p}^{M}_{x_{(i,j)}} = \dot{p}^{M_A}_{x_{(i,j)}} + \dot{p}^{M_B}_{x_{(i,j)}} + \dot{p}^{M_C}_{x_{(i,j)}} + \dot{p}^{M_C}_{x_{(i,j)}}, \tag{3.3}$$

where indices A, B and C correspond to the interaction between atoms at positions A) (i,j) and $(i,j+1)$, B) (i,j) and $(i,j-1)$, and C) (i,j) and $(i-1,j)$, or (i,j) and $(i+1,j)$. Then the part of the variational equation of $\dot{\delta p}_{x_{(i,j)}}$ related to the Morse potential (3.2) can be written as

$$\dot{\delta p}^{M}_{x_{(i,j)}} = \dot{\delta p}^{M_A}_{x_{(i,j)}} + \dot{\delta p}^{M_B}_{x_{(i,j)}} + \dot{\delta p}^{M_C}_{x_{(i,j)}}. \tag{3.4}$$

Let us now see in more detail each one of these terms. The part of the variational equation related

to the Morse potential between atoms at positions (i,j) and $(i,j+1)$ can be written as

$$
\begin{aligned}
\dot{\delta p}^{M_A}_{x_{(i,j)}} = {} & \frac{\partial}{\partial x_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j)}}\left([x_{(i,j)}-x_{(i,j+1)}]\right)\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]\frac{\partial}{\partial x_{(i,j)}}\left(\frac{1}{r^{(i,j+1)}_{(i,j)}}\right)\delta x_{(i,j)} \\
& +\frac{\partial}{\partial x_{(i,j+1)}}\left(-2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j+1)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j+1)}}\left([x_{(i,j)}-x_{(i,j+1)}]\right)\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j+1)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{1}{r^{(i,j+1)}_{(i,j)}}\right)\delta x_{(i,j+1)} \\
& +\frac{\partial}{\partial y_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j)}}\left([x_{(i,j)}-x_{(i,j+1)}]\right)\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{r^{(i,j+1)}_{(i,j)}}\right)\delta y_{(i,j)} \\
& +\frac{\partial}{\partial y_{(i,j+1)}}\left(-2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta y_{(i,j+1)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j+1)}}\left([x_{(i,j)}-x_{(i,j+1)}]\right)\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta y_{(i,j+1)} \\
& -2D\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{1}{r^{(i,j+1)}_{(i,j)}}\right)\delta y_{(i,j+1)},
\end{aligned}
\tag{3.5}
$$

which leads to the expression

$$
\begin{aligned}
\dot{\delta p}^{M_A}_{x_{(i,j)}} = &-\left(2Da^2\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&+2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^3}\delta x_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta x_{(i,j+1)}\\
&+\left(2Da^2e^{-2a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta x_{(i,j+1)}\\
&-2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i,j+1)}_{(i,j)}}\delta x_{(i,j+1)}\\
&+2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}]^2\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^3}\delta x_{(i,j+1)}\\
&-\left(2Da^2\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}][y_{(i,j)}-y_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}][y_{(i,j)}-y_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}][y_{(i,j)}-y_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^3}\delta y_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}][y_{(i,j)}-y_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta y_{(i,j+1)}\\
&+\left(2Da^2e^{-2a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}\right)[y_{(i,j)}-y_{(i,j+1)}][x_{(i,j)}-x_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^2}\delta y_{(i,j+1)}\\
&+2Da\left[e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j+1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j+1)}][y_{(i,j)}-y_{(i,j+1)}]\frac{1}{\left(r^{(i,j+1)}_{(i,j)}\right)^3}\delta y_{(i,j+1)}.
\end{aligned}
\tag{3.6}
$$

The part of the variational equation related to the Morse potential between atoms at positions (i,j)

and $(i, j-1)$ can be written as

$$\begin{aligned}
\dot{\delta p}^{M_B}_{x_{(i,j)}} = &\frac{\partial}{\partial x_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j)}}\left([x_{(i,j)}-x_{(i,j-1)}]\right)\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]\frac{\partial}{\partial x_{(i,j)}}\left(\frac{1}{r^{(i,j-1)}_{(i,j)}}\right)\delta x_{(i,j)}\\
&+\frac{\partial}{\partial x_{(i,j-1)}}\left(-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j-1)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j-1)}}\left([x_{(i,j)}-x_{(i,j-1)}]\right)\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j-1)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]\frac{\partial}{\partial x_{(i,j-1)}}\left(\frac{1}{r^{(i,j-1)}_{(i,j)}}\right)\delta x_{(i,j-1)}\\
&+\frac{\partial}{\partial y_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta y_{(i,j)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j)}}\left([x_{(i,j)}-x_{(i,j-1)}]\right)\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta y_{(i,j)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{r^{(i,j-1)}_{(i,j)}}\right)\delta y_{(i,j)}\\
&+\frac{\partial}{\partial y_{(i,j-1)}}\left(-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta y_{(i,j-1)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j-1)}}\left([x_{(i,j)}-x_{(i,j-1)}]\right)\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta y_{(i,j-1)}\\
&-2D\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]\frac{\partial}{\partial y_{(i,j-1)}}\left(\frac{1}{r^{(i,j-1)}_{(i,j)}}\right)\delta y_{(i,j-1)},
\end{aligned} \tag{3.7}$$

which leads to the expression

$$
\begin{aligned}
\dot{\delta p}^{M_B}_{x_{(i,j)}} = &-\left(2Da^2\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&+2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^3}\delta x_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta x_{(i,j-1)}\\
&+\left(2Da^2e^{-2a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta x_{(i,j-1)}\\
&-2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i,j-1)}_{(i,j)}}\delta x_{(i,j-1)}\\
&+2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}]^2\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^3}\delta x_{(i,j-1)}\\
&-\left(2Da^2\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}][y_{(i,j)}-y_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}][y_{(i,j)}-y_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}][y_{(i,j)}-y_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^3}\delta y_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j-1)}][y_{(i,j)}-y_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta y_{(i,j-1)}\\
&+\left(2Da^2e^{-2a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}\right)[y_{(i,j)}-y_{(i,j-1)}][x_{(i,j)}-x_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^2}\delta y_{(i,j-1)}\\
&+2Da\left[e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i,j-1)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i,j-1)}][y_{(i,j)}-y_{(i,j-1)}]\frac{1}{\left(r^{(i,j-1)}_{(i,j)}\right)^3}\delta y_{(i,j-1)}.
\end{aligned}
\tag{3.8}
$$

Finally, the parts related to the interaction between atoms at (i,j) and $(i-1,j)$, and atoms at (i,j)

and $(i+1,j)$ have the forms

$$
\begin{aligned}
\dot{\delta p}^{M_C}_{x_{(i,j)}} = & \frac{\partial}{\partial x_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j)}}\left([x_{(i,j)}-x_{(i-1,j)}]\right)\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]\frac{\partial}{\partial x_{(i,j)}}\left(\frac{1}{r^{(i-1,j)}_{(i,j)}}\right)\delta x_{(i,j)} \\
& +\frac{\partial}{\partial x_{(i-1,j)}}\left(-2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i-1,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i-1,j)}}\left([x_{(i,j)}-x_{(i-1,j)}]\right)\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i-1,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]\frac{\partial}{\partial x_{(i-1,j)}}\left(\frac{1}{r^{(i-1,j)}_{(i,j)}}\right)\delta x_{(i-1,j)} \\
& +\frac{\partial}{\partial y_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j)}}\left([x_{(i,j)}-x_{(i-1,j)}]\right)\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{r^{(i-1,j)}_{(i,j)}}\right)\delta y_{(i,j)} \\
& +\frac{\partial}{\partial y_{(i-1,j)}}\left(-2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta y_{(i-1,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i-1,j)}}\left([x_{(i,j)}-x_{(i-1,j)}]\right)\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta y_{(i-1,j)} \\
& -2D\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]\frac{\partial}{\partial y_{(i-1,j)}}\left(\frac{1}{r^{(i-1,j)}_{(i,j)}}\right)\delta y_{(i-1,j)},
\end{aligned}
\tag{3.9}
$$

and

$$
\begin{aligned}
\dot{\delta p}^{M_C}_{x_{(i,j)}} = {} & \frac{\partial}{\partial x_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i,j+1)}]\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i,j)}}\left([x_{(i,j)}-x_{(i+1,j)}]\right)\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]\frac{\partial}{\partial x_{(i,j)}}\left(\frac{1}{r^{(i+1,j)}_{(i,j)}}\right)\delta x_{(i,j)} \\
& +\frac{\partial}{\partial x_{(i+1,j)}}\left(-2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i+1,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial x_{(i+1,j)}}\left([x_{(i,j)}-x_{(i+1,j)}]\right)\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i+1,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]\frac{\partial}{\partial x_{(i+1,j)}}\left(\frac{1}{r^{(i+1,j)}_{(i,j)}}\right)\delta x_{(i+1,j)} \\
& +\frac{\partial}{\partial y_{(i,j)}}\left(-2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i,j)}}\left([x_{(i,j)}-x_{(i+1,j)}]\right)\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta y_{(i,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{r^{(i+1,j)}_{(i,j)}}\right)\delta y_{(i,j)} \\
& +\frac{\partial}{\partial y_{(i+1,j)}}\left(-2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta y_{(i+1,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{\partial}{\partial y_{(i+1,j)}}\left([x_{(i,j)}-x_{(i+1,j)}]\right)\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta y_{(i+1,j)} \\
& -2D\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right](-a)e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]\frac{\partial}{\partial y_{(i+1,j)}}\left(\frac{1}{r^{(i+1,j)}_{(i,j)}}\right)\delta y_{(i+1,j)},
\end{aligned}
\tag{3.10}
$$

leading respectively to

$$
\begin{aligned}
\dot{\delta p}^{M_C}_{x_{(i,j)}} = &-\left(2Da^2\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&+2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^3}\delta x_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta x_{(i-1,j)}\\
&+\left(2Da^2e^{-2a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta x_{(i-1,j)}\\
&-2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i-1,j)}_{(i,j)}}\delta x_{(i-1,j)}\\
&+2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}]^2\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^3}\delta x_{(i-1,j)}\\
&-\left(2Da^2\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}][y_{(i,j)}-y_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}][y_{(i,j)}-y_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}][y_{(i,j)}-y_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^3}\delta y_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i-1,j)}][y_{(i,j)}-y_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta y_{(i-1,j)}\\
&+\left(2Da^2e^{-2a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}\right)[y_{(i,j)}-y_{(i-1,j)}][x_{(i,j)}-x_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^2}\delta y_{(i-1,j)}\\
&+2Da\left[e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i-1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i-1,j)}][y_{(i,j)}-y_{(i-1,j)}]\frac{1}{\left(r^{(i-1,j)}_{(i,j)}\right)^3}\delta y_{(i-1,j)},
\end{aligned}
\tag{3.11}
$$

and

$$
\begin{aligned}
\dot{\delta p}^{M_C}_{x_{(i,j)}} = &-\left(2Da^2\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta x_{(i,j)}\\
&+2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^3}\delta x_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta x_{(i+1,j)}\\
&+\left(2Da^2e^{-2a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta x_{(i+1,j)}\\
&-2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\frac{1}{r^{(i+1,j)}_{(i,j)}}\delta x_{(i+1,j)}\\
&+2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}]^2\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^3}\delta x_{(i+1,j)}\\
&-\left(2Da^2\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}][y_{(i,j)}-y_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-\left(2Da^2e^{-2a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}][y_{(i,j)}-y_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta y_{(i,j)}\\
&-2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}][y_{(i,j)}-y_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^3}\delta y_{(i,j)}\\
&+\left(2Da^2\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[x_{(i,j)}-x_{(i+1,j)}][y_{(i,j)}-y_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta y_{(i+1,j)}\\
&+\left(2Da^2e^{-2a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}\right)[y_{(i,j)}-y_{(i+1,j)}][x_{(i,j)}-x_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^2}\delta y_{(i+1,j)}\\
&+2Da\left[e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}-1\right]e^{-a\left(r^{(i+1,j)}_{(i,j)}-r_0\right)}[x_{(i,j)}-x_{(i+1,j)}][y_{(i,j)}-y_{(i+1,j)}]\frac{1}{\left(r^{(i+1,j)}_{(i,j)}\right)^3}\delta y_{(i+1,j)}.
\end{aligned}
\tag{3.12}
$$

3.1.2 The part of the variational equations related to the bending potential

Let us now discuss the contribution of the bending potential (2.4) on the variational equation related to $\delta\dot{p}_{x_{(i,j)}}$. The angles whose vertices are at site (i,j) are ${}_{(i,j)}\phi^{(i,j-1)}_{(i-1,j)}$, ${}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}$, ${}_{(i,j)}\phi^{(i-1,j)}_{(i,j+1)}$ and ${}_{(i,j)}\phi^{(i,j+1)}_{(i+1,j)}$, ${}_{(i,j)}\phi^{(i,j-1)}_{(i,j+1)}$, ${}_{(i,j)}\phi^{(i+1,j)}_{(i,j-1)}$, depending on the value of $i+j$. In order to derive the part of the variational equation related to the angles having atom (i,j) at their vertices we denote these angles by ${}_{(i,j)}\phi^{(m,n)}_{(k,l)}$. The first 3 right hand side terms of (2.24) are related to angles having the atom (i,j) at their vertices and give rise to terms denoted as $\dot{p}^{A_c}_{x_{(i,j)}}$. The remaining 6 terms of (2.24) are related to case having the atom (i,j) at the edge of an angle and give us terms denoted by $\dot{p}^{A_e}_{x_{(i,j)}}$. Equation (2.24) can be written as

$$\dot{p}^{A}_{x_{(i,j)}} = \dot{p}^{A_c}_{x_{(i,j)}} + \dot{p}^{A_e}_{x_{(i,j)}}, \tag{3.13}$$

where

$$\dot{p}^{A_c}_{x_{(i,j)}} = \dot{p}^{A_{c_1}}_{x_{(i,j)}} + \dot{p}^{A_{c_2}}_{x_{(i,j)}} + \dot{p}^{A_{c_3}}_{x_{(i,j)}} + \dot{p}^{A_{c_1}}_{x_{(i,j)}} + \dot{p}^{A_{c_2}}_{x_{(i,j)}} + \dot{p}^{A_{c_3}}_{x_{(i,j)}}, \tag{3.14}$$

and

$$\begin{aligned}\dot{p}^{A_e}_{x_{(i,j)}} &= \dot{p}^{A_{e_1}}_{x_{(i,j)}} + \dot{p}^{A_{e_2}}_{x_{(i,j)}} + \dot{p}^{A_{e_3}}_{x_{(i,j)}} + \dot{p}^{A_{e_4}}_{x_{(i,j)}} + \dot{p}^{A_{e_5}}_{x_{(i,j)}} + \dot{p}^{A_{e_6}}_{x_{(i,j)}} \\ &+\dot{p}^{A_{e_1}}_{x_{(i,j)}} + \dot{p}^{A_{e_2}}_{x_{(i,j)}} + \dot{p}^{A_{e_3}}_{x_{(i,j)}} + \dot{p}^{A_{e_4}}_{x_{(i,j)}} + \dot{p}^{A_{e_5}}_{x_{(i,j)}} + \dot{p}^{A_{e_6}}_{x_{(i,j)}}.\end{aligned} \tag{3.15}$$

Then the part of the variational equation related to A_c is

$$\begin{aligned}\delta\dot{p}^{A_c}_{x_{(i,j)}} &= \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} \\ &+\frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{c_1}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} \\ &+\frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} + \frac{\partial}{\partial x_{(i-1,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i-1,j)} \\ &+\frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} + \frac{\partial}{\partial y_{(i-1,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i-1,j)} \\ &+\frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i-1,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i-1,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} \\ &+\frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i-1,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i-1,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} \\ &+\frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i+1,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta x_{(i+1,j)} \\ &+\frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i+1,j)}}\left(\dot{p}^{A_{c_2}}_{x_{(i,j)}}\right)\delta y_{(i+1,j)} \\ &+\frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i+1,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i+1,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} \\ &+\frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i+1,j)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i+1,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{c_3}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)}.\end{aligned} \tag{3.16}$$

In order to explicitly compute these terms we have to plug in the equations of motion for the appropriate angles, something which will lead to quite complicated formulas. To illustrate how the variational equations are derived we use as an example the angle ${}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}$. This angle has its vertex at (i,j) and its derivatives are

$$\delta \dot{p}^{A_{c_1}}_{x_{(i,j)}} = \left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big] - d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]^2 \right) \frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial x_{(i,j)}} \left(\frac{\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[x_{(i,j)} - x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)}} - \frac{\mathbf{r}^{(i,j+1)}_{(i,j)} \cdot \mathbf{r}^{(i,j-1)}_{(i,j)} \left[\Big[x_{(i,j)} - x_{(i,j-1)}\Big] \left(r^{(i,j+1)}_{(i,j)}\right)^2 + \Big[x_{(i,j)} - x_{(i,j+1)}\Big] \left(r^{(i,j-1)}_{(i,j)}\right)^2 \right]}{\left[r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)} \right]^3} \right) \delta x_{(i,j)}$$

$$+ \left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big] - d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]^2 \right) \frac{\partial}{\partial x_{(i,j)}} \left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)} \right)$$

$$\left(\frac{\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[x_{(i,j)} - x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)}} - \frac{\mathbf{r}^{(i,j+1)}_{(i,j)} \cdot \mathbf{r}^{(i,j-1)}_{(i,j)} \left[\Big[x_{(i,j)} - x_{(i,j-1)}\Big] \left(r^{(i,j+1)}_{(i,j)}\right)^2 + \Big[x_{(i,j)} - x_{(i,j+1)}\Big] \left(r^{(i,j-1)}_{(i,j)}\right)^2 \right]}{\left[r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)} \right]^3} \right) \delta x_{(i,j)}$$

$$+ \frac{\partial}{\partial x_{(i,j)}} \left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big] - d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]^2 \right) \frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[x_{(i,j)} - x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)}} - \frac{\mathbf{r}^{(i,j+1)}_{(i,j)} \cdot \mathbf{r}^{(i,j-1)}_{(i,j)} \left[\Big[x_{(i,j)} - x_{(i,j-1)}\Big] \left(r^{(i,j+1)}_{(i,j)}\right)^2 + \Big[x_{(i,j)} - x_{(i,j+1)}\Big] \left(r^{(i,j-1)}_{(i,j)}\right)^2 \right]}{\left[r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)} \right]^3} \right) \delta x_{(i,j)}$$

$$+ \left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big] - d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]^2 \right) \frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial x_{(i,j-1)}} \left(\frac{\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[x_{(i,j)} - x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)}} - \frac{\mathbf{r}^{(i,j+1)}_{(i,j)} \cdot \mathbf{r}^{(i,j-1)}_{(i,j)} \left[\Big[x_{(i,j)} - x_{(i,j-1)}\Big] \left(r^{(i,j+1)}_{(i,j)}\right)^2 + \Big[x_{(i,j)} - x_{(i,j+1)}\Big] \left(r^{(i,j-1)}_{(i,j)}\right)^2 \right]}{\left[r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)} \right]^3} \right) \delta x_{(i,j-1)}$$

$$+ \left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big] - d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]^2 \right) \frac{\partial}{\partial x_{(i,j-1)}} \left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)} \right)$$

$$\left(\frac{\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[x_{(i,j)} - x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)}} - \frac{\mathbf{r}^{(i,j+1)}_{(i,j)} \cdot \mathbf{r}^{(i,j-1)}_{(i,j)} \left[\Big[x_{(i,j)} - x_{(i,j-1)}\Big] \left(r^{(i,j+1)}_{(i,j)}\right)^2 + \Big[x_{(i,j)} - x_{(i,j+1)}\Big] \left(r^{(i,j-1)}_{(i,j)}\right)^2 \right]}{\left[r^{(i,j+1)}_{(i,j)} r^{(i,j-1)}_{(i,j)} \right]^3} \right) \delta x_{(i,j-1)}$$

$$+\frac{\partial}{\partial x_{(i,j-1)}}\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta x_{(i,j-1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta x_{(i,j+1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)$$

$$\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta x_{(i,j+1)}$$

$$+\frac{\partial}{\partial x_{(i,j+1)}}\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta x_{(i,j+1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial y_{(i,j)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)$$

$$\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j)}$$

$$+\frac{\partial}{\partial y_{(i,j)}}\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial y_{(i,j-1)}}\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j-1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i,j-1)}}\left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)$$

$$\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j-1)}$$

$$+\frac{\partial}{\partial y_{(i,j-1)}}\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j-1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j+1)}$$

$$+\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)$$

$$\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j+1)}$$

$$+\frac{\partial}{\partial y_{(i,j+1)}}\left(d\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]-d'\Big[_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}$$

$$\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r^{(i,j-1)}_{(i,j)}\right)^2\right]}{\left[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\right]^3}\right)\delta y_{(i,j+1)}\Bigg].$$

Let us now see how we can evaluate all the partial derivatives appearing in the previous equation. Using the fact that

$$\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)} = \frac{\sqrt{\Big[x_{(i,j)}-x_{(i,j-1)}\Big]^2+\Big[y_{(i,j)}-y_{(i,j-1)}\Big]^2}\sqrt{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]^2+\Big[y_{(i,j)}-y_{(i,j+1)}\Big]^2}}{\sqrt{\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2}}, \tag{3.17}$$

we get

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial x_{(i,j)}} = \frac{\Big[x_{(i,j)}-x_{(i,j-1)}\Big]r_2}{r_1\|\mathbf{r}_1\times\mathbf{r}_2\|}+\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]r_1}{r_2\|\mathbf{r}_1\times\mathbf{r}_2\|} \\ -\frac{r_1r_2\Big(\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\Big(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\Big)^{\frac{3}{2}}}, \tag{3.18}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial x_{(i,j-1)}} = -\frac{\Big[x_{(i,j)}-x_{(i,j-1)}\Big]r_2}{r_1\|\mathbf{r}_1\times\mathbf{r}_2\|}+\frac{r_1r_2\Big[y_{(i,j)}-y_{(i,j+1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\Big(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\Big)^{\frac{3}{2}}}, \tag{3.19}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial x_{(i,j+1)}} = -\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]r_1}{r_2\|\mathbf{r}_1\times\mathbf{r}_2\|}-\frac{r_1r_2\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\Big(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\Big)^{\frac{3}{2}}}, \tag{3.20}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial y_{(i,j)}} = \frac{\Big[y_{(i,j)}-y_{(i,j-1)}\Big]r_2}{r_1\|\mathbf{r}_1\times\mathbf{r}_2\|}+\frac{\Big[y_{(i,j)}-y_{(i,j+1)}\Big]r_1}{r_2\|\mathbf{r}_1\times\mathbf{r}_2\|} \\ -\frac{r_1r_2\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\Big(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\Big)^{\frac{3}{2}}}, \tag{3.21}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial y_{(i,j-1)}}=-\frac{\Big[y_{(i,j)}-y_{(i,j-1)}\Big]r_2}{r_1\|\mathbf{r}_1\times\mathbf{r}_2\|}-\frac{r_1r_2\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\right)^{\frac{3}{2}}} \tag{3.22}$$

and

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\right)}\right)}{\partial y_{(i,j+1)}}=-\frac{\Big[y_{(i,j)}-y_{(i,j+1)}\Big]r_1}{r_2\|\mathbf{r}_1\times\mathbf{r}_2\|}+\frac{r_1r_2\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]-\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big)^2\right)^{\frac{3}{2}}}. \tag{3.23}$$

We also have the following derivatives

$$\begin{aligned}
&\frac{\partial}{\partial x_{(i,j)}}\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\Big[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\Big]}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\Big]^3}\right)=\\
&=\frac{2}{r_1r_2}-\frac{2\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)}{r_1^3r_2}-\frac{2\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)}{r_1r_2^3}\\
&-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1^3r_2}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1r_2^3}+\frac{2\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1^3r_2^3}+\frac{3\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big)\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1^5r_2}\\
&+\frac{3\Big(\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1r_2^5}
\end{aligned} \tag{3.24}$$

$$\begin{aligned}
&\frac{\partial}{\partial x_{(i,j-1)}}\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\Big[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r^{(i,j+1)}_{(i,j)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r^{(i,j-1)}_{(i,j)}\right)^2\Big]}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j-1)}_{(i,j)}\Big]^3}\right)=\\
&=\frac{-1}{r_1r_2}+\frac{\Big(\Big[x_{(i,j)}-x_{(i,j-1)}\Big]+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big)\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r_1^3r_2}+\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1^3r_2}+\frac{\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[x_{(i,j)}-x_{(i,j+1)}\Big]}{r_1^3r_2}\\
&-\frac{3\Big[x_{(i,j)}-x_{(i,j-1)}\Big]^2\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}}{r_1^5r_2}+\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]^2}{r_1r_2^3}-\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j-1)}_{(i,j)}\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[x_{(i,j)}-x_{(i,j+1)}\Big]}{r_1^3r_2^3},
\end{aligned} \tag{3.25}$$

$$\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}}-\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r_{(i,j)}^{(i,j+1)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r_{(i,j)}^{(i,j-1)}\right)^2\right]}{\left[r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}\right]^3}\right)=$$

$$=\frac{-1}{r_1r_2}+\frac{\left(\left[x_{(i,j)}-x_{(i,j-1)}\right]+\left[x_{(i,j)}-x_{(i,j+1)}\right]\right)\left[x_{(i,j)}-x_{(i,j+1)}\right]}{r_1r_2^3}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}}{r_1r_2^3}+\frac{\left[x_{(i,j)}-x_{(i,j-1)}\right]\left[x_{(i,j)}-x_{(i,j+1)}\right]}{r_1r_2^3} \tag{3.26}$$

$$-\frac{3\left[x_{(i,j)}-x_{(i,j+1)}\right]^2\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}}{r_1r_2^5}+\frac{\left[x_{(i,j)}-x_{(i,j-1)}\right]^2}{r_1^3r_2}-\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}\left[x_{(i,j)}-x_{(i,j-1)}\right]\left[x_{(i,j)}-x_{(i,j+1)}\right]}{r_1^3r_2^3},$$

$$\frac{\partial}{\partial y_{(i,j)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}}-\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r_{(i,j)}^{(i,j+1)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r_{(i,j)}^{(i,j-1)}\right)^2\right]}{\left[r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}\right]^3}\right)=$$

$$=\frac{-\left(\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]\right)\left[y_{(i,j)}-y_{(i,j-1)}\right]}{r_1^3r_2}-\frac{\left(\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]\right)\left[y_{(i,j)}-y_{(i,j+1)}\right]}{r_1r_2^3}$$

$$-\frac{\left(\left[y_{(i,j)}-y_{(i,j+1)}\right]+\left[y_{(i,j)}-y_{(i,j-1)}\right]\right)\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r_1^3r_2}-\frac{\left(\left[y_{(i,j)}-y_{(i,j+1)}\right]+\left[y_{(i,j)}-y_{(i,j-1)}\right]\right)\left[x_{(i,j)}-x_{(i,j+1)}\right]}{r_1r_2^3} \tag{3.27}$$

$$+\frac{3\left[x_{(i,j)}-x_{(i,j-1)}\right]\left[y_{(i,j)}-y_{(i,j-1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^5r_2}+\frac{\left[x_{(i,j)}-x_{(i,j-1)}\right]\left[y_{(i,j)}-y_{(i,j+1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^3r_2^3}$$

$$+\frac{3\left[x_{(i,j)}-x_{(i,j+1)}\right]\left[y_{(i,j)}-y_{(i,j+1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1r_2^5}+\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]\left[y_{(i,j)}-y_{(i,j-1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^3r_2^3},$$

$$\frac{\partial}{\partial y_{(i,j-1)}}\left(\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}}-\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}\left[\left[x_{(i,j)}-x_{(i,j-1)}\right]\left(r_{(i,j)}^{(i,j+1)}\right)^2+\left[x_{(i,j)}-x_{(i,j+1)}\right]\left(r_{(i,j)}^{(i,j-1)}\right)^2\right]}{\left[r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}\right]^3}\right)=$$

$$=\frac{\left(\left[x_{(i,j)}-x_{(i,j+1)}\right]+\left[x_{(i,j)}-x_{(i,j-1)}\right]\right)\left[y_{(i,j)}-y_{(i,j-1)}\right]}{r_1^3r_2}+\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]\left[y_{(i,j)}-y_{(i,j+1)}\right]}{r_1r_2^3}+\frac{\left[y_{(i,j)}-y_{(i,j+1)}\right]\left[x_{(i,j)}-x_{(i,j-1)}\right]}{r_1^3r_2} \tag{3.28}$$

$$-\frac{3\left[x_{(i,j)}-x_{(i,j-1)}\right]\left[y_{(i,j)}-y_{(i,j-1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^5r_2}-\frac{\left[x_{(i,j)}-x_{(i,j+1)}\right]\left[y_{(i,j)}-y_{(i,j-1)}\right]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^3r_2^3},$$

and

$$\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}}-\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i,j)}^{(i,j-1)}\Big[\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\left(r_{(i,j)}^{(i,j+1)}\right)^2+\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\left(r_{(i,j)}^{(i,j-1)}\right)^2\Big]}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i,j)}^{(i,j-1)}\Big]^3}\right)=$$

$$=\frac{\Big(\Big[x_{(i,j)}-x_{(i,j+1)}\Big]+\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big)\Big[y_{(i,j)}-y_{(i,j+1)}\Big]}{r_1r_2^3}+\frac{\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j-1)}\Big]}{r_1r_2^3}+\frac{\Big[y_{(i,j)}-y_{(i,j-1)}\Big]\Big[x_{(i,j)}-x_{(i,j-1)}\Big]}{r_1^3r_2} \tag{3.29}$$

$$-\frac{3\Big[x_{(i,j)}-x_{(i,j+1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1r_2^5}-\frac{\Big[x_{(i,j)}-x_{(i,j-1)}\Big]\Big[y_{(i,j)}-y_{(i,j+1)}\Big]\mathbf{r_1}\cdot\mathbf{r_2}}{r_1^3r_2^3}.$$

The final set of partial derivatives related to the atom at (i,j) being at the vertex of an angle are computed below. First of all we note that the second derivative of the bending potential with respect to angles with vertices at (i,j) is

$$\frac{\partial^2 V^A\Big[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\Big]}{\partial\Big[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}\Big]^2} = d - 2d'\Big[{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} - \phi_0\Big]. \tag{3.30}$$

Since the angle ${}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}$ can be written as

$${}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)} = \cos^{-1}\left(\frac{\Big(\Big[x_{(i,j)} - x_{(i,j-1)}\Big]\Big[x_{(i,j)} - x_{(i,j+1)}\Big] + \Big[y_{(i,j)} - y_{(i,j-1)}\Big]\Big[y_{(i,j)} - y_{(i,j+1)}\Big]\Big)}{\sqrt{\Big[x_{(i,j)} - x_{(i,j-1)}\Big]^2 + \Big[y_{(i,j)} - y_{(i,j-1)}\Big]^2}\sqrt{\Big[x_{(i,j)} - x_{(i,j+1)}\Big]^2 + \Big[y_{(i,j)} - y_{(i,j+1)}\Big]^2}}\right), \tag{3.31}$$

its partial derivatives are

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial x_{(i,j)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\Big(x_{(i,j)} - x_{(i,j-1)}\Big) + \Big(x_{(i,j)} - x_{(i,j+1)}\Big)}{r_1 r_2} - \frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(x_{(i,j)} - x_{(i,j-1)}\Big)}{r_1^3 r_2}\right) + \frac{1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(x_{(i,j)} - x_{(i,j+1)}\Big)}{r_1 r_2^3}\right), \tag{3.32}$$

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial x_{(i,j-1)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(x_{(i,j)} - x_{(i,j-1)}\Big)}{r_1^3 r_2} - \frac{\Big(x_{(i,j)} - x_{(i,j+1)}\Big)}{r_1 r_2}\right), \tag{3.33}$$

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial x_{(i,j+1)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(x_{(i,j)} - x_{(i,j+1)}\Big)}{r_1 r_2^3} - \frac{\Big(x_{(i,j)} - x_{(i,j-1)}\Big)}{r_1 r_2}\right), \tag{3.34}$$

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial y_{(i,j)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\Big(y_{(i,j)} - y_{(i,j-1)}\Big) + \Big(y_{(i,j)} - y_{(i,j+1)}\Big)}{r_1 r_2} - \frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(y_{(i,j)} - y_{(i,j-1)}\Big)}{r_1^3 r_2}\right) + \frac{1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(y_{(i,j)} - y_{(i,j+1)}\Big)}{r_1 r_2^3}\right), \tag{3.35}$$

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial y_{(i,j-1)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(y_{(i,j)} - y_{(i,j-1)}\Big)}{r_1^3 r_2} - \frac{\Big(y_{(i,j)} - y_{(i,j+1)}\Big)}{r_1 r_2}\right), \tag{3.36}$$

and

$$\frac{\partial_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}{\partial y_{(i,j+1)}} = \frac{-1}{\sin{}_{(i,j)}\phi^{(i,j+1)}_{(i,j-1)}}\left(\frac{\mathbf{r}_1\cdot\mathbf{r}_2\Big(y_{(i,j)} - y_{(i,j+1)}\Big)}{r_1 r_2^3} - \frac{\Big(y_{(i,j)} - y_{(i,j-1)}\Big)}{r_1 r_2}\right). \tag{3.37}$$

Let us now derive the part of the variational equation of $\dot{\delta p}_{x_{(i,j)}}$ related to the angles having atom (i,j) at their edges (see Figure 2.6). For these angles we note that the vector from (i,j) to (a,b) is given as

$$\vec{r}^{\,(a,b)}_{(i,j)} = (x_{(a,b)} - x_{(i,j)})\hat{\mathbf{i}} + (y_{(a,b)} - y_{(i,j)})\hat{\mathbf{j}}, \tag{3.38}$$

where (a,b) are generalized indices representing atoms at positions $(i-1,j)$, $(i,j-1)$, $(i,j+1)$ and $(i+1,j)$, $(i,j+1)$, $(i,j-1)$. Assuming in addition indices (c,d) to represent of atoms at $(i,j+2)$,

$(i-1,j+1)$, $(i-1,j-1)$, $(i,j-2)$, $(i+1,j-1)$, $(i+1,j+1)$ and $(i,j+2)$, $(i-1,j+1)$, $(i-1,j-1)$, $(i,j-2)$, $(i+1,j-1)$, $(i+1,j+1)$, the corresponding vector is

$$\vec{r}^{(a,b)}_{(c,d)} = (x_{(a,b)} - x_{(c,d)})\hat{\mathbf{i}} + (y_{(a,b)} - y_{(c,d)})\hat{\mathbf{j}}. \tag{3.39}$$

Then, the cosines of the related angles are given by

$$\cos\left({}_{(a,b)}\phi^{(c,d)}_{(i,j)}\right) = \frac{(x_{(a,b)} - x_{(i,j)})(x_{(a,b)} - x_{(c,d)}) + (y_{(a,b)} - y_{(i,j)})(y_{(a,b)} - y_{(c,d)})}{\sqrt{(x_{(a,b)} - x_{(i,j)})^2 + (y_{(a,b)} - y_{(i,j)})^2}\sqrt{(x_{(a,b)} - x_{(c,d)})^2 + (y_{(a,b)} - y_{(c,d)})^2}}, \tag{3.40}$$

and

$$\cos\left({}_{(a,b)}\phi^{(i,j)}_{(c,d)}\right) = \frac{(x_{(a,b)} - x_{(i,j)})(x_{(a,b)} - x_{(c,d)}) + (y_{(a,b)} - y_{(i,j)})(y_{(a,b)} - y_{(c,d)})}{\sqrt{(x_{(a,b)} - x_{(i,j)})^2 + (y_{(a,b)} - y_{(i,j)})^2}\sqrt{(x_{(a,b)} - x_{(c,d)})^2 + (y_{(a,b)} - y_{(c,d)})^2}}. \tag{3.41}$$

Representing vectors $\vec{r}^{(a,b)}_{(i,j)}$ and $\vec{r}^{(a,b)}_{(c,d)}$ by $\mathbf{r}_3$ and $\mathbf{r}_4$ respectively, we can write their cross product as

$$\|\mathbf{r}_3 \times \mathbf{r}_4\| = \begin{vmatrix} \hat{\mathbf{i}} & \hat{\mathbf{j}} & \hat{\mathbf{k}} \\ (x_{(a,b)} - x_{(i,j)}) & (y_{(a,b)} - y_{(i,j)}) & 0 \\ (x_{(a,b)} - x_{(c,d)}) & (y_{(a,b)} - y_{(c,d)}) & 0 \end{vmatrix} = \|\mathbf{r}_3\|\|\mathbf{r}_4\| \sin\left({}_{(a,b)}\phi^{(c,d)}_{(i,j)}\right). \tag{3.42}$$

This leads to the expressions

$$\frac{1}{\sin\left({}_{(a,b)}\phi^{(c,d)}_{(i,j)}\right)} = \frac{\sqrt{\left[x_{(a,b)} - x_{(i,j)}\right]^2 + \left[y_{(a,b)} - y_{(i,j)}\right]^2}\sqrt{\left[x_{(a,b)} - x_{(c,d)}\right]^2 + \left[y_{(a,b)} - y_{(c,d)}\right]^2}}{\sqrt{\left(\left[x_{(a,b)} - x_{(i,j)}\right]\left[y_{(a,b)} - y_{(c,d)}\right] - \left[x_{(a,b)} - x_{(c,d)}\right]\left[y_{(a,b)} - y_{(i,j)}\right]\right)^2}}, \tag{3.43}$$

and

$$\frac{1}{\sin\left({}_{(a,b)}\phi^{(i,j)}_{(c,d)}\right)} = \frac{\sqrt{\left[x_{(a,b)} - x_{(i,j)}\right]^2 + \left[y_{(a,b)} - y_{(i,j)}\right]^2}\sqrt{\left[x_{(a,b)} - x_{(c,d)}\right]^2 + \left[y_{(a,b)} - y_{(c,d)}\right]^2}}{\sqrt{\left(-\left[x_{(a,b)} - x_{(i,j)}\right]\left[y_{(a,b)} - y_{(c,d)}\right] + \left[x_{(a,b)} - x_{(c,d)}\right]\left[y_{(a,b)} - y_{(i,j)}\right]\right)^2}}. \tag{3.44}$$

The 6 angles whose vertices are at (a,b) are ${}_{(i,j+1)}\phi^{(i,j)}_{(i,j+2)}$, ${}_{(i-1,j)}\phi^{(i-1,j+1)}_{(i,j)}$, ${}_{(i-1,j)}\phi^{(i,j)}_{(i-1,j-1)}$, ${}_{(i,j-1)}\phi^{(i,j-2)}_{(i,j)}$, ${}_{(i,j-1)}\phi^{(i,j)}_{(i+1,j-1)}$, ${}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}$ and ${}_{(i,j+1)}\phi^{(i,j+2)}_{(i,j)}$, ${}_{(i+1,j)}\phi^{(i,j)}_{(i+1,j+1)}$, ${}_{(i+1,j)}\phi^{(i+1,j-1)}_{(i,j)}$, ${}_{(i,j-1)}\phi^{(i,j)}_{(i,j-2)}$, ${}_{(i,j-1)}\phi^{(i-1,j-1)}_{(i,j)}$, ${}_{(i,j+1)}\phi^{(i,j)}_{(i-1,j+1)}$.

Then the corresponding contributions to the variational equation of $\dot{\delta p}_{x_{(i,j)}}$ are obtained by the following terms:

$$\begin{aligned}\dot{\delta p}^{A_{e_1}}_{x_{(i,j)}} = {} & \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} + \frac{\partial}{\partial x_{(i+1,j+1)}}\left(\dot{p}^{A_{e_1}}_{qx_{(i,j)}}\right)\delta x_{(i+1,j+1)} \\ & + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} + \frac{\partial}{\partial y_{(i+1,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i+1,j+1)},\end{aligned} \tag{3.45}$$

$$\begin{aligned}\dot{\delta p}^{A_{e_2}}_{x_{(i,j)}} = {} & \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} + \frac{\partial}{\partial x_{(i,j+2)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j+2)} \\ & + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} + \frac{\partial}{\partial y_{(i,j+2)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j+2)},\end{aligned} \tag{3.46}$$

$$\delta\dot{p}^{A_{e_3}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i-1,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i-1,j)} + \frac{\partial}{\partial x_{(i-1,j+1)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i-1,j+1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i-1,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i-1,j)} + \frac{\partial}{\partial y_{(i-1,j+1)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i-1,j+1)}, \tag{3.47}$$

$$\delta\dot{p}^{A_{e_4}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i-1,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i-1,j)} + \frac{\partial}{\partial x_{(i-1,j-1)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i-1,j-1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i-1,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i-1,j)} + \frac{\partial}{\partial y_{(i-1,j-1)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i-1,j-1)}, \tag{3.48}$$

$$\delta\dot{p}^{A_{e_5}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i,j-2)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j-2)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i,j-2)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j-2)}, \tag{3.49}$$

and

$$\delta\dot{p}^{A_{e_6}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i+1,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i+1,j-1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i+1,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i+1,j-1)}, \tag{3.50}$$

or

$$\delta\dot{p}^{A_{e_1}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} + \frac{\partial}{\partial x_{(i-1,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta x_{(i-1,j+1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} + \frac{\partial}{\partial y_{(i-1,j+1)}}\left(\dot{p}^{A_{e_1}}_{x_{(i,j)}}\right)\delta y_{(i-1,j+1)}, \tag{3.51}$$

$$\delta\dot{p}^{A_{e_2}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j+1)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j+1)} + \frac{\partial}{\partial x_{(i,j+2)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta x_{(i,j+2)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j+1)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j+1)} + \frac{\partial}{\partial y_{(i,j+2)}}\left(\dot{p}^{A_{e_2}}_{x_{(i,j)}}\right)\delta y_{(i,j+2)}, \tag{3.52}$$

$$\delta\dot{p}^{A_{e_3}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i+1,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i+1,j)} + \frac{\partial}{\partial x_{(i+1,j+1)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta x_{(i+1,j+1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i+1,j)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i+1,j)} + \frac{\partial}{\partial y_{(i+1,j+1)}}\left(\dot{p}^{A_{e_3}}_{x_{(i,j)}}\right)\delta y_{(i+1,j+1)}, \tag{3.53}$$

$$\delta\dot{p}^{A_{e_4}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i+1,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i+1,j)} + \frac{\partial}{\partial x_{(i+1,j-1)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta x_{(i+1,j-1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i+1,j)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i+1,j)} + \frac{\partial}{\partial y_{(i+1,j-1)}}\left(\dot{p}^{A_{e_4}}_{x_{(i,j)}}\right)\delta y_{(i+1,j-1)}, \tag{3.54}$$

$$\delta\dot{p}^{A_{e_5}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i,j-2)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta x_{(i,j-2)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i,j-2)}}\left(\dot{p}^{A_{e_5}}_{x_{(i,j)}}\right)\delta y_{(i,j-2)}, \tag{3.55}$$

and

$$\delta\dot{p}^{A_{e_6}}_{x_{(i,j)}} = \frac{\partial}{\partial x_{(i,j)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i,j)} + \frac{\partial}{\partial x_{(i,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i,j-1)} + \frac{\partial}{\partial x_{(i-1,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta x_{(i-1,j-1)} + \frac{\partial}{\partial y_{(i,j)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i,j)} + \frac{\partial}{\partial y_{(i,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i,j-1)} + \frac{\partial}{\partial y_{(i-1,j-1)}}\left(\dot{p}^{A_{e_6}}_{x_{(i,j)}}\right)\delta y_{(i-1,j-1)}. \tag{3.56}$$

The variational equations along the y axis are obtained in a similar manner.

Let us now demonstrate how one cane evaluate the terms appearing in equations (3.45)–(3.56) by studying in detail one particular case, namely equation (3.45). The expansion of that equation leads

to

$$\dot{\delta p}^{A_{e_1}}_{x_{(i,j)}} = \left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\frac{\partial}{\partial x_{(i,j)}}\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{\partial}{\partial x_{(i,j)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j)}$$
$$+\frac{\partial}{\partial x_{(i,j)}}\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j+1)}$$
$$+\frac{\partial}{\partial x_{(i,j+1)}}\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\frac{\partial}{\partial x_{(i+1,j+1)}}\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i+1,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big] - d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)} - \phi_0\Big]^2\right)\frac{\partial}{\partial x_{(i+1,j+1)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)} - x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}} + \frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)} - x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta x_{(i+1,j+1)}$$

$$+\frac{\partial}{\partial x_{(i+1,j+1)}}\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta x_{(i+1,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}$$
$$\frac{\partial}{\partial y_{(i,j)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j)}$$
$$+\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i,j)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j)}$$
$$+\frac{\partial}{\partial y_{(i,j)}}\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j)}$$
$$+\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}$$
$$\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j+1)}$$
$$+\frac{\partial}{\partial y_{(i,j+1)}}\left(d\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right)}$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)\delta y_{(i,j+1)}$$

$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$\frac{\partial}{\partial y_{(i+1,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}}+\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta y_{(i+1,j+1)}$$
$$+\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]^2\right)\frac{\partial}{\partial y_{(i+1,j+1)}}\left(\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)$$
$$\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}}+\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta y_{(i+1,j+1)}$$
$$+\frac{\partial}{\partial y_{(i+1,j+1)}}\left(d\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]-d'\Big[_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}-\phi_0\Big]^2\right)\frac{1}{\sin\left(_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}$$
$$+\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}}+\frac{\mathbf{r}^{(i,j+1)}_{(i,j)}\cdot\mathbf{r}^{(i,j+1)}_{(i+1,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r^{(i,j+1)}_{(i+1,j+1)}\right)^2}{\Big[r^{(i,j+1)}_{(i,j)}r^{(i,j+1)}_{(i+1,j+1)}\Big]^3}\right)\delta y_{(i+1,j+1)}\Bigg]$$

The remaining terms related to angles having point (i,j) at their edge are obtained in a similar manner to the one presented above.

Let us now evaluate all the partial derivatives appearing in the previous equation. Since the angle $_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}$ can be written as

$$_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}=$$
$$\cos^{-1}\left(\frac{\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]+\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]\Big)}{\sqrt{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]^2+\Big[y_{(i,j+1)}-y_{(i,j)}\Big]^2}\sqrt{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]^2+\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]^2}}\right),\tag{3.57}$$

its partial derivatives are given by the following equations

$$\frac{\partial_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}{\partial x_{(i,j)}}=\frac{-1}{\sin{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}\left(\frac{-\Big(x_{(i,j+1)}-x_{(i+1,j+1)}\Big)}{r_3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big(x_{(i,j+1)}-x_{(i,j)}\Big)}{r_3^3r_4}\right),\tag{3.58}$$
$$\frac{\partial_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}{\partial x_{(i,j+1)}}=\frac{1}{\sin{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}\left(\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big(x_{(i,j+1)}-x_{(i+1,j+1)}\Big)}{r_3r_4^3}\right)$$
$$-\frac{1}{\sin{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}\left(\frac{\Big(x_{(i,j+1)}-x_{(i,j)}\Big)+\Big(x_{(i,j+1)}-x_{(i+1,j+1)}\Big)}{r_3r_4}-\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big(x_{(i,j+1)}-x_{(i,j)}\Big)}{r_3^3r_4}\right),\tag{3.59}$$
$$\frac{\partial_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}{\partial x_{(i+1,j+1)}}=\frac{-1}{\sin{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}\left(\frac{-\Big(x_{(i,j+1)}-x_{(i,j)}\Big)}{r_3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big(x_{(i,j+1)}-x_{(i+1,j+1)}\Big)}{r_3r_4^3}\right),\tag{3.60}$$
$$\frac{\partial_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}{\partial y_{(i,j)}}=\frac{-1}{\sin{}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}}\left(\frac{-\Big(y_{(i,j+1)}-y_{(i+1,j+1)}\Big)}{r_3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big(y_{(i,j+1)}-y_{(i,j)}\Big)}{r_3^3r_4}\right),\tag{3.61}$$

$$\frac{\partial_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}}{\partial y_{(i,j+1)}} = \frac{1}{\sin{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}}\left(\frac{\mathbf{r}_3\cdot\mathbf{r}_4\left(y_{(i,j+1)}-y_{(i+1,j+1)}\right)}{r_3r_4^3}\right)$$
$$-\frac{1}{\sin{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}}\left(\frac{\left(y_{(i,j+1)}-y_{(i,j)}\right)+\left(y_{(i,j+1)}-y_{(i+1,j+1)}\right)}{r_3r_4}-\frac{\mathbf{r}_3\cdot\mathbf{r}_4\left(y_{(i,j+1)}-y_{(i,j)}\right)}{r_3^3r_4}\right), \tag{3.62}$$

and

$$\frac{\partial_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}}{\partial y_{(i+1,j+1)}} = \frac{-1}{\sin{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}}\left(\frac{-\left(y_{(i,j+1)}-y_{(i,j)}\right)}{r_3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\left(y_{(i,j+1)}-y_{(i+1,j+1)}\right)}{r_3r_4^3}\right). \tag{3.63}$$

As in equation (3.30) the second derivatives of the bending potential are

$$\frac{\partial^2 V^A\left[{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right]}{\partial\left[{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}\right]^2} = d-2d'\left[{}_{(i,j+1)}\phi_{(i,j)}^{(i+1,j+1)}-\phi_0\right], \tag{3.64}$$

and

$$\frac{\partial^2 V^A\left[{}_{(i,j+1)}\phi_{(i+1,j+1)}^{(i,j)}\right]}{\partial\left[{}_{(i,j+1)}\phi_{(i+1,j+1)}^{(i,j)}\right]^2} = d-2d'\left[{}_{(i,j+1)}\phi_{(i+1,j+1)}^{(i,j)}-\phi_0\right]. \tag{3.65}$$

From equation (3.43) we get

$$\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)} = \frac{\sqrt{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]^2+\Big[y_{(i,j+1)}-y_{(i,j)}\Big]^2}\sqrt{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]^2+\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]^2}}{\sqrt{\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2}}. \tag{3.66}$$

Thus

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial x_{(i,j)}} = \frac{r_3r_4\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}} - \frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]r_4}{r_3\|\mathbf{r}_3\times\mathbf{r}_4\|}, \tag{3.67}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial x_{(i,j+1)}} = \frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]r_4}{r_3\|\mathbf{r}_3\times\mathbf{r}_4\|} + \frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]r_3}{r_4\|\mathbf{r}_3\times\mathbf{r}_4\|} + \frac{r_3r_4\Big(\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}}, \tag{3.68}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial x_{(i+1,j+1)}} = -\frac{r_3r_4\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}} - \frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]r_3}{r_4\|\mathbf{r}_3\times\mathbf{r}_4\|}, \tag{3.69}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial y_{(i,j)}} = \frac{r_3 r_4\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}} - \frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]r_4}{r_3\|\mathbf{r}_3\times\mathbf{r}_4\|}, \tag{3.70}$$

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial y_{(i,j+1)}} = \frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]r_4}{r_3\|\mathbf{r}_3\times\mathbf{r}_4\|} + \frac{\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]r_3}{r_4\|\mathbf{r}_3\times\mathbf{r}_4\|} + \frac{r_3 r_4\left(\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\right)\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}}, \tag{3.71}$$

and

$$\frac{\partial\left(\frac{1}{\sin\left({}_{(i,j+1)}\phi^{(i+1,j+1)}_{(i,j)}\right)}\right)}{\partial y_{(i+1,j+1)}} = -\frac{r_3 r_4\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{\left(\Big(\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)^2\right)^{\frac{3}{2}}} - \frac{\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]r_3}{r_4\|\mathbf{r}_3\times\mathbf{r}_4\|}. \tag{3.72}$$

The remaining partial derivatives are

$$\frac{\partial}{\partial x_{(i,j)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(r_{(i+1,j+1)}^{(i,j+1)}\Big)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{3\Big[x_{(i,j+1)}-x_{(i,j)}\Big]^2\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}-\frac{\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4}-\frac{2\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_3^3r_4}, \tag{3.73}$$

$$\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(r_{(i+1,j+1)}^{(i,j+1)}\Big)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]^2}{r_3r_4^3}-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3}$$
$$+\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i,j)}\Big]}{r_3^3r_4}-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]+\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big)}{r_3^3r_4}$$
$$+\frac{\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4}-\frac{3\Big[x_{(i,j+1)}-x_{(i,j)}\Big]^2\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}-\frac{1}{r_3^3r_4}, \tag{3.74}$$

$$\frac{\partial}{\partial x_{(i+1,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(r_{(i+1,j+1)}^{(i,j+1)}\Big)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{1}{r_3r_4}-\frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]^2}{r_3r_4^3}-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]^2}{r_3^3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_3^3r_4^3}, \tag{3.75}$$

$$\frac{\partial}{\partial y_{(i,j)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(r_{(i+1,j+1)}^{(i,j+1)}\Big)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]}{r_3^3r_4}-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3^3r_4}$$
$$+\frac{3\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}, \tag{3.76}$$

$$\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big(r_{(i+1,j+1)}^{(i,j+1)}\Big)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]}{r_3^3r_4}+\frac{\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3r_4^3}$$
$$+\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3^3r_4}$$
$$+\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]}{r_3^3r_4}-\frac{3\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}$$
$$-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3}, \tag{3.77}$$

$$\frac{\partial}{\partial y_{(i+1,j+1)}}\left(\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{-\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3r_4^3}-\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]}{r_3^3r_4} \tag{3.78}$$
$$+\frac{\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3},$$

$$\frac{\partial}{\partial y_{(i,j)}}\left(\frac{-\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{3\Big[y_{(i,j+1)}-y_{(i,j)}\Big]^2\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}-\frac{\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4}-\frac{2\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3^3r_4}, \tag{3.79}$$

$$\frac{\partial}{\partial y_{(i,j+1)}}\left(\frac{-\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]^2}{r_3r_4^3}-\frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3}+\frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i,j)}\Big]}{r_3^3r_4}$$
$$-\frac{1}{r_3^3r_4}-\frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big(\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]+\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big)}{r_3^3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4}-\frac{3\Big[y_{(i,j+1)}-y_{(i,j)}\Big]^2\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}, \tag{3.80}$$

$$\frac{\partial}{\partial y_{(i+1,j+1)}}\left(\frac{-\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{1}{r_3r_4}-\frac{\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]^2}{r_3r_4^3}-\frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]^2}{r_3^3r_4}+\frac{\mathbf{r}_3\cdot\mathbf{r}_4\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_3^3r_4^3}, \tag{3.81}$$

$$\frac{\partial}{\partial x_{(i,j)}}\left(\frac{-\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\Big[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\Big]^3}\right)$$
$$=\frac{-\Big[y_{(i,j+1)}-y_{(i+1,j+1)}\Big]\Big[x_{(i,j+1)}-x_{(i,j)}\Big]}{r_3^3r_4}-\frac{\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i+1,j+1)}\Big]}{r_3^3r_4} \tag{3.82}$$
$$+\frac{3\Big[y_{(i,j+1)}-y_{(i,j)}\Big]\Big[x_{(i,j+1)}-x_{(i,j)}\Big]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4},$$

$$\frac{\partial}{\partial x_{(i,j+1)}}\left(\frac{-\left[y_{(i,j+1)}-y_{(i+1,j+1)}\right]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\left[y_{(i,j+1)}-y_{(i,j)}\right]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\left[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\right]^3}\right)$$
$$=\frac{\left[y_{(i,j+1)}-y_{(i+1,j+1)}\right]\left[x_{(i,j+1)}-x_{(i,j)}\right]}{r_3^3r_4}+\frac{\left[y_{(i,j+1)}-y_{(i+1,j+1)}\right]\left[x_{(i,j+1)}-x_{(i+1,j+1)}\right]}{r_3r_4^3}$$
$$+\frac{\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i+1,j+1)}\right]}{r_3^3r_4}+\frac{\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i,j)}\right]}{r_3^3r_4} \tag{3.83}$$
$$-\frac{3\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i,j)}\right]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^5r_4}-\frac{\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i+1,j+1)}\right]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3},$$

and

$$\frac{\partial}{\partial x_{(i+1,j+1)}}\left(\frac{-\left[y_{(i,j+1)}-y_{(i+1,j+1)}\right]}{r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}}+\frac{\mathbf{r}_{(i,j)}^{(i,j+1)}\cdot\mathbf{r}_{(i+1,j+1)}^{(i,j+1)}\left[y_{(i,j+1)}-y_{(i,j)}\right]\left(r_{(i+1,j+1)}^{(i,j+1)}\right)^2}{\left[r_{(i,j)}^{(i,j+1)}r_{(i+1,j+1)}^{(i,j+1)}\right]^3}\right)$$
$$=\frac{-\left[y_{(i,j+1)}-y_{(i+1,j+1)}\right]\left[x_{(i,j+1)}-x_{(i+1,j+1)}\right]}{r_3r_4^3}-\frac{\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i,j)}\right]}{r_3^3r_4} \tag{3.84}$$
$$+\frac{\left[y_{(i,j+1)}-y_{(i,j)}\right]\left[x_{(i,j+1)}-x_{(i+1,j+1)}\right]\mathbf{r}_3\cdot\mathbf{r}_4}{r_3^3r_4^3}.$$

3.2 Evaluation of the system's mLCE

Using the variational equation determined in Section 3.1 and the Tangent Map method presented in Section 1.6 we can now quantify the chaoticity of our graphene model by computing its mLCE (1.22).

In Figure 3.1 we present the time evolution of the mLCE for a graphene system with 12×16 atoms for some particular initial excitations of $2eV$, $10eV$ and $20eV$. The initial values of the orbit where obtained as explained earlier in Section 2.5.1. The initial values of the deviation vector were derived from a standard normal distribution which was then normalised to make the initial vector unitary as explained in Section 1.7. We note that the results presented in Figure 3.1 were obtained by using the ABA864 SI with integration time step $\tau = 0.15$. We see that as the system's total energy increases the value of the mLCE becomes larger, indicating (as expected) the increase of the system's chaoticity.

Having developed the computer code for evaluating the system's mLCE we intend to study graphene's chaotic behavior in more detail in the future.

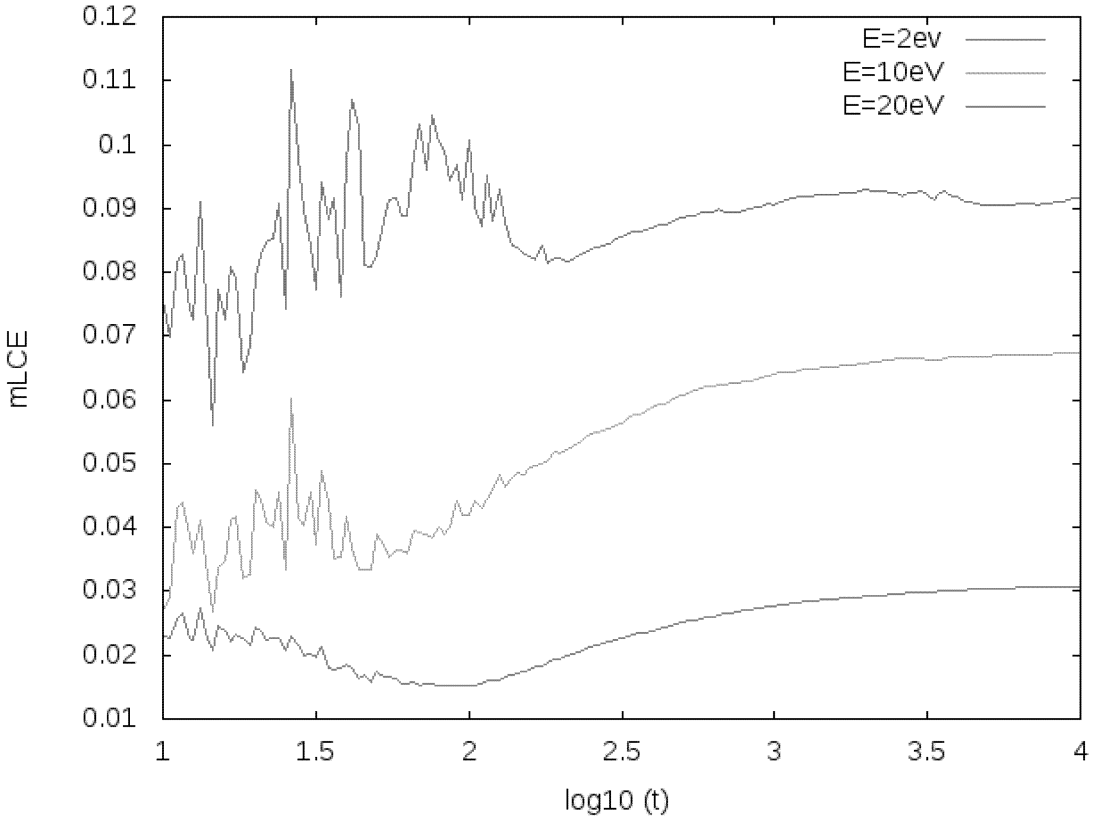

Figure 3.1: The time evolution of the mLCE of the graphene model (2.5) for individual, random excitations of total energy $2eV$, $10eV$ and $20eV$.

Chapter 4

Summary

In this thesis we considered a Hamiltonian model describing the dynamical behavior of graphene. Initially we described in detail several notions of Hamiltonian dynamics using as a toy model the Hénon-Heiles system to illustrate them. In particular we defined what is a dynamical system and more specifically a Hamiltonian one, we presented its equations of motion which govern its time evolution, as well its variational equations governing the time evolution of small perturbations from an orbit. We provided the definition of chaos and presented two techniques which allow its identification: the Poincaré surface of section and the computation of the maximum Lyapunov Characteristic Exponent. A significant part of Chapter 1 was devoted to the detailed description of a very efficient technique for integrating the Hamilton equations of motion, along with the variational equations needed for the computation of the maximum Lyapunov Characteristic Exponent: the symplectic integration method.

Then in Chapter 2 we focused our attention the dynamics of graphene. After a brief presentation of the physical properties and importance of this material, we presented in great detail a Hamiltonian system which tries to describe its dynamics. For this model we explained how one can obtain its equation of motion, a not so simple task as their derivation is rather complicated. Then we presented some numerical results obtained for this model through several numerical simulations we performed using an appropriately chosen SI scheme. We note that in the Appendix we provide the computer code we used for this task. We also explained in detail how we chose this particular SI after performing a comparison between different integration schemes.

Finally, in Chapter 3 we presented on the main topic of the thesis, i.e. the derivation of the variational equations of the particular Hamiltonian model of graphene. This was a tedious task which, to the best of our knowledge, has not been presented in the literature. Using the obtained variational equations we computed the maximum Lyapunonv Characteristic Exponent of the system for a representative initial excitation. We expect to perform a more systematic investigation of graphene's chaotic behavior in the immediate future.

Bibliography

Allen, M. (2009). Honeycomb carbon – A study of graphene. *Am. Chem. Soc.*, page 184.

Anderson, P. W. (1958). Absence of diffusion in certain random lattices. *Phys. Rev.*, 109(5):1492–1505.

Bao, H., Pan, Y., Ping, Y., Sahoo, N. G., Wu, T., Li, L., Li, J., and Gan, L. H. (2011). Chitosan-functionalized graphene oxide as a nanocarrier for drug and gene delivery. *Small*, 7(11):1569–1578.

Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J. M. (1980a). Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; a method for computing all of them. Part 1: Theory. *Meccanica*, 15(1):9–20.

Benettin, G., Galgani, L., Giorgilli, A., and Strelcyn, J. M. (1980b). Lyapunov Characteristic Exponents for smooth dynamical systems and for hamiltonian systems; A method for computing all of them. Part 2: Numerical application. *Meccanica*, 15(1):21–30.

Blanes, S., Casas, F., Farrés, A., Laskar, J., Makazaga, J., and Murua, A. (2013). New families of symplectic splitting methods for numerical integration in dynamical astronomy. *Appl. Numer. Math.*, 68:58–72.

Bodyfelt, J. D., Laptyeva, T. V., Skokos, C., Krimer, D. O., and Flach, S. (2011). Nonlinear waves in disordered chains: Probing the limits of chaos and spreading. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, 84(1):1–10.

Box, G. E. P. and Muller, M. E. (1958). A note on the generation of random normal deviates. *Ann. Math. Stat.*, 29(2):610–611.

Cai, W., Moore, A. L., Zhu, Y., Li, X., Chen, S., Shi, L., and Ruoff, R. S. (2010). Thermal transport in suspended and supported monolayer graphene grown by chemical vapor deposition. *Nano Lett.*, 10(5):1645–1651.

Castro Neto, A. H., Guinea, F., Peres, N. M. R., Novoselov, K. S., and Geim, A. K. (2007). The electronic properties of graphene. *Rev. Mod. Phys.*, 81(1):109–162.

Chen, S. S., Moore, A. L., Cai, W. W., Suk, J. W., An, J. H., Mishra, C., Amos, C., Magnuson, C. W., Kang, J. Y., Shi, L., and Ruoff, R. S. (2011). Raman Measurements of Thermal Transport in Suspended Monolayer Graphene of Variable Sizes in Vacuum and Gaseous Environments. *ACS Nano*, 5(1):321–328.

Da Silva, A., Cândido, L., Teixeira Rabelo, J., Hai, G.-Q., and Peeters, F. (2014). Anharmonic effects on thermodynamic properties of a graphene monolayer. *EPL (Europhysics Lett.*, 107(5):56004.

Devaney, R. L. (1989). *An Introductipon to Chaotic Dynamical systems*. Addison-Wesley Publishing Company, New York, 2 edition.

Fermi, E. , Pasta, J., Ulam, S. (1955). Studies of nonlinear problems. Technical report, Los Alamos Scientific Laboratory, California.

Flach, S., Krimer, D. O., and Skokos, C. (2009). Universal spreading of wave packets in disordered nonlinear systems. *Phys. Rev. Lett.*, 102(2):1–4.

Forest, E. and Ruth, R. D. (1990). Fourth-Order Symplectic Integration. *Phys. D*, 43:105–117.

Frank, O., Tsoukleri, G., Parthenios, J., Papagelis, K., Riaz, I., Jalil, R., Novoselov, K. S., and Galiotis, C. (2010). Compression Behavior of Single-layer Graphene. *ACS Nano*, 4(6):3131–3138.

Frenkel, D. and Smit, B. (2002). *Understanding Molecular Simulation*. Academic Press, second edition.

Geim, A. K. (2009). Graphene : Status and Prospects. *Science*, 324(5934):1530–1535.

Gerlach, E. and Skokos, C. (2011). Comparing the efficiency of numerical techniques for the integration of variational equations. *Discret. Contin. Dyn. Syst. Ser. A*, (SUPPL.):475–484.

Ghosh, S., Calizo, I., Teweldebrhan, D., Pokatilov, E. P., Nika, D. L., Balandin, A. A., Bao, W., Miao, F., and Lau, C. N. (2008). Extremely high thermal conductivity of graphene: Prospects for thermal management applications in nanoelectronic circuits. *Appl. Phys. Lett.*, 92(15):2–4.

Hénon, M. (1982). On the Numerical Computation of Poincari˜ Maps. *Phys. D*, pages 412–414.

Hénon, M. and Heiles, C. (1964). The Applicability of the Third Integral of Motion: Some Numerical Experiments. *Astron. J.*, 69(1):73–79.

Kalosakas, G., Lathiotakis, N., Galiotis, C., and Papagelis, K. (2013). In-plane force fields and elastic properties of graphene. *J. Appl. Phys.*, 113(13):1–7.

Koukaras, E. N., Kalosakas, G., Galiotis, C., and Papagelis, K. (2015). Phonon properties of graphene derived from molecular dynamics simulations. *Sci. Rep.*, 5:12923.

Kutta, W. (1901). Beitrag zur n˜herungsweisen Integration totaler Differentialgleichungen,. *Z. Math. Phys*, 46:435–453.

Lahini, Y., Avidan, A., Pozzi, F., Sorel, M., Morandotti, R., Christodoulides, D. N., and Silberberg, Y. (2008). Anderson localization and nonlinearity in one-dimensional disordered photonic lattices. *Phys. Rev. Lett.*, 100(1):1–4.

Landau, L. D. (1937). On the theory of phase transitions. *Zh. Eks. Teor. Fiz.*, 7(1937):19–32.

Laptyeva, T. V., Bodyfelt, J. D., Krimer, D. O., Skokos, C., and Flach, S. (2010). The crossover from strong to weak chaos for nonlinear waves in disordered systems. *Europhys. Lett.*, 91(3):5.

Los, J. H. and Fasolino, a. (2003). Intrinsic long-range bond-order potential for carbon: Performance in Monte Carlo simulations of graphitization. *Phys. Rev. B*, 68(2):24107.

Los, J. H., Ghiringhelli, L. M., Meijer, E. J., and Fasolino, A. (2005). Improved long-range reactive bond-order potential for carbon. I. Construction. *Phys. Rev. B*, 72(21):214102–214114.

Mendes, R. G., Bachmatiuk, A., Buechner, B., Cuniberti, G., and Rümmeli, M. (2013). Carbon nanostructures as multi-functional drug delivery platforms. *J. Mater. Chem. B*, pages 401–428.

Michel, K. H., Costamagna, S., and Peeters, F. M. (2015). Theory of anharmonic phonons in two-dimensional crystals. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 91(13).

Mulansky, M. and Pikovsky, A. (2013). Energy spreading in strongly nonlinear disordered lattices. *New J. Phys.*, 15.

Novoselov, K. S., Fal, V. I., Colombo, L., Gellert, P. R., Schwab, M. G., Kim, K., Ko, V. I. F., Colombo, L., Gellert, P. R., Schwab, M. G., and Kim, K. (2012). A roadmap for graphene. *Nature*, 490(7419):192–200.

Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D., Katsnelson, M. I., Grigorieva, I. V., Dubonos, S. V., and Firsov, A. A. (2005a). Two-dimensional gas of massless Dirac fermions in graphene. *Nature*, 438(7065):197–200.

Novoselov, K. S., Geim, A. K., Morozov, S. V., Jiang, D., Zhang, Y., Dubonos, S. V., Grigorieva, I. V., and Firsov, A. A. (2004). Electric field effect in atomically thin carbon films. *Science*, 306(5696):666–669.

Novoselov, K. S., Jiang, D., Schedin, F., Booth, T. J., Khotkevich, V. V., Morozov, S. V., and Geim, A. K. (2005b). Two-dimensional atomic crystals. *Proc. Natl. Acad. Sci. U. S. A.*, 102(30):10451–10453.

Oseledec, V. (1968). A Multiplicative Ergodic Theorem. Lyapunov Characteristic numbers for dynamical systems. *Trans. Moscow Math. Soc.*, 19:197–231.

Palleari, S. and Penati, T. (2008). Numerical Methods and Results in the FPU Problem. In Gavallotti, G., editor, *Fermi-Pasta-Ulam Probl. A Status Rep.*, chapter 7, pages 239–279. Springer, Berlin-Heidelberg.

Paulatto, L., Mauri, F., and Lazzeri, M. (2013). Anharmonic properties from a generalized third-order ab initio approach: Theory and applications to graphite and graphene. *Phys. Rev. B - Condens. Matter Mater. Phys.*, 87(21):1–18.

Ramanathan, T., Abdala, a. a., Stankovich, S., Dikin, D. a., Herrera-Alonso, M., Piner, R. D., Adamson, D. H., Schniepp, H. C., Chen, X., Ruoff, R. S., Nguyen, S. T., Aksay, I. a., Prud'Homme, R. K., and Brinson, L. C. (2008). Functionalized graphene sheets for polymer nanocomposites. *Nat. Nanotechnol.*, 3:327–331.

Skokos, C., Gkolias, I., and Flach, S. (2013). Nonequilibrium chaos of disordered nonlinear waves. *Phys. Rev. Lett.*, 111(6):1–5.

Skokos, C. H. (2010). Lecture Notes in Physics. In Gottwald, G. and Laskar, J., editors, *Lyapunov Charact. Exponents their Comput.*, pages 63–135. Springer Heidelberg, Berlin Heidelberg.

Stankovich, S., Dikin, D. A., Dommett, G. H. B., Kohlhaas, K. M., Zimney, E. J., Stach, E. A., Piner, R. D., Nguyen, S. T., and Ruoff, R. S. (2006). Graphene-based composite materials. *Nature*, 442(7100):282–286.

Stuart, S. J., Tutein, A. B., and Harrison, J. A. (2000). A reactive potential for hydrocarbons with intermolecular interactions. *J. Chem. Phys.*, 112(2000):6472–6486.

Tersoff, J. (1988). New empirical approach for the structure and energy of covalent systems. *Phys. Rev. B*, 37(12):6991–7000.

Vicencio, R. A. and Flach, S. (2009). Control of wave packet spreading in nonlinear finite disordered lattices. *Phys. Rev. E - Stat. Nonlinear, Soft Matter Phys.*, 79(1):2–7.

Williams, G. (1997). *Chaos theory tamed*. Taylor and Francis, London.

Xu, Z. and Buehler, M. J. (2010). Geometry controls conformation of graphene sheets: Membranes, ribbons, and scrolls. *ACS Nano*, 4(7):3869–3876.

Zhang, L., Xia, J., Zhao, Q., Liu, L., and Zhang, Z. (2010). Functional graphene oxide as a nanocarrier for controlled loading and targeted delivery of mixed anticancer drugs. *Small*, 6(4):537–544.

Zhang, W., Guo, Z., Huang, D., Liu, Z., Guo, X., and Zhong, H. (2011). Synergistic effect of chemo-photothermal therapy using PEGylated graphene oxide. *Biomaterials*, 32(33):8555–8561.

Zhao, Q., Nardelli, M. B., and Bernholc, J. (2002). Ultimate strength of carbon nanotubes: A theoretical study. *Phys. Rev. B*, 65(14):144105.

Appendix

A1. Computer code for the time evolution of the graphene model

We provide here the Fortran90 code used for the time evolution of an initial configuration of the graphene model. As input to this code we provide the (usually) randomly created initial conditions of the lattice atoms and follow their evolution in time through the numerical integration of the Hamilton equations of motion. After some relaxation time (variable tequil in the code), the program evaluates the relative energy error (1.16), the system's temperature, as well as the mean temperature and its standard deviation. All computed results are written to appropriate data files.

```
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
!
!                    Cape Town 13 - 09 - 2017
!
!                      Program arn_all.f90
!                      ___________________
!  The program computes the evolution of an orbit for the Graphene
!  Hamiltonian system of 2 Degrees of Freedom.
!  In this program symplectic integration schemes are used with a
!  choice of 4 fourth order integrators: the Forest and Ruth, the
!  ABA104, the ABA864 and the ABA1064.
!
!=============================================================================
!==============      PARAMETERS AND VARIABLES      ===========================
!=============================================================================
!
!  nscreen              = The unit number of screen (6).
!  nfin                 = The unit number of the input file.
!  nfout1               = The unit number of the 1st output file.
!  nfout2               = The unit number of the 2nd output file.
!  nfout3               = The unit number of the 3rd output file.
!  zero                 = Parameter for 0.0_rk
!  wrong                = A logical variable used for checking the consistency
!                         of input values in check_arn. Initially it is set
!                         wrong=.false.. If it becomes .true. there exists an
!                         error and the program stops.
!  f_text               = Name of the output file where we write messages
!                         for the run of the program.
!  f_all_data           = Name of the output file where we write data
!                         for the whole evolution of the orbit.
!  f_all_temp_plot      = Name of the output file where we write data
!                         for the average temperature of the system for different
!                         excitations.
!  4*(nx*ny)            = The degrees of freedom of the system. Hence the
```

```
!                       dimension of the space where the orbit
!                       evolves is 8*(nx*ny).
!                       Parameters of the potential functions:
!                       The Morse potential is V(r) = dm(e^-am(r(i,j) - a) - 1)^2
!                       The bond deformation potential is
!                       V(phi) = rk1*(phi-pi0)**2/2._rk - rk2*(phi-pi0)**3/3._rk
!                       pi = 4._rk * atan(1._rk), pi0 = 2._rk*pi/3._rk, am = 1.96,
!                       dm = 5.7, a = 1.42, mc = 12.0, rk1 = 7.0, rk2 = 4.0.
!  en1                  = The nominal value of the Hamiltonian (the energy). It is
!                       given for each excitation.
!  x(: : 1)             = The array of positions in x direction.
!  x(: : 2)             = The array of positions in y direction.
!  x(: : 3)             = The array of momenta in x direction.
!  x(: : 4)             = The array of momenta in y direction.
!  t                    = The time.
!  tend                 = The final time of the evolution.
!  tau                  = The integration time step (positive or negative).
!  i_step_print         = When the number of iteration steps (i_step)
!                       is a multiple of i_step_print and
!                       the time t is written on the screen then the current
!                       point and time are written in f_all_data file.
!                       i_step_print should be positive.
!  i_scheme             = index of integration scheme to be used. This choice
!                       is made from the input file arn_all.ini.
!                       The choices are Forest and Ruth (i_scheme =1),
!                       ABA104 (i_scheme =2), ABA864 (i_scheme =3),
!                       ABA1064 (i_scheme =4).
!
!=============================================================
!================     INPUT OUTPUT FILES     ================
!=============================================================
!
!  INPUT                : arn_all.ini ------> nfin
!                       We give data in various namelists. As an example
!                       for times we get the following
!  t                    = starting time
!  tequil               = relaxation time
!  tau                  = time step
!  tend                 = integration time
!  enmin                = minimun (starting) energy of the system
!  enmax                = maximum (end) energy of the system
!  enstep               = the increase in system energy after all realization
!                       have been run
!  energy_acc           = is the desired accuarry of the system. If this accuraccy
!                       is breached the program stops
!  nx                   = number of atoms in x direction
!  ny                   = number of atoms in y direction
!  nreal                = number of realizations for each energy excitation
!
!  OUTPUT               : f_text      ------> nfout1
!                       We write messages for the run of the program.
!
!                       : f_all_data ------> nfout2
!                       We write data for the whole evolution of the orbit.
!                       If i_all=1_ik we write
!                       time, relative energy error, instantanous system temperature
!                       and root mean square displacement
!                       every time the number of iteration steps becomes
```

```
!                          equal to i_step_print.
!
!                          : f_all_templot ------> nfout3
!                          We write the average values of the energy, energy per
!                          atom, average temperature and standard deviation of the
!                          temperature
!
!=============================================================================
!==============       NEEDED PROGRAM STRUCTURES       ========================
!=============================================================================
!
!---MODULES
!    module kinds
!    module a_interfaces
!    module arn_init_val
!
!---SUBROUTINES
!    subroutine check_arn(nfile,wrong)
!    subroutine write_arn(nfile)
!
!---FUNCTIONS
!    function energy_arn(x)
!    function temp_arn(x)
!
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  program arn_all
        use arn_init_val
        use a_interfaces, only : energy_arn,write_arn,check_arn,&
        o4,aba104,aba864,aba1064, temp_arn

        implicit none
        integer(ik), parameter:: nscreen=6_ik
        integer(ik), parameter:: nfin=10_ik, nfout1=11_ik, &
                                 nfout2=12_ik,nfout3=13_ik
        real(rk), parameter:: zero=0.0_rk
        integer(ik):: i, k,j,iunit
        real(rk):: time1,time2,time_of_run,thresh
        logical:: wrong = .false.

        namelist /files/f_text,f_all_data,f_all_templot
        namelist /times/ t,tau,tend,tequil
        namelist /indices/i_step_print,i_scheme
        namelist /data_1/enmin,enmax, enstep, energy_acc
        namelist /lattice_info/nx,ny,nreal
        call CPU_time(time1)
        open(unit=nfin, file='arn_all.ini')

        read(nfin,files)
        read(nfin,times)
        read(nfin,indices)
        read(nfin,data_1)
        read(nfin,lattice_info)

        open(unit=nfout1,file=f_text)
        open(unit=nfout2,file=f_all_data)
        open(unit=nfout3,file=f_all_templot)

        p = log10(tend)
```

```
        c = p / 200.0_rk
        jd = c
        phi = (pi/3._rk)
        nlat = nx * ny
        nen = int(( enmax -enmin ) / enstep ) + 1
        thresh = tend - 500._rk * tau

        write(nfout3,16)
 16 format(12x,'# E(eV) ',24x,'E/atom (eV) ', 20x,'T(K)',22x,'dT(K)' )
!************************* Equilibrium Configuration ********************
        has = - a
        do i= 1,nx
             if (mod(i,2).eq.1) then
                has = has + a
             else
                has = has + 2._rk * a
             end if
             do j= 1,ny
                if (mod(j,2).eq.1) then
                        x(i,j,1) = has
                else
                        x(i,j,1) = has - (-1)**i * a *cos(phi)
                end if
                x(i,j,2) = real(j-1) * a *  sin(phi)
                x(i,j,3) = 0._rk
                x(i,j,4) = 0._rk
              end do
        end do
        xeq(1:nx,1:ny,1:2)  = x(1:nx,1:ny,1:2)

        do ien = 1 , nen
            energy = enmin + enstep * real(ien-1)
            print*,energy
            std_dev = 0._rk
            temp = 0._rk
            temp2 = 0._rk
            temptot = 0._rk
            do kan = 4001, (4000 + nreal)
!seeds for the generation of random deviates of the orbit
                seed(1:2*nlat) = kan
!generating random momenta for the orbit using the Box-Muller algorithm
                call random_seed(PUT=seed)
                call random_number(randp)
                x(1:nx,1:ny,3) = sqrt(-2._rk*log(randp(1:nx,1:ny,1)))*&
                cos(2._rk*pi*randp(1:nx,1:ny,2))
                x(1:nx,1:ny,4) = sqrt(-2._rk*log(randp(1:nx,1:ny,1)))*&
                sin(2._rk*pi*randp(1:nx,1:ny,2))
!Normalizing the random numbers and getting the momenta center of mass
                sumx = sum(x(1:nx,1:ny,3)) / real(nlat)
                sumy = sum(x(1:nx,1:ny,4)) / real(nlat)
                x(1:nx,1:ny,3) = x(1:nx,1:ny,3) - sumx
                x(1:nx,1:ny,4) = x(1:nx,1:ny,4) - sumy
                ekin = sum(x(1:nx,1:ny,3)**2) + sum(x(1:nx,1:ny,4)**2)
                ekin = ekin/(2._rk * mc)
                ld = sqrt(energy / ekin)
                x(1:nx,1:ny,3:4) = x(1:nx,1:ny,3:4) * ld
                en1 = energy
```

```
                do while((t-tequil)*tau<zero)
                   select case (i_scheme)
                   case(1_ik)
                   call o4(x,tau)
                   case(2_ik)
                   call aba104(x,tau)
                   case(3_ik)
                   call aba864(x,tau)
                   case(4_ik)
                   call aba1064(x,tau)
                   case default
                   write(nscreen,*) 'Wrong value of i_scheme = ', i_scheme
                   write(nfout1,*) 'Wrong value of i_scheme = ', i_scheme
                   stop
                   end select
                   t=t+tau
                end do

                poten = energy_arn(x) - (sum(x(1:nx,1:ny,3:4)**2)/2._rk/mc)
                skin = energy - poten
                ekin = sum(x(1:nx,1:ny,3:4)**2)
                ekin = ekin/(2._rk * mc)
                ld = sqrt(skin /ekin)
                x(1:nx,1:ny,3) = x(1:nx,1:ny,3) * ld
                x(1:nx,1:ny,4) = x(1:nx,1:ny,4) * ld
                particle_energy = energy_arn(x) / real(nlat)

!************************** End of Equilibrium Configuration *********************
                t = 0._rk
                call write_arn(nscreen)
                call write_arn(nfout1)
                call check_arn(nscreen,wrong)
                wrong = .false.
                call check_arn(nfout1,wrong)
                if (wrong .eqv. .true.) stop
                i_step=0_ik
                it = 0

                do while((t-tend)*tau<zero)
                   select case (i_scheme)
                   case(1_ik)
                   call o4(x,tau)
                   case(2_ik)
                   call aba104(x,tau)
                   case(3_ik)
                   call aba864(x,tau)
                   case(4_ik)
                   call aba1064(x,tau)
                   case default
                   write(nscreen,*) 'Wrong value of i_scheme = ', i_scheme
                   write(nfout1,*) 'Wrong value of i_scheme = ', i_scheme
                   stop
                   end select
                   it = it + 1
                   t=t+tau
                   i_step=i_step+1_ik

!Checking the energy accuracy
```

```
                    if(abs((energy_arn(x)-en1)/en1)>energy_acc) then
                      write(nfout1,100)t,en1,energy_arn(x),&
                      abs((energy_arn(x)-en1)/en1)
                      write(nscreen,100) t,en1,energy_arn(x),&
                      abs((energy_arn(x)-en1)/en1)
                      stop
                    end if
!Compute several quantities
                    RMSD = sqrt( sum((xeq(1:nx,1:ny,1:2) - x(1:nx,1:ny,1:2))**2)/&
                    real(nlat) )
                    ekin = sum(x(1:nx,1:ny,3:4)**2)
                    ekin = ekin/(2._rk * mc)
                    write(nfout2,200) t,(abs((energy_arn(x)-en1)/en1)),&
                    temp_arn(x),RMSD
                    if(t .gt. thresh )then
                      temp = temp_arn(x)
                      temptot = temptot + temp_arn(x)
                      temp2 = temp2 + temp * temp
                      i_step=0_ik
                      c = c + jd
                    end if
                  end do
                end do

!************************ Computation of the temperature *********************
                sqtemp = temp2 /((1._rk + real(tend -thresh) /tau)*real(nreal))
                avtemp = temptot / ((1._rk + real(tend -thresh)/tau)*real(nreal))
                std_dev =  sqrt (sqtemp - (avtemp *avtemp))
                std_dev = std_dev *sqrt((real(tend -thresh)/tau)*real(nreal))
                std_dev = std_dev/sqrt((1._rk + real(tend-thresh)/tau)*real(nreal))
                print*, energy,energy/real(nlat),avtemp,std_dev
                write(nfout3,300) energy,energy/real(nlat),avtemp,std_dev
              end do
              call CPU_time(time2)
              time_of_run = time2 -time1
              print*,time_of_run
              close(nfin)
              close(nfout1)
              close(nfout2)
              close(nfout3)

100 format(' ATTENTION, THE ENERGY CHANGES !!!!! ',/, &
          ' at time t       = ',f28.18,/,             &
          ' initial energy  = ',f28.18,/,             &
          ' current energy  = ',f28.18,/,             &
          ' relative difference = ',d28.18,/)
200 format(4(f28.18,tr1))
300 format(4(f28.18,tr1))
end program
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

A2. Subroutines and Functions

In what follows we provide all the subroutines and functions called in the main program.

The forces for the central atom of a triplet of atoms seen in Figure 2.4 are computed in the subroutine

fangcen, whilst those for the edge atoms are computed in the subroutine fangedge. The functions energy_arn and temp_arn compute respectively the total energy of the system and the temperature at each time step. The operators la_arn and lb_arn are related to the kinetic energy and potential energy parts of the Hamiltonian and are needed for the action of SIs. The subroutines which implement the action of the SIs Forest and Ruth, ABA104, ABA864 and ABA1064 are respectively o4, aba104, aba864 and aba1064. The subroutine check_arn performs some checks on the various values used in the code. Finally, subroutine write_arn is used for writing data in files and at the computer screen.

```
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
!
!     THIS FILE CONTAINS SUBROUTINES CREATED BY A. NGAPASARE
!     THE SUBROUTINES ARE LISTED IN ALPHABETICAL ORDER
!
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
  subroutine la_arn(x,tau)
      use kinds
      implicit none
      real(rk), parameter:: mc = 12._rk, a = 1.42_rk
      real(rk), parameter:: pi = 4._rk * atan(1._rk),pi0 = 2._rk*pi/3._rk
      real(rk), parameter:: kb = 0.00008617_rk,dm = 5.7_rk
      real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
      integer(ik), parameter::  nd = 102,nx = 12, ny = 16
      real(rk), dimension(-1:nd,-1:nd,1:4):: x
      real(rk), intent(in):: tau
      x(1:nx,1:ny,1:2) = x(1:nx,1:ny,1:2) + tau*(x(1:nx,1:ny,3:4)/mc)
  end subroutine la_arn
!============================================================================
  subroutine lb_arn(x,tau)
      use kinds
      implicit none
      real(rk), parameter:: mc = 12._rk, a = 1.42_rk
      real(rk), parameter:: pi = 4._rk * atan(1._rk),pi0 = 2._rk*pi/3._rk
      real(rk), parameter:: kb = 0.00008617_rk,dm = 5.7_rk
      real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
      integer(ik), parameter::  nd = 102,nx = 12, ny = 16
      real(rk), dimension(-1:nd,-1:nd,1:4):: x
      real(rk):: hr1,hr2,hf1,hf2,hr3,hf3
      real(rk):: fr1x,fr1y,fr2x,fr2y,fr3x,fr3y
      real(rk):: fgc2x,fgc2y,fgc1x,fgc1y,fga1ax,fga1ay,fga1bx,fga1by,e,b,c,d
      real(rk):: fga2bx,fga2by,fga2ax,fga2ay,fga3ax,fga3ay
      real(rk):: fga3bx,fga3by,fgc3x,fgc3y
      real(rk), intent(in):: tau
      integer(ik):: i,j,k
      do i = 1,nx
          x(i,0,1) = x(i,ny,1)
          x(i,0,2) = x(i,ny,2) - real(ny) * a * sqrt(3._rk) / 2._rk
          x(i,ny+1,1) = x(i,1,1)
          x(i,ny+1,2) = x(i,1,2) + real(ny) * a * sqrt(3._rk) / 2._rk
          x(i,-1,1) = x(i,ny-1,1)
          x(i,-1,2) = x(i,ny-1,2) - real(ny) * a * sqrt(3._rk) / 2._rk
          x(i,ny+2,1) = x(i,2,1)
          x(i,ny+2,2) = x(i,2,2) + real(ny) * a * sqrt(3._rk) / 2._rk
      end do
      do j = 0,ny+1
          x(0,j,1) = x(nx,j,1) - real(nx) * 3._rk * a / 2._rk
          x(0,j,2) = x(nx,j,2)
```

```
      x(nx+1,j,1) = x(1,j,1) + real(nx) * 3._rk * a / 2._rk
      x(nx+1,j,2) = x(1,j,2)
    end do
    do i = 1,nx
      hr1 = sqrt( (x(i,1,1)-x(i,0,1))**2 + (x(i,1,2)-x(i,0,2))**2 )
      hf1 = exp( -am * (hr1-a) )
      hf1 = 2._rk*am*dm*hf1*(hf1-1._rk) / hr1
      do j = 1,ny
        hr2 = hr1
        hf2 = hf1
        fr2x = hf2 * ( x(i,j,1) - x(i,j-1,1) )
        fr2y = hf2 * ( x(i,j,2) - x(i,j-1,2) )
        hr1 = sqrt( (x(i,j,1)-x(i,j+1,1))**2 + (x(i,j,2)-x(i,j+1,2))**2 )
        hf1 = exp( -am * (hr1-a) )
        hf1 = 2._rk*am*dm*hf1*(hf1-1._rk) / hr1
        fr1x = hf1 * ( x(i,j,1) - x(i,j+1,1) )
        fr1y = hf1 * ( x(i,j,2) - x(i,j+1,2) )
        if (mod(i+j,2).eq.1) then
          hr3 = sqrt( (x(i,j,1)-x(i+1,j,1))**2 + (x(i,j,2)-x(i+1,j,2))**2 )
          hf3 = exp( -am * (hr3-a) )
          hf3 = 2._rk*am*dm*hf3*(hf3-1._rk) / hr3
          fr3x = hf3 * ( x(i,j,1) - x(i+1,j,1) )
          fr3y = hf3 * ( x(i,j,2) - x(i+1,j,2) )
          call fangcen(x(i,j,1),x(i,j,2),x(i,j+1,1),x(i,j+1,2),x(i+1,j,1),&
          x(i+1,j,2),fgc2x,fgc2y)
          call fangcen(x(i,j,1),x(i,j,2),x(i,j-1,1),x(i,j-1,2),x(i+1,j,1),&
          x(i+1,j,2),fgc1x,fgc1y)
          call fangcen(x(i,j,1),x(i,j,2),x(i,j+1,1),x(i,j+1,2),x(i,j-1,1),&
          x(i,j-1,2),fgc3x,fgc3y)
          call fangedge(x(i,j+1,1),x(i,j+1,2),x(i,j,1),x(i,j,2),x(i,j+2,1),&
          x(i,j+2,2),fga1ax,fga1ay)
          call fangedge(x(i,j+1,1),x(i,j+1,2),x(i,j,1),x(i,j,2),x(i-1,j+1,1),&
          x(i-1,j+1,2),fga1bx,fga1by)
          call fangedge(x(i,j-1,1),x(i,j-1,2),x(i,j,1),x(i,j,2),x(i-1,j-1,1),&
          x(i-1,j-1,2),fga2ax,fga2ay)
          call fangedge(x(i,j-1,1),x(i,j-1,2),x(i,j,1),x(i,j,2),x(i,j-2,1),&
          x(i,j-2,2),fga2bx,fga2by)
          call fangedge(x(i+1,j,1),x(i+1,j,2),x(i,j,1),x(i,j,2),x(i+1,j-1,1),&
          x(i+1,j-1,2),fga3ax,fga3ay)
          call fangedge(x(i+1,j,1),x(i+1,j,2),x(i,j,1),x(i,j,2),x(i+1,j+1,1),&
          x(i+1,j+1,2),fga3bx,fga3by)
        else
          hr3 = sqrt( (x(i,j,1)-x(i-1,j,1))**2 + (x(i,j,2)-x(i-1,j,2))**2 )
          hf3 = exp( -am * (hr3-a) )
          hf3 = 2._rk*am*dm*hf3*(hf3-1._rk) / hr3
          fr3x = hf3 * ( x(i,j,1) - x(i-1,j,1) )
          fr3y = hf3 * ( x(i,j,2) - x(i-1,j,2) )
          call fangcen(x(i,j,1),x(i,j,2),x(i,j+1,1),x(i,j+1,2),x(i-1,j,1),&
          x(i-1,j,2),fgc2x,fgc2y)
          call fangcen(x(i,j,1),x(i,j,2),x(i-1,j,1),x(i-1,j,2),x(i,j-1,1),&
          x(i,j-1,2),fgc1x,fgc1y)
          call fangcen(x(i,j,1),x(i,j,2),x(i,j-1,1),x(i,j-1,2),x(i,j+1,1),&
          x(i,j+1,2),fgc3x,fgc3y)
          call fangedge(x(i,j+1,1),x(i,j+1,2),x(i,j,1),x(i,j,2),x(i+1,j+1,1),&
          x(i+1,j+1,2),fga1ax,fga1ay)
          call fangedge(x(i,j+1,1),x(i,j+1,2),x(i,j,1),x(i,j,2),x(i,j+2,1),&
          x(i,j+2,2),fga1bx,fga1by)
          call fangedge(x(i-1,j,1),x(i-1,j,2),x(i,j,1),x(i,j,2),x(i-1,j+1,1),&
```

```
                x(i-1,j+1,2),fga3ax,fga3ay)
                call fangedge(x(i-1,j,1),x(i-1,j,2),x(i,j,1),x(i,j,2),x(i-1,j-1,1),&
                x(i-1,j-1,2),fga3bx,fga3by)
                call fangedge(x(i,j-1,1),x(i,j-1,2),x(i,j,1),x(i,j,2),x(i,j-2,1),&
                x(i,j-2,2),fga2ax,fga2ay)
                call fangedge(x(i,j-1,1),x(i,j-1,2),x(i,j,1),x(i,j,2),x(i+1,j-1,1),&
                x(i+1,j-1,2),fga2bx,fga2by)
              end if
              x(i,j,3) = x(i,j,3) +  tau * (fr1x + fr2x + fr3x) + tau * (fgc1x + &
              fgc2x + fgc3x + fga1ax + fga1bx + fga2ax +fga2bx + fga3ax +fga3bx)
              x(i,j,4) = x(i,j,4) + tau * (fr1y + fr2y + fr3y) + tau * (fgc1y + &
              fgc2y + fgc3y + fga1ay + fga1by + fga2ay +fga2by + fga3ay +fga3by)
            end do
            end do
  end subroutine lb_arn
!================ Force at a central atom ===================================
  subroutine fangcen(xc,yc,xa1,ya1,xa2,ya2,fx,fy)
      use kinds
      implicit none
      real(rk), parameter:: pi = 4._rk * atan(1._rk),pi0 = 2._rk*pi/3._rk
      real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
      real(rk), intent(in) :: xc,yc,xa1,ya1,xa2,ya2
      real(rk) :: h1x,h1y,h2x,h2y,ha,r1,r2,hb,g,hfx,hfy,fg
      real(rk), intent(out) :: fx,fy

      h1x = xc - xa1
      h1y = yc - ya1
      h2x = xc - xa2
      h2y = yc - ya2
      ha = h1x * h2x + h1y * h2y
      r1 = sqrt( h1x * h1x + h1y * h1y )
      r2 = sqrt( h2x * h2x + h2y * h2y )
      hb = r1 * r2
      g = acos( ha / hb )
      hfx = ( h1x + h2x) * hb - ha * ( h1x * r2*r2 + h2x * r1*r1 ) / hb
      hfx = hfx / hb / hb
      hfy = ( h1y + h2y) * hb - ha * ( h1y * r2*r2 + h2y * r1*r1 ) / hb
      hfy = hfy / hb / hb
      fg = ( rk1 * (g-pi0) - rk2 * (g-pi0)**2 ) / sin(g)
      fx = fg * hfx
      fy = fg * hfy
  end subroutine fangcen
!================ Force at an atom at the edge ==============================
  subroutine fangedge(xc,yc,xa1,ya1,xa2,ya2,fx,fy)
      use kinds
      implicit none
      real(rk), parameter:: pi = 4._rk * atan(1._rk),pi0 = 2._rk*pi/3._rk
      real(rk), parameter:: am = 1.96_rk, rk1 = 7._rk, rk2 = 4._rk
      real(rk), intent(in) :: xc,yc,xa1,ya1,xa2,ya2
      real(rk) :: h1x,h1y,h2x,h2y,ha,r1,r2,hb,g,hfx,hfy,fg
      real(rk), intent(out) :: fx,fy

      h1x = xc - xa1
      h1y = yc - ya1
      h2x = xc - xa2
      h2y = yc - ya2
      ha = h1x * h2x + h1y * h2y
      r1 = sqrt( h1x * h1x + h1y * h1y )
```

```
      r2 = sqrt( h2x * h2x + h2y * h2y )
      hb = r1 * r2
      g = acos( ha / hb )
      hfx = - h2x * hb + ha * h1x * r2*r2 / hb
      hfx = hfx / hb / hb
      hfy = - h2y * hb + ha * h1y * r2*r2 / hb
      hfy = hfy / hb / hb
      fg = ( rk1 * (g-pi0) - rk2 * (g-pi0)**2 ) / sin(g)
      fx = fg * hfx
      fy = fg * hfy
  end subroutine fangedge
!========================== Forest and Ruth integrator ========================
  subroutine o4(x,tau)
      use a_interfaces, only:la_arn,lb_arn
      use kinds
      implicit none
      integer(ik), parameter::  nd = 102
      real(rk), dimension(-1:nd,-1:nd,4),intent(inout) :: x
      real(rk), intent(in) ::tau
      real(rk) :: c1,c2,c3,c4,d1,d2,d3,rr,tt
       rr=2._rk**(1._rk/3._rk)
       tt=2._rk - rr
       c1=0.5_rk/tt
       c2=0.5_rk*(1._rk-rr)/tt
       c3=c2
       c4=c1
       d1=1._rk/tt
       d2=-rr/tt
       d3=d1

       call la_arn(x,c1*tau)
       call lb_arn(x,d1*tau)
       call la_arn(x,c2*tau)
       call lb_arn(x,d2*tau)
       call la_arn(x,c3*tau)
       call lb_arn(x,d3*tau)
       call la_arn(x,c4*tau)
  end subroutine o4
!================== ABA104 integrator ================================================
  subroutine aba104(x,tau)
      use a_interfaces, only:la_arn,lb_arn
      use kinds
      implicit none
      integer(ik), parameter::  nd = 102
      real(rk), dimension(-1:nd,-1:nd,8),intent(inout) :: x
      real(rk), intent(in) ::tau
      real(rk) :: c1,c2,c3,c4,d1,d2,d3,d4
       c1=0.047067100645972506129478876372_rk
       c2=0.184756935417088106924737619370_rk
       c3=0.282706005679836205324361656554_rk
       c4=-0.014530041742896818378578152296_rk
       d1=0.118881917368197019945350395085_rk
       d2=0.241050460551501565744166786590_rk
       d3=-0.273286666705323806054311398166_rk
       d4=0.826708577571250440729588432981_rk

       call la_arn(x,c1*tau)
       call lb_arn(x,d1*tau)
```

```
        call la_arn(x,c2*tau)
        call lb_arn(x,d2*tau)
        call la_arn(x,c3*tau)
        call lb_arn(x,d3*tau)
        call la_arn(x,c4*tau)
        call lb_arn(x,d4*tau)
        call la_arn(x,c4*tau)
        call lb_arn(x,d3*tau)
        call la_arn(x,c3*tau)
        call lb_arn(x,d2*tau)
        call la_arn(x,c2*tau)
        call lb_arn(x,d1*tau)
        call la_arn(x,c1*tau)
  end subroutine aba104
!============================== ABA864 integrator ==============================
  subroutine aba864(x,tau)
      use a_interfaces, only:la_arn,lb_arn
      use kinds
      implicit none
      integer(ik), parameter::  nd = 102
      real(rk), dimension(-1:nd,-1:nd,8),intent(inout) :: x
      real(rk), intent(in) ::tau
      real(rk) :: c1,c2,c3,c4,d1,d2,d3,d4
       c1=0.071133426498223117777938730006_rk
       c2=0.241153427956640098736487795326_rk
       c3=0.521411761772814789212136078067_rk
       c4=-0.333698616227678005726562603400_rk
       d1=0.183083687472197221961703757166_rk
       d2=0.310782859898574869507522291054_rk
       d3=-0.026564618511958800697212137916_rk
       d4=0.065396142282373418455972179391_rk

       call la_arn(x,c1*tau)
       call lb_arn(x,d1*tau)
       call la_arn(x,c2*tau)
       call lb_arn(x,d2*tau)
       call la_arn(x,c3*tau)
       call lb_arn(x,d3*tau)
       call la_arn(x,c4*tau)
       call lb_arn(x,d4*tau)
       call la_arn(x,c4*tau)
       call lb_arn(x,d3*tau)
       call la_arn(x,c3*tau)
       call lb_arn(x,d2*tau)
       call la_arn(x,c2*tau)
       call lb_arn(x,d1*tau)
       call la_arn(x,c1*tau)
  end subroutine aba864
!============================== ABA1064 integrator =============================
  subroutine aba1064(x,tau)
      use a_interfaces, only:la_arn,lb_arn
      use kinds
      implicit none
      integer(ik), parameter::  nd = 102
      real(rk), dimension(-1:nd,-1:nd,8),intent(inout) :: x
      real(rk), intent(in) ::tau
      real(rk) :: c1,c2,c3,c4,c5,d1,d2,d3,d4
       c1=0.038094497422412195456975322308_rk
```

```
        c2=0.14529871611691374929402007266 0_rk
        c3=0.20762769572554125071620561132 4_rk
        c4=0.43590970365152615922315486240 1_rk
        c5=-0.65386122583278670938071173739 0_rk
        d1=0.09585888083707521061077150377 1_rk
        d2=0.20444615314299878068050778391 6_rk
        d3=0.21707034797899110171433859243 0_rk
        d4=-0.01737538195906509300561788011 8_rk

        call la_arn(x,c1*tau)
        call lb_arn(x,d1*tau)
        call la_arn(x,c2*tau)
        call lb_arn(x,d2*tau)
        call la_arn(x,c3*tau)
        call lb_arn(x,d3*tau)
        call la_arn(x,c4*tau)
        call lb_arn(x,d4*tau)
        call la_arn(x,c5*tau)
        call lb_arn(x,d4*tau)
        call la_arn(x,c4*tau)
        call lb_arn(x,d3*tau)
        call la_arn(x,c3*tau)
        call lb_arn(x,d2*tau)
        call la_arn(x,c2*tau)
        call lb_arn(x,d1*tau)
        call la_arn(x,c1*tau)
  end subroutine aba1064
!=======================================================================
  subroutine check_arn(nf,wrong)
      use arn_init_val
      implicit none
      integer(ik), intent(in):: nf
      logical, intent(inout):: wrong
      real(rk), parameter:: zero=0._rk

      if ((tend-t)*tau<=zero) then
          write(nf,100) t,tend,tau
          wrong = .true.
      end if
      if (i_step_print<=0_ik) then
          write(nf,200) i_step_print, tau
          wrong = .true.
      end if

100 format(1x,'---- ERROR  ----',/                    &
     1x, ' t, tend, tau are not correct ',/      &
     1x, ' initial time, t       = ',f30.18,/,  &
     1x, ' final time, tend      = ',f30.18,/,  &
     1x, ' integration step, tau = ',d30.18,/)
200 format(1x,'---- ERROR  ----',/  &
     1x, ' Probably we should put i_step_print > 0. ',/ &
     1x, ' i_step_print      = ',i30,/, &
     1x, ' integration step, tau = ',d30.18,/)
  end subroutine check_arn
!=======================================================================
  subroutine write_arn(nf)
      use arn_init_val
      use a_interfaces, only : energy_arn
```

```
         implicit none
         integer(ik), intent(in):: nf
         write(nf,100) f_text,f_all_data, i_scheme
         write(nf,200) t,energy_arn(x)
         write(nf,300) t,tequil, tend, nreal,tau, energy_acc
100 format(1x,'-----_OUTPUT__FILES_---',/                       &
      1x,'Text_with_data_about_the_run.______FILE_=_',a30,/   &
      1x,'All_numerical_results:_x,_t,_energy_error.____FILE_=_',a30,/&
      1x,'Integration_scheme_used',i5,/)
200 format('_-----_INITIAL_CONDITIONS_-----',/, &
      1x, '_at_t_=_',f28.18,/,        1x, '_with_ENERGY_=_',f28.18,/)
300 format(1x,'-----_INTEGRATION_---',/       &
      1x, '_initial_time________________=_',f30.18,/,  &
      1x, '_relaxation_time_____________=_',f30.18,/,  &
      1x, '_final_time__________________=_',f30.18,/,  &
      1x, '_total_realizations__________=_',i5,/,    &
      1x, '_integration_step____________=_',d30.18,/,  &
      1x, '_accuracy_of_energy__________=_',d30.18,/,  &
      1x, /)
   end subroutine write_arn
!==========================================================================
```

```
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
!
!      THIS FILE CONTAINS FUNCTIONS CREATED BY A. NGAPASARE
!      THE FUNCTIONS ARE LISTED IN ALPHABETICAL ORDER
!
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    function energy_arn(x)
!--------------------------------------------------------------------------
! The Graphene Hamiltonian:
! H=0.5*(px^2(i,j)+py^2(i,j))/m_c+0.5*(V(r))+V(phi).
! The Morse potential is V(r) = dm(e^-am(r(i,j) - a) - 1  )^2
! The Bond deformation potential is
! V(phi) = rk1 * (phi-pi0)**2 / 2._rk - rk2 * (phi-pi0)**3 / 3._rk
!--------------------------------------------------------------------------
         use kinds
         implicit none
         real(rk), parameter:: mc = 12._rk, a = 1.42_rk
         real(rk), parameter:: pi = 4._rk * atan(1._rk),pi0 = 2._rk*pi/3._rk
         real(rk), parameter:: kb = 0.00008617_rk,dm = 5.7_rk
         real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
         real(rk):: hr1,hr2,hf1,hf2,hr3,hf3,ga1,ga2,ga3,gb1,gb2,gb3
         integer(ik), parameter::  nd = 102,nx = 12, ny = 16
         real(rk), dimension(-1:nd,-1:nd,4):: x
         real(rk):: energy_arn, pot_en,const,gf1,gf2,gf3, angpot1,angpot2,angpot3
         integer(ik):: i,j,k
         const = 2._rk * mc
         const = 1._rk / const
!Implementing periodic boundary conditions
         do i = 1,nx
             x(i,0,1) = x(i,ny,1)
             x(i,0,2) = x(i,ny,2) - real(ny) * a * sqrt(3._rk) / 2._rk
             x(i,ny+1,1) = x(i,1,1)
             x(i,ny+1,2) = x(i,1,2) + real(ny) * a * sqrt(3._rk) / 2._rk
         end do
         do j = 0,ny+1
             x(0,j,1) = x(nx,j,1) - real(nx) * 3._rk * a / 2._rk
             x(0,j,2) = x(nx,j,2)
```

```
            x(nx+1,j,1) = x(1,j,1) + real(nx) * 3._rk * a / 2._rk
            x(nx+1,j,2) = x(1,j,2)
        end do
        energy_arn = 0._rk
        do i = 1, nx
          do j = 1, ny
            hr1 = sqrt( (x(i,j,1)-x(i,j+1,1))**2 + (x(i,j,2)-x(i,j+1,2))**2 )
            hf1 = 0.5_rk * dm * ( exp(-2._rk*am*(hr1-a)) &
              - 2._rk * exp(-am*(hr1-a)) + 1._rk )
            hr2 = sqrt( (x(i,j,1)-x(i,j-1,1))**2 + (x(i,j,2)-x(i,j-1,2))**2 )
            hf2 = 0.5_rk * dm * ( exp(-2._rk*am*(hr2-a)) &
            - 2._rk * exp(-am*(hr2-a)) + 1._rk)
            if (mod(i+j,2).eq.1) then
                hr3 = sqrt((x(i,j,1) - x(i+1,j,1))**2 + (x(i,j,2) - x(i+1,j,2))**2)
            else
                hr3 = sqrt((x(i,j,1) - x(i-1,j,1))**2 + (x(i,j,2) - x(i-1,j,2))**2)
            endif
            hf3 = 0.5_rk * dm * ( exp(-2._rk*am*(hr3-a)) &
            - 2._rk * exp(-am*(hr3-a)) + 1._rk)
! Establishing the angles between the atoms centred at (i,j)
            ga1 = (x(i,j+1,1)-x(i,j,1)) * (x(i,j-1,1)-x(i,j,1)) + &
            (x(i,j+1,2)-x(i,j,2))* (x(i,j-1,2)-x(i,j,2))
            gb1 = hr1 * hr2
            gf1 = acos( ga1 / gb1 )
            angpot1 = rk1 * (gf1-pi0)**2 / 2._rk - rk2 * (gf1-pi0)**3 / 3._rk
            if (mod(i+j,2).eq.1) then
                ga2 = (x(i,j+1,1)-x(i,j,1)) * (x(i+1,j,1)-x(i,j,1)) + &
                (x(i,j+1,2)-x(i,j,2))  * (x(i+1,j,2)-x(i,j,2))
                gb2 = hr1 * hr3
            else
                ga2 = (x(i,j-1,1)-x(i,j,1)) * (x(i-1,j,1)-x(i,j,1)) + &
                (x(i,j-1,2)-x(i,j,2))* (x(i-1,j,2)-x(i,j,2))
                gb2 = hr2 * hr3
            end if
            gf2 = acos( ga2 / gb2 )
            angpot2 = rk1 * (gf2-pi0)**2 / 2._rk - rk2 * (gf2-pi0)**3 / 3._rk
            gf3 = 2._rk * pi - gf1 - gf2
            angpot3 = rk1 * (gf3-pi0)**2 / 2._rk - rk2 * (gf3-pi0)**3 / 3._rk
            energy_arn = energy_arn + const*(x(i,j,3)**2 + x(i,j,4)**2) &
            + hf1 + hf2 + hf3 + angpot1 + angpot2 + angpot3
          end do
        end do
    end function energy_arn
!=============================================================
    function temp_arn(x)
        use kinds
        implicit none
        real(rk), parameter:: mc = 12._rk, a = 1.42_rk
        real(rk), parameter:: pi = 4._rk * atan(1.00_rk),pi0 = 2._rk*pi/3._rk
        real(rk), parameter:: kb = 0.00008617_rk,dm = 5.70_rk
        real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
        integer(ik), parameter::  nd = 102,nx = 12, ny = 16
        real(rk), dimension(-1:nd,-1:nd,4):: x
        real(rk):: temp_arn, ekin
        integer(ik):: i,j
        temp_arn = 0._rk
        ekin = 0._rk
! Summing up the total Kinetic Energy of the system
```

```
        do i = 1,nx
          do j = 1, ny
            ekin = ekin + (x(i,j,3)**2)/2._rk/ mc + (x(i,j,4)**2)/2._rk/mc
          end do
        end do
!Getting the system temperature from < E_kin > = nx * ny * kb * T
        ekin = ekin/real(nx*ny)
        temp_arn = ekin/kb
    end function temp_arn
```

A3. Modules

Here we list the modules used in our program, along with the interfaces of the various subroutines.

```
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
!
!     THIS FILE CONTAINS MODULES CREATED BY A.NGAPASARE
!
!$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
!=======================================================================
    module kinds
        integer, parameter:: rk=8
        integer, parameter:: ik=4
    end module kinds
!=======================================================================
    module arn_init_val
        use kinds
        real(rk), parameter:: mc = 12._rk, a = 1.42_rk
        real(rk), parameter:: pi = 4._rk * atan(1.00_rk),pi0 = 2._rk*pi/3._rk
        real(rk), parameter:: kb = 0.00008617_rk,dm = 5.7_rk
        real(rk), parameter:: am = 1.96_rk,rk1 = 7._rk,rk2 = 4._rk
        integer(ik), parameter:: nd = 102
        integer(ik) :: i_step_print,i_step, i_scheme, it
        real(rk), dimension(-1:nd,-1:nd,4):: x, xeq
        real(rk), dimension(1:12,1:16,1:2) :: randp
        real(rk):: t, tau, tend,tequil, enmin,has,energy,diff,scal
        real(rk):: en1,energy_acc,phi,enstep,ld,poten
        real(rk):: enmax,ener_diff,sumx,sumy,ekin,skin,temp2,temptot
        real(rk):: avrealztemps,avtemps, avtemp, sqtemp
        real(rk):: temp,std_dev,p,c,jd,dek,dtemp,particle_energy,RMSD
        character(len=30):: f_text, f_all_data,f_all_templot
        integer(ik), dimension(1:192):: seed
        integer(ik):: nx, ny,ireal,nreal,ien,nen,kan,nlat
    end module arn_init_val
!=======================================================================
!
!                 I N T E R F A C E S
!
!=======================================================================
!
    module a_interfaces
!
!=======================================================================
! Interfaces in alphabetical order for subroutines (S)
! and functions (F)
! INDEX:
```

```
!      S-aba104
!      S-aba864
!      S-aba1064
!      S-check_arn
!      F-energy_arn
!      S-fangcen
!      S-fangedge
!      S-la_arn
!      S-lb_arn
!      S-o4
!      F-temp_arn
!      S-write_arn
!-------------------------------------------------------------------------------
        interface
          subroutine aba104(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
                real(rk):: c1, c2, c3,c4, d1,d2,d3,d4
           end subroutine aba104
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine aba864(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
                real(rk):: c1, c2, c3,c4, d1,d2,d3,d4
           end subroutine aba864
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine aba1064(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
                real(rk):: c1, c2, c3,c4,c5,d1,d2,d3,d4
           end subroutine aba1064
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine check_arn(nf,wrong)
                use kinds
                integer(ik), intent(in):: nf
                logical, intent(inout):: wrong
           end subroutine check_arn
        end interface
!-------------------------------------------------------------------------------
        interface
          function energy_arn(x)
                use kinds
                real(rk), dimension(12,16,4), intent(in):: x
                real(rk):: energy_arn
           end function energy_arn
        end interface
```

```
!-------------------------------------------------------------------------------
        interface
            subroutine fangcen(xc,yc,xa1,ya1,xa2,ya2,fx,fy)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4) :: xc,yc,xa1,ya1,xa2,ya2
                real(rk) :: h1x,h1y,h2x,h2y,ha,r1,r2,hb,g,hfx,hfy,fg
                real(rk), intent(out) :: fx,fy
           end subroutine fangcen
        end interface
!-------------------------------------------------------------------------------
        interface
            subroutine fangedge(xc,yc,xa1,ya1,xa2,ya2,fx,fy)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4) :: xc,yc,xa1,ya1,xa2,ya2
                real(rk) :: h1x,h1y,h2x,h2y,ha,r1,r2,hb,g,hfx,hfy,fg
                real(rk), intent(out) :: fx,fy
           end subroutine fangedge
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine la_arn(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
           end subroutine la_arn
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine lb_arn(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
           end subroutine lb_arn
        end interface
!-------------------------------------------------------------------------------
        interface
          subroutine o4(x,tau)
                use kinds
                integer(ik), parameter::  nd = 102
                real(rk), dimension(-1:nd,-1:nd,4), intent(inout):: x
                real(rk), intent(in):: tau
                real(rk):: c1, c2, c3,c4, d1,d2,d3,rr, tt
           end subroutine o4
        end interface
!-------------------------------------------------------------------
        interface
          function temp_arn(x)
                use kinds
                real(rk), dimension(12,16,4), intent(in):: x
                real(rk):: temp_arn
           end function temp_arn
        end interface
!-------------------------------------------------------------------------
        interface
```

```
          subroutine write_arn(nf)
                use kinds
                integer(ik), intent(in):: nf
           end subroutine write_arn
        end interface
!-------------------------------------------------------------------------------
    end module a_interfaces
!===============================================================================
```

A4. Input file

In order to show the format of the data provided to the code we present below a typical input file.

```
&files
  f_text=     'graphene.txt'
  f_all_data='all.dat'
  f_all_templot='temp_plot.dat'
&end
&times
  t=    0.
  tau = 0.1
  tend= 50.0
  tequil=10.0
&end
&indices
  i_step_print=1
  i_scheme =3
&end
&data_1
  enmin = 2.00
  enmax = 160.00
  enstep = 8.0
  energy_acc=1-6
&end
&lattice_info
  nx   = 12
  ny   = 16
  nreal = 500
&end
```