Atomistic-Based Finite Element Simulation of Carbon Nanotubes

Yang Yang
Western Michigan University

Follow this and additional works at: https://scholarworks.wmich.edu/dissertations

Part of the Aerospace Engineering Commons, and the Mechanical Engineering Commons

Recommended Citation

This Dissertation-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Dissertations by an authorized administrator of ScholarWorks at WMU. For more information, please contact wmu-scholarworks@wmich.edu.
ATOMISTIC-BASED FINITE ELEMENT SIMULATION OF CARBON NANOTUBES

by

Yang Yang

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Mechanical and Aeronautical Engineering
Dr. William W. Liou, Advisor

Western Michigan University
Kalamazoo, Michigan
December 2008
INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.
ACKNOWLEDGMENTS

I am grateful to many people. First of all is my thesis advisor, Dr. William Liou, who provides me with invaluable academic guidance, generous financial support, and a good personal friendship. Over these years at Western, Dr. Liou is always ready to offer his help, which I will remember forever. And from him I learned how to be a successful scholar and how to work as a team member.

Also, I'd like to thank my thesis committee members, Drs. Christopher Cho, Tianshu Liu and Yirong Mo. Thanks for their time in reading my thesis and the help offered during the course of my research.

I also thank my lab mates, Meng-huang Lu, Srinvasa Pantula, and Yongqing Peng for their friendliness and help.

I thank all my family members, especially my grandparents, parents and parents in-law who always put trust in me even though I sometimes don't.

Finally I thank my wife Yu Zhang for her love.

Yang Yang
TABLE OF CONTENTS

ACKNOWLEDGMENTS .................................................................................................................. ii

LIST OF TABLES ........................................................................................................................ vi

LIST OF FIGURES ....................................................................................................................... vii

CHAPTER

1 INTRODUCTION ....................................................................................................................... 1

1.1 Fabrications and definitions of carbon nanotubes ............................................................. 2

1.1.1 Synthesis of carbon nanotubes .................................................................................. 2

1.1.2 Structure of carbon nanotubes .................................................................................. 6

1.2 Computational methods for carbon nanotubes ................................................................. 11

1.2.1 Atomistic energy function based finite element method ........................................... 13

1.2.2 Formulation of shells and plates theory ................................................................. 21

1.2.3 $C^1$ thin shell finite element method ..................................................................... 26

1.2.4 Subdivision surfaces ............................................................................................... 27

1.3 Motivations ......................................................................................................................... 29

2 IMPLEMENTATION OF KIRCHHOFF-LOVE THIN SHELL THEORY .................................. 31

2.1 Kinematics of thin shells ............................................................................................... 31

2.2 Procedure of subdivision method .................................................................................. 35

2.2.1 Loop's triangular subdivision rule ......................................................................... 35

2.3 Validations for Kirchhoff-Love finite element method ..................................................... 54
5 VALIDATIONS AND NUMERICAL EXAMPLES ........................................ 119

5.1 Two dimensional carbon atomic ring ........................................ 119

5.2 Elastic properties of carbon nanotubes .................................... 124

5.3 Buckling of carbon nanotubes ................................................. 130

5.4 Twisting of carbon nanotube ................................................... 137

6 CONCLUSIONS AND FUTURE PLAN ........................................... 142

APPENDICES

A Picking matrix ........................................................................... 145

B Green-Lagrange strain tensor of the shell ...................................... 147

C Linearizing the bending strains .................................................. 149

D B-spline base functions .............................................................. 158

E Derivation for Green strain tensor ............................................... 160

BIBLIOGRAPHY ............................................................................. 161
## LIST OF TABLES

1.1 Comparison of Mechanical Properties. (ASSDA; Chae and Kumar) .......... 29

3.1 Parameters for REBO potential. .................................................. 88

3.2 Parameters for carbon-carbon Lennard-Jones potential. ................. 91
LIST OF FIGURES

1.1 Curled and branched single-walled carbon nanotubes under high magnification, reproduced from Harris (1999). .................................................. 2

1.2 Methods for synthesizing carbon nanotubes. ........................................ 4

1.3 Schematic of graphene honeycomb lattice with Bravais vectors $B^1$ and $B^2$. ............................................................................................................. 7

1.4 Unit cells for three typical kinds of carbon nanotubes (Reich et al., 2004). .................................................................................................................. 10

1.5 Shell geometry. .............................................................................................. 23

2.1 Shell geometry in the reference (left) and the deformed (right) configura­tion. .................................................................................................................... 32

2.2 Mask showing Loop’s rule for calculating the edge vertex (hollow circle, $\eta_\tau$), that is $\eta_\tau = \frac{1}{8} p + \frac{3}{8} r + \frac{3}{8} s + \frac{1}{8} q$. ..................................................................................................................... 36

2.3 The new vertex of the old vertex (the center one) after one subdivision step is determined by a linear relation with weights in dictated on the figure. ............................................................................................................. 36

2.4 The green triangle $\Delta 1^0 2^0 8^0$ is the irregular patch since the vertex $1^0$ has 7 valences while the rest two have 6. ......................................................... 37

2.5 New edge vertices generated anchoring vertex $2^0$. ..................................... 39

2.6 New edge vertices generated anchoring vertex $8^0$. .................................... 40

2.7 Left-over edge vertices ($14^1, 15^1$, and $16^1$) for vertex $2^0$ in pink shadow. ......................................................................................................................... 41

2.8 Left-over edge vertices ($17^1, 18^1$, and $19^1$) for vertex $8^0$ in purple shadow. ......................................................................................................................... 42

2.9 Effects of subdivision matrix $\tilde{A}$. .............................................................. 44
2.10 Unit triangle (parameter domain) constructed by two parameters $\theta_1$ and $\theta_2$. .......................................................... 45

2.11 The parameter domain is partitioned into an infinite set of triangular tiles (Stam, 1998). .......................................................... 46

2.12 The parameter domain is partitioned into an infinite set of triangular tiles. .......................................................... 50

2.13 A single regular triangular patch defined by 12 control vertices. ............. 51

2.14 Three new regular patches (shaded ones) are formed by one subdi­vision step. .......................................................... 52

2.15 1-neighbor of subpatch 1, 2, and 3 their numbering order. ................. 53

2.16 Definitions for plate and cylinder. .................................................. 57

2.17 Simulation results for a flat square plate. ........................................ 57

2.18 Simulation results for a pinched cylinder. ....................................... 58

2.19 Time variation of the central deflection of the clamped plate. .......... 62

2.20 Time variation of the central deflection of the simply-support plate. ....... 63

3.1 Schematics showing the linkage between atomic lattice and macros­scopic finite element method. ............................................... 65

3.2 Cauchy-Born rule applied to a cubic lattice. ...................................... 66

3.3 The representative crystallite used for carbon nanotube calculations. ...... 67

3.4 Direct use of Cauchy-Born rule leads to inaccurate results. ............... 68

3.5 Schematics showing exponential map. ............................................. 69

3.6 Schematics showing exponential map between two configurations. ........ 70

3.7 Relations for the grapheme sheet between undeformed and deformed configurations. ....................................................... 73

3.8 Lattice of grapheme sheet. ........................................................... 81

3.9 Undeformed carbon bonds. .......................................................... 83
3.10 Local structure of a CNT. ................................................................. 86
3.11 Apply REBO potential to representative cell. ................................. 88
3.12 Schematics showing Lennard-Jones potential calculation over the repre-
sentative cells. .................................................................................. 89
3.13 Modified Lennard-Jones potential. ................................................ 92
4.1 Mapping relations for finite element approximation. ....................... 94
4.2 Ghost mesh makes boundary nodes become regular. ...................... 95
4.3 Indexing order for a regular finite element. ................................... 96
5.1 Setup for the carbon ring/substrate interaction using molecular dy-
namics method. ............................................................................... 119
5.2 Setup for the carbon ring/substrate interaction using the general conti-
nuum method. .................................................................................. 120
5.3 van der Waals force between the ring and the substrate. ................ 121
5.4 The carbon ring under a point loading. ........................................ 123
5.5 Variation of Carbon nanotube strain energy (relative to graphene
sheet) with respect to radii. ............................................................... 125
5.6 Setup for the tensile test. ................................................................. 126
5.7 Variation of Carbon nanotube strain energy with respect to strain. .... 127
5.8 Stretched carbon nanotube. .............................................................. 129
5.9 The Young’s modulus based on wall thickness of 3.4 Å compared with
the values from (Arroyo and Belytschko, 2002). .................................. 131
5.10 Dramatic change in carbon nanotube’s morphological patterns. ....... 132
5.11 Buckled carbon nanotube from two different perspectives. ............ 133
5.12 The onset of the second instability. ................................................ 134
5.13 The buckling configurations for (7, 7), 60 Å carbon nanotube by mole-
cular dynamics simulation (Yakobson et al., 1996). ............................ 135
5.14 Strain energy variation with strain during buckling. ........................................... 136
5.15 The setup for twisting carbon nanotube. ............................................................... 137
5.16 The onset of the first instability and comparison with molecular dynamics simulation. ................................................................................................................................. 139
5.17 The side buckling of twisted carbon nanotubes and comparison with molecular dynamics simulation. ............................................................................................................ 140
5.18 Strain energy vs. twisting angles. ............................................................................ 141
Chapter 1
INTRODUCTION

In this chapter, a background introduction is provided for various aspects of carbon nanotubes (CNTs). First in section 1.1 a brief review about how carbon nanotubes are generated will be given as well as the scheme that is used to define the structure of CNTs. Numerical models ranging from high resolution molecular dynamics method to reduced order continuum based methods have emerged in efforts to study the mechanical behavior of CNTs. A review on the continuum based numerical methods for CNTs is given in section 1.2.1. A $C^1$ continuous finite element method has to be adopted to simulate very thin structures such as the CNTs considered in this dissertation which only have one layer of carbon lattice. A literature survey regarding these topics will be covered in section 1.2.2 and 1.2.3. Finally in section 1.2.4, the idea about subdivision surface is introduced, which was borrowed directly from the field of computational geometry where it was originated. Subdivision surfaces guarantee the sufficient smoothness in discretizing CNTs so that the finite element formulae with $C^1$ continuity won't break down. And finally the motivations for this study.
1.1 Fabrications and definitions of carbon nanotubes

1.1.1 Synthesis of carbon nanotubes

Carbon nanotube is a hollow tubule with a single or multiple layers. It can be seen as a structure closed by rolling up a layer of graphite sheet with a typical diameter for an individual tube of 1 nm. It is curled and looped rather than straight, (Harris, 1999). The observations of nanotube-like structure, especially the MWNTs have long been noted in literatures before 1991, the year which has been universally recognized as the inception of CNTs research. Harris (1999) listed several such carbon products that shared some common characteristics with CNTs.

A thread-like carbon structure obtained from the reaction of CO and $Fe_3O_4$ at 450 °C was reported in 1953 as a catalytic fabrication method. It is similar but not identical to CNTs in that it is usually contaminated by the catalyst during the production process and it does not have the same fullerene-like caps.
as CNT does. Graphite whiskers, the highly perfect form of carbon fibers were obtained first in 1960 by the technique that involves using a DC carbon arc, which is very similar to the modern way of synthesizing CNTs, but the macroscopic length scale (typically about 5 μm in diameter and up to 3 cm in length) in graphite whiskers distinguish themselves from CNTs. Another instance of observing CNTs was by Iijima himself in 1980s. He found some extraordinary structures scattered in a carbon support film which was prepared for electron microscopy. Those structures were believed to have concentric closed shells and were confirmed to be of tubular nature later by Iijima, but they do not have pentagonal rings.

It wasn’t until 1991 when Iijima (1991) in his paper recognized nanotubes with at least two graphitic layers as a new important form of carbon, especially in 1993 when he and other workers reported to have produced single-walled nanotubes (SWNTs) that people started to appreciate and study the unique properties of CNTs. The importance of SWNTs, see Fig. 1.1, is that they are truly new material and unlike MWNTs as discussed above, had never been observed in prior experiments.

There are many variations in producing CNTs aiming to improve the rate of yield, pureness and good alignment. The categorization of the synthesis of CNTs is normally based on whether that method uses catalysts or not.
(a) Illustration of arc-evaporation apparatus from Harris (1999).

(b) Illustration of vapor condensation method from Harris (1999).

(c) Chemical vapor deposition for organized assembly of CNTs from Wei et al. (2002).

Figure 1.2: Methods for synthesizing carbon nanotubes.
Fig. 1.2 demonstrates the basic ideas about three common techniques, first two of which are non-catalytic while the third one is catalytic. The arc-evaporation method (Fig. 1.2(a)) utilizes an arc generated under the discharge voltage of 20 V between two closely positioned graphite rods (the gap maintained at ≤ 1 mm). The chamber that houses both anode and cathode is filled with Helium gas modulated at a given pressure. The material of the anode that is made of pure carbon begins to evaporate under the current over the range of 50 – 100 A and recondenses in the form of cylindrical deposit onto the cathode. Hence the product of CNTs could be found on the cathode. Decisive parameters in the production process are the pressure of Helium, discharge current, and cooling of electrodes and the chamber. It's shown that the higher Helium pressure in the chamber the higher rate of yield of CNTs will be, and the optimum pressure is 500 Torr; the current however needs to be as low as possible in favor of generating more free nanotubes instead of sintered ones; efficient cooling of the whole apparatus is also important to prevent CNTs from getting melt down. A method that uses laser (of YAG or Nd-YAG type) rather than electric field to evaporate graphite is illustrated in Fig. 1.2(b) where a graphite target is held in an oven filled with Helium or Argon at a temperature of 1200 °C. The condensed material is collected on a cooled target which is usually made of

\[ 1 \text{ Torr} \approx 133.322 \text{ Pa} \]
graphite-Co-Ni, and is found to contain a significant proportion of nanotubes and nano particles with perfect structures. The problems with this method are that it is hard to control the chirality, length and diameter of CNTs and also the yielding rate is relatively low around 1.5 g/hr. An extremely fast pulse of laser and the use of porous target are needed to achieve a higher yielding rate, which is economically ineffective. The chemical vapor deposition (CVD) involves the decomposition of hydrocarbons such as methane, benzene, acetylene, naphthalene, etc over metal catalysts such as Co, Ni, Fe, Pt, and Pd that are deposited at predefined locations on a substrate such as Si. As shown in Fig.1.2(c), a template was first lithographed onto the silicon/silic substrates then expose it to a xylene/ferrocene ($C_8H_{10}/Fe(C_5H_5)_2$) vapor mixture at around 800 °C. Taking advantage of the preference of CNTs to grow selectively on and normal to silica surfaces, an orderly aligned assembly could be obtained. By this way the growth of CNTs could be somehow directed as designed.

1.1.2 Structure of carbon nanotubes

To mathematically describe carbon nanotubes, an SWNT can be looked at as a hollow tube with the thickness of one graphene sheet rolled up; an MWNT is composed of a series of concentric hollow tubes. The mean diameter $d_0$ for an SWNT is around 1.0 -1.5 nm; that number for an MWNT is around 5
and 100 nm and the inter-tube distance is 3.41 Å, (Reich et al., 2004). As briefly mentioned in section 1.1.1, the hallmark of CNTs lies in, but not limited to the fact that the electronic properties vary with the structural characteristics such as diameter, chirality and etc. Since graphene is nothing but a single, two-dimensional graphite, this very close relation with CNTs allows for the direct use of nomenclatures designed for carbon lattice.

Figure 1.3: Schematic of graphene honeycomb lattice with Bravais vectors $\mathbf{B}_1$ and $\mathbf{B}_2$.

Fig.1.3 demonstrates an undeformed flat honeycomb graphene sheet, each cell of which is perfectly hexagonal. The carbon atoms are located on the vertices of every hexagon whereas each edge represents the carbon-carbon bond. The bond angle formed by any two connecting bonds is 120° and the bond length is 1.42 Å. Graphene sheet is of translational symmetry and iden-
tical groups of carbon atoms are repeated in a regular pattern over the entire sheet. A unit cell in the lattice can be identified and is defined here as a surface spaned by two Bravais or basis vectors $B^1$ and $B^2$, shown as a small parallelogram on the lower left corner of Fig.1.3. As can be observed from Fig.1.3, for each graphene unit cell there are two carbon atoms and three inequivalent bonds. The length of the basis vectors by carbon lattice geometry is $\|B^1\| = \|B^2\| = B_0 = 2.461 \text{ Å}. Every carbon atom on the sheet can be represented distinctively by a pair of integers $(n_1, n_2)$. For an example, the longest vector in solid line on Fig.1.3 is the sum of two vectors that are parallel to the two bases respectively. The first adding vector is three times as long as the basis vector $B^1$, that is $3 B^1$ or $(3,0)$. While the length of the second adding vector is equivalent to six lengths of the basis vector $B^2$ or $6 B^2$, hence $(0,6)$. The resultant vector is then $3 B^1 + 6 B^2$ or $(3,6)$. A vector defined this way $C = n_1 B^1 + n_2 B^2$ is called the graphene lattice vector or the chiral vector in the context of CNT research. Graphene sheet is rolled up into the cylindrical shape along the direction of the chiral vector which thus becomes the circumference of the resulting tube.

There are infinite ways to roll a graphene sheet into a cylinder simply by following different lattice vectors. The resulting CNTs will have different chiralities and diameters as expected, which engenders dramatically different CNTs. As pointed out by Reich et al. (2004), although similar in diameters, the
(10, 10) CNT is metallic while the (10, 9) is semi-conductive for their having different chiral vectors or chiral angles. The angle between the chiral vector $C$ and one of the basis vectors $B^2$ is defined as the chiral angle, $\theta$, as demonstrated in Fig.1.3. The chiral angle can be calculated as follows

$$\cos \theta = \frac{B^2 \cdot C}{\|B^2\| \cdot \|C\|} = \frac{n_1/2 + n_2}{\sqrt{n_1^2 + n_1n_2 + n_2^2}}$$

(1.1)

For the uniqueness of using Eqn.1.1, it's necessary to have $n_1$ and $n_2$ satisfying the relation $0 \leq n_1 \leq n_2$, in other words requiring $0^\circ \leq \theta \leq 30^\circ$.

There are two special types of CNTs worth noting. If we roll the graphene sheet along the vector $(n, n)$ making the chiral angle of $30^\circ$ then an armchair CNT is formed, so named for its base line profile, see Fig.1.3. Another special kind of CNT is generated if rolled up along vector zig-zag, $(0, n)$. By substituting $(0, n)$ into Eqn.1.1, it is verified that the chiral angle $\theta$ is $0^\circ$. Both armchair and zig-zag CNTs are achiral while any tubes whose chiral angle is $\theta \in (0^\circ, 30^\circ)$ are chiral.

The diameter $d$ of a CNT is determined by its chiral vector $C$

$$d = \frac{\|C\|}{\pi} = \frac{B^0}{\pi \sqrt{n_1^2 + n_1n_2 + n_2^2}}$$

(1.2)

where $B^0$ is the carbon-carbon bond length. Similar to flat graphene sheet, CNTs also demonstrate translational symmetry as shown in Fig.1.4. The dark dots represents carbon atoms in the unit cell of carbon nanotubes. A trans-
Figure 1.4: Unit cells for three typical kinds of carbon nanotubes. (Reich et al., 2004)

The translational vector $a$ perpendicular to the tube axis and its length can be determined as follows

\[
a = \frac{2n_2 + n_1}{nR} B_1 + \frac{2n_1 + n_2}{nR} B_2
\]

(1.3)

\[
a = ||a|| = \sqrt{\frac{3(n_1^2 + n_1n_2 + n_2^2)}{nR}} B^0
\]

(1.4)

The number of carbon atoms in the unit cell, $n_c$, can be calculated from the ratio of the area of the cylinder surface, $S_t = a \cdot c$ to the area of the hexagon, $S_h$, where $c$ is the length of the chiral vector $c$. This ratio $q$ gives the number of hexagons in the unit cell,

\[
q = \frac{S_t}{S_h} = \frac{2(n_1^2 + n_1n_2 + n_2^2)}{nR}
\]

(1.5)

\[
\mathcal{R} = 3, \text{ if } (n_1 - n_2) / 3n \text{ is integer}
\]

(1.6)

\[
\mathcal{R} = 1, \text{ for all other values}
\]

(1.7)
n in Eqn.1.6 is the greatest common divisor of \((n_1, n_2)\). Then the number of carbons in the unit cell is

\[
n_c = 2q = \frac{4(n_1^2 + n_1n_2 + n_2^2)}{nR}
\] (1.8)

1.2 Computational methods for carbon nanotubes

Two major types of computational approaches to the simulation of mechanical properties of CNTs can be recognized in literature, that is, the classical molecular dynamics methods (MD) (e.g. Iijima et al., 1996) and the methods linking the atomistic constitutive law to the finite element framework (see Tadmor et al., 1996; Arroyo and Belytschko, 2002; Pantano et al., 2004b; Garg et al., 2007; Tadmor et al., 1999, for examples). It is well known that the MD method excels in modeling structural details of a crystal at the lattice level; however the computational cost is also known to be prohibitive for large systems. As pointed out by Tadmor et al. (1999) even the most ambitious atomic simulations using empirical potentials can only handle a system with the size on the order of \(10^9\) atoms, which corresponds to less than a few hundred nanometers in size. While the mean diameter for a MWNT is in the range of 5 to 100 \(nm\) and the length about several \(\mu m\) (Reich et al., 2004). Hence for the analysis of CNT nanocomposites which consists of a matrix material reinforced by fiber-
like carbon nanotubes, the MD method is disqualified as a good candidate for this end. A more intrinsic reason is that the most majority of the lattice deforms smoothly and closely obeys continuum elasticity. Yakobson et al. (1996) used the MD method to study carbon tubes under three basic mechanical loads: axial compression, bending, and torsion. They reported a remarkable synergism between the methods of MD and those of macroscopic structural mechanics. This observation has been shown and verified many times in literature. Arroyo and Belytschko (2005) used a two dimensional atomic rope deformation case to study the convergence property of atomistic based finite element methods (or continuum methods). It is shown that with the increase of the number of atoms on the rope, the continuum methods will converge to the parent atomic model in terms of the relative difference between the energies obtained from respective models. Consequently in this review, focus will be given to the continuum methods with the constitutive laws constructed upon atomistic energy functions.

The underlying finite element method is also of importance. Because curvature tensors of the deformed graphene sheet generally enters the formulae, the $C^1$ continuity is required for interpolation schemes. Subdivision finite element method is a new numerical paradigm aiming at simulating thin shell structures and maintaining $C^1$ continuity. This topic will also be reviewed in
the section.

1.2.1 Atomistic energy function based finite element method

The name follows the fact that all the methods described here marry the macroscopic structural similarities embedded in CNTs and the atomistic nature that distinguishes CNTs. Although there is no such a standard for categorizing many continuum methods present in literature, we can roughly give two names to label them as the general continuum method and elasticity model based continuum method.

In general continuum method, the constitutive law is built bottom-up from an atomistic energy function by intrinsic geometric quantities describing the deformation. It thus doesn't need any parameter fitting as required in the elasticity model method. The earliest work to the author's knowledge in this regard attributes to Tadmor et al. (1996). In the paper, they introduced a quasicontinuum method suitable for the analysis of defects in solids. Every point in the continuum is modeled by a representative atom embedded in a crystallite of radius $R_c$ and a finite element mesh is also generated for the same continuum. As a result every finite element will anchor an underlying crystallite with radius $R_c$. Then for every quadrature point in the mesh, the nearest atom is selected as a representative atom. As such the relative orientation for the finite
element mesh and the crystal lattice is set. The concepts of local and nonlocal formulations are introduced in the paper to account for the inhomogeneous phenomena of material such as stacking fault and dislocations. If the size of a finite element is larger than the crystallite radius $R_c$, then the homogeneity can be assumed for the element, in other words the deformation gradient $F$ is uniform over the element. The position of an atom in the element can be expressed by $x = FX$, where $x$ is the current position of the atom and $X$ is the original one. The status of the element is deemed local. The total strain energy is found by summing up the contributions from each local element, which in turn defines the first Piola-Kirchhoff stress tensor and the Lagrangian tangent stiffness tensor. The local formulation can thus be built upon those quantities. But when the crystallite radius is large enough to take more than one finite element, a local uniform deformation gradient $F$ is not available for describing the motion of atoms in the crystallite. The status of the element is considered to be nonlocal. For this case, the actual displacement of each atom according to the global continuum is needed for the construction of strain energy. Combining the local and the nonlocal formulations is effective in modeling material defects in that at grain boundaries where the inhomogeneous structural events take place the nonlocal formulation is called while at low strain area the local formulation is used.
Arroyo and Belytschko (2002, 2004, 2005) applied the quasicontinuum method to the study of mechanical response of CNTs. A carbon nanotube can be thought of as a rectangular graphene sheet being rolled up into a cylindrical shell without overlapping the edges. It is a curved lattice structure. The deformation of the carbon nanotube is assumed to be homogeneous so that all finite elements are local according to Tadmor et al. (1996). Hence the change in bond lengths and bond angles can be expressed in terms of the local deformation gradient \( \mathbf{F} \). The above statement can be summarized by the Cauchy-Born rule (Zanzotto, 1996) which postulates that under the deformation gradient \( \mathbf{F} \) about a macroscopic continuum, the lattice vectors behave as material vectors that undergo the same deformation as the continuum. But the deformed bond of a carbon nanotube is in fact the chord of a curved surface other than the tangent of it, which negates the validity of direct application of the Cauchy-Born rule \( \mathbf{w} = \mathbf{F} \mathbf{a} \) where \( \mathbf{w} \) is the tangent vector to the CNT surface about some atom and \( \mathbf{a} \) the undeformed carbon-carbon bond on the rectangular graphene sheet at the same atomic location. The exponential Cauchy-Born rule is proposed to correct this error, that is \( \mathbf{a} = \exp(\mathbf{F})\mathbf{a} \) where \( \mathbf{a} \) is the deformed carbon-carbon bond. The constitutive law is constructed by using appropriate atomistic energy function as well as the geometric quantities such as bond length, bond angle, principal curvature of the surface and etc., which characterize the deforma-
tion of the lattice. Since the homogeneity of deformation is assumed for carbon nanotubes, the relative orientation of finite elements and their underlying crystallite is immaterial. The strain energy density can be calculated based on a unit cell with convenient choice of location. With the method, the rippling effects of CNTs under bending and twisting loading conditions were faithfully reproduced in simulations. The continuum model of shell used in works of Arroyo and Belytschko (2002, 2004, 2005) was replaced by the intrinsic beam model to study the bulk dynamic responses of CNTs by Leamy (2007). The intrinsicality of the continuum model refers to the fact that other than displacements or rotations, only curvatures and strains are used to describe the configuration of CNTs, which yields the first-order partial differential governing equations with lower-order nonlinearity terms and allowing for the use of lower-order interpolation functions while maintaining the inter-element continuity. Because of the beam model used in the work, the 3-D characteristics of CNTs can not be obtained.

Another computational model belongs to the category of general continuum method was proposed by Chung and Namburu (2003) and it assumes that there co-exist in a structure two scales, a coarse one and a fine one; the displacements can be asymptotically decomposed into the coarse scale component and the perturbed displacement due to inhomogeneity at the fine scale.
Consequently two sets of coupled equations are obtained following the similar operation as other general continuum methods do. Then these two coupled equations are solved in iterative fashion. This method distinguishes itself from the quasicontinuum method proposed by Tadmor et al. (1996) in the way of how structural inhomogeneity is treated. In the quasicontinuum method, only regions with defects are considered nonlocal elements and the local elements apply to the rest of the structure, which leads to a unified framework. While Chung and Namburu's method assumes that every location can have inhomogeneous components and a system of coupled equations yields.

The family of elasticity model based continuum methods for CNTs are more dependent on the advantages bestowed by the remarkable similarity between atomic mechanics and macroscopic mechanics for lattice than the general continuum methods do. The implementation is thus simpler and results are in good agreement with higher-order schemes such as the MD method, especially for the cases where the material behavior is well understood and the deformation demonstrates not too much non-linearity. Many efforts have been devoted to the development of appropriate continuum elastic models for CNT simulations, such as using the nonlocal elastic beam (Wang and Hu, 2005) and the nested structural shell (Dong et al., 2008) models for wave propagation calculations in CNTs; simulations of multi-walled CNTs by the elastic shell model with
wall to wall surface interactions furnished by van der Waals energy potential (Pantano et al., 2004b); establishing effective wall thickness and Young's modulus amongst other equivalent elastic parameters for CNTs using truss and beam models by Odegard et al. (2002) and the 3-D elastic models used for modeling the CNTs embedded in a matrix (Chen and Liu, 2004). However all of these methods have to solve a dilemma i.e. the obscure definition of the length scale or the wall thickness since a single-walled CNT is in fact composed of a single atomic layer which doesn't have an obvious thickness. Without a properly defined wall thickness the equivalent elastic parameters such as the Young's modulus and shell flexural rigidity are meaningless. The solution is essentially based on the Cauchy-Born rule and the effective wall thickness is found by requiring the energy of a representative cell of the crystal to be in balance with that of an equivalent volume of continuous medium. The atomistic energy of a CNT has contributions from the bond stretching, bond-angle variation, dihedral angle torsion, out-of-plane angle bending, and non-bonded interactions. Many empirical parameter-fitting atomistic potential energy functions and quantum mechanics based formula with varying ability to represent the aforementioned components are available and are used to determine the effective wall thickness for CNTs. Slightly different elastic parameters for single-walled CNTs are reported in literature by different authors so the effective wall
thickness alone can not convey a self-contained description of the equivalent continuum model. Hence the effective wall thickness and the effective Young’s modulus has to be used in pair in order to correctly capture the atomistic details when equivalent continuum modeling for CNTs are implemented. For example the effective thickness and Young’s modulus obtained by molecular dynamics method are 0.066 nm and 5.5 TPa respectively (Yakobson et al., 1996) while those by tight binding method 0.074 nm and 5.1 TPa respectively (Zhou et al., 2000).

Pantano et al. (2004b,a, 2007); Garg et al. (2007) did a series of work on the mechanical and electromechanical behavior of single/multi-wall CNTs or CNT reinforced nanocomposite based on the equivalent continuum shell model. The effective wall thickness and Young’s modulus used are 0.075 nm and 4.84 TPa respectively. Great savings in numerical implementations can be achieved by resort to the commercial finite element software Abaqus so long as four conditions of modeling equivalence mandated in Pantano et al. (2004a) are met, that is, an appropriate set of elastic parameters, i.e. wall thickness and Young’s modulus; the existence of an initial stress in the CNT for the rolling of a flat graphene sheet; the non-bonded van der Waals interaction potential energy and a negligibly small wall-to-wall shear resistance. The van der Waals interactions occurring between two adjacent layers in a multi-walled CNTs or be-
tween two contacting CNTs or between a CNT and a substrate are incorporated into simulations through a special interaction element provided to Abaqus in the form of user-defined subroutine. The simulations demonstrate as good results as those from higher order models and provide an efficient alternative especially for large scale CNT structure studies. However the critics have also voiced in literature about the validity and reliability of the parameter fitting process used in determining the effective elastic parameters. For example Arroyo and Belytschko (2005) via their own numerical tests shows that although the elasticity model based continuum method is qualitatively correct and can capture the geometric nonlinearities, it doesn’t converge to the atomistic model and can’t capture the material nonlinearities arising from the atomistic potentials.

We are for the argument of Arroyo and Belytschko and the numerical method used in this thesis will be based on theirs. The underlying continuum model is the Kirchhoff thin shell which requires an overall $C^1$ surface continuity. The subdivision finite element method is to be used to discretize and represent the shell surface. In the following sections, literature reviews will be given for those topics.
1.2.2 Formulation of shells and plates theory

Thin structure is a special category in engineering designs and is widely used in many different areas ranging from some prominent large-scale applications such as airplane fuselage and the dome roof of sports arena to trendy small electronic gadgets like the enclosures of cell phone and MP3 player. The most noticeable feature of course is the thickness that is significantly smaller than its other two dimensions, which poses great challenges to finite element analysis. Although theoretically the full three dimensional governing equations should be applicable for thin structure calculations anyway, as pointed out by Belytschko et al. (2000), it's usually not numerically efficient and feasible because modelling such models requires a huge number of elements leading to a very expensive computation and the resulting mesh would be of high aspect ratio which will degrade the conditioning of the equations and the accuracy of the solution. Therefore in practice some specialized form equations are used instead.

People began to pursue appropriate formulae for shells and thin plates as early as in the nineteenth century. Two important figures in this area are Kirchhoff and Love, the first of who in 1850 derived the classic theory for bending of thin elastic plates under certain special assumptions while the second one contributing to the three-dimensional equations of linear elasticity for
shells in 1888, (Naghdi, 1972). Works of the two have become a cornerstone of the field of shells and plates. Although numerous successful applications of Kirchhoff-Love theory prove it useful and capable in solving practical problems, there have always been some voices in the area that doubt or concern the mathematical rigor of the theory for the way Kirchhoff and Love derived the equations. Since both formulated their respective equations by imposing some assumptions (now named the Kirchhoff-Love assumptions) about the plates and shells onto the general three dimensional elastic equations, a bloc of scholars such as Naghdi (1972); Simo and Fox (1989) believed that such derivations were not firmly established in mathematics. They preferred a more direct approach given the fact that the purpose of the theory of shells and plates is to provide an appropriate two-dimensional equations applicable to shell-like bodies, (Naghdi, 1972). The ground which the direct approach stands is the notion that the shell could be treated as a one-director Cosserat surface such that the laws of conservation and constitutive equations are to be applied locally, which leads to the direct derivation without resort to the general three dimensional elastic equations. The definition of a Cosserat surface by Naghdi (1972) is given as follows, for the complete account of the subject (see Naghdi, 1972, sec. 4)

A Cosserat surface also called a directed or an oriented surface is a body \( \mathcal{B} \) comprising a surface (embedded in a Euclidean 3-space)
and a single deformable director attached to every point of the surface.

In other words, by referring to Fig.1.5, a shell or a plate can be completely represented by a middle surface plus shell directors, the combination of which is the Cosserat surface. For simplicity, if we assume that the thickness of the shell is uniform and the middle surface is right in the middle although it’s not required to be so, then the position vector $r$ of any point on the shell can be determined by

$$r(\theta^1, \theta^2, \theta^3) = x(\theta^1, \theta^2) + \theta^3 a_3(\theta^1, \theta^2), \quad -\frac{h}{2} \leq \theta^1 \leq \frac{h}{2}$$

**Figure 1.5: Shell geometry.**

where $(\theta^1, \theta^2, \theta^3)$ is the convected curvilinear coordinates, $\theta^1$ and $\theta^2$ also serve
as the parametrization of the middle surface \( \mathbf{x}(\theta^1, \theta^2), \theta^3 \) indicates the normal coordinate to the middle surface, \((-\frac{h}{2} \leq \theta^3 \leq \frac{h}{2})\) for the current configuration and \(\mathbf{a}_3\) is the unit shell director which is in the same direction as \(\theta^3\) coordinate. Based on the Cosserat surface, the kinematics of a general shell or plate is established then the basic principles and constitutive relations are applied including conservation laws and invariance conditions. The direct approach is elegant but rather complex because it has to deal with a general differential shell geometry, which becomes even worse for nonlinear shells. However Naghdi (1972); Simo and Fox (1989) presented thin shell theory by direct approach based on solid mathematical foundations; the latter authors’ results were obtained by regarding the shell as an inextensible one-director Cosserat surface, which is especially useful for the implementation of subdivision finite element analysis.

The other side of the story is about those who stick with applying shell constraints to the already-derived three-dimensional equations. This method gains lot more popularities both in software industries and in academic circles because it is rather straightforward and can yield excellent results, (Belytschko et al., 2000). Either the term *degenerated continuum* approach or the term *continuum-based* (CB) approach is suitable to referring this type of method. We will follow the preference of Belytschko et al. (2000) by using CB instead of *degenerated continuum*. As Stanley (1985) pointed out that the CB shell equations
are derived from three-dimensional solid continuum theory with appropriate assumptions applied, which differs from the direct approach in that the intrinsic shell hypotheses are postponed until after the continuum equations have been fully derived. The direct approach on the other hand utilizes those assumptions as early as the laws of conservation are applied locally. CB method starts with a nonlinear full continuum equations, Eqn.1.10,

$$\frac{\partial \sigma_{ij}}{\partial x_j} + b_i = \rho u_i$$ \hspace{1cm} (1.10)

then linearize the variational version of it to reduce the nonlinear problem to an iterative sequence of linear problems, after that the shell assumptions are applied to the equations, then pre-integrate through the thickness to get a two-dimensional resultant form, which can be used to solve for the solution numerically. Two assumptions mentioned earlier are very important because they determine how the finite element analysis will be organized. By referring to Fig.1.5, assuming there are infinitely large amount of fibers passing the middle surface along the direction of shell director $a_3$ and the two ends of an individual fiber are on the top and bottom surfaces respectively then when the shell deforms those fiber will follow. The two assumptions prescribe how these fiber should behave under such deformations, (see Belytschko et al., 2000, chap. 9).
1. *Kirchhoff-Love shell theory* The fiber or the shell director $a_3$ during deformations remains straight and normal to the middle plane;

2. *Mindlin-Reissner shell theory* The fiber or the shell director $a_3$ during deformations only remains straight.

### 1.2.3 $C^1$ thin shell finite element method

It’s shown that the application of Kirchhoff-Love shell theory will have the second derivative of the velocity appearing in the expression for the rate of deformation and have the transverse shear vanished. This implies that the theory is most suitable for very thin shell-like structures and the velocity must have a continuous first order derivative or $C^1$ continuity; while the Mindlin-Reissner theory is applicable to thick shell or plate problems and since only the first derivative is present in the rate of deformation, the velocity needs only to be $C^0$ continuous or have a continuous velocity. Hence the Kirchhoff-Love shell theory may also be called $C^1$ theory to signify the fact it needs $C^1$ interpolation, likewise the Mindlin-Reissner shell theory as $C^0$ theory. The Mindlin-Reissner theory is more easier to implement in finite element coding than the Kirchhoff-Love theory because general unstructured meshes can not ensure $C^1$ continuity in the conventional sense of strict slope continuity across finite elements when the elements are defined with purely local polynomial shape functions, (Cirak
et al., 2000). Because of complicated nature of $C^1$ shell theory most commercial softwares avoid this trouble by using Mindlin-Reissner theory or by superimposing flat plate elements to simulate thin shells. Nevertheless for the extremely thin shells, many commercial softwares can not produce satisfactory results. A new finite element paradigm taking advantage of subdivision methods have been proposed and put into applications and reported very pleasing results (see Cirak et al., 2000; Cirak and Ortiz, 2001; Cirak et al., 2002, 2005). Cirak et al. (2000) formulated a $C^1$ thin shell functional by adopting the direct approach based on the work of Simo and Fox (1989) and imposing Kirchhoff-Love conditions. To overcome the $C^1$ continuity requirement for the unstructured mesh, he implemented the Loop’s subdivision algorithm (Loop, 1987) to discretize the shell surface. The resulting surface by this method is $C^2$ smooth overall and $C^1$ on isolated extraordinary points so that it is strictly within the framework of Kirchhoff-Love shell theory and at the same time satisfying all the convergence requirements of the displacement finite-element method.

1.2.4 Subdivision surfaces

Subdivision surfaces were developed in the area of computer-aided design by Catmull and Clark (1978); Doo and Sabin (1978) for a convenient representation for modeling objects of arbitrary topological type. Loop (1987) greatly
improved the method by proposing a triangular subdivision scheme based on quartic triangular B-splines, also known as quartic Box-splines. It is very valuable in describing complex geometric shapes because subdivision method in itself is a recursive process which starts from a set of control points gradually proceeding to the smooth limit surface. Loop (1987) proved in his thesis that after infinite iterations the triangulated surface will converge and the resulting surface has the curvature continuity ($C^2$ continuity) generally and tangent continuity ($C^1$ continuity) at isolated extraordinary points. This conclusion is very attractive to people in thin shell realm who suffered the lack of a tool that can produce such a well-behaved surface. As already stated in section 1.2.3, Kirchhoff-Love shell theory requires that the displacement have at least the continuous first derivative throughout (since evaluation of the second derivative is needed). But conventional low-order interpolation methods can only produce $C^0$ mesh. Although $C^1$ mesh could be obtained at the expense of high-order schemes, the numerical behavior of such schemes will be spurious (Cirak et al., 2000). Loop’s subdivision scheme provides a viable means to make it possible to directly apply Kirchhoff-Love theory to thin shells without any compromise.
1.3 Motivations

Thanks to the covalent $sp^2$ bonds formed between the individual carbon atoms, carbon nanotubes are the strongest and stiffest materials in world in terms of tensile strength and Young's modulus. The comparisons between two other materials are listed in Table 1.1. Because of the symmetry and unique electronic structure of graphene. The structure of a nanotube strongly affects its electrical properties. It can be metallic or semiconductor depending on nanotube's characteristic numbers ($n_1,n_2$). The metallic nanotubes can have an electrical current density more than 1,000 times greater than metals such as silver and copper. Carbon nanotubes find themselves to be of many potential uses in different fields. Such as in producing waterproof tear-resistant cloth fibers, combat jackets, sportswear, reinforced materials; in electrical circuits, it is hopeful to miniaturizing electronics beyond micro scale; in biomedical research, single-wall carbon nanotube has been tested for deliver cancer drugs to localized body area in mice (Schipper et al., 2008)

<table>
<thead>
<tr>
<th>Material</th>
<th>Young's Modulus (TPa)</th>
<th>Tensile Strength (GPa)</th>
<th>Max. Tensile Strain(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWNT</td>
<td>from 1 to 5</td>
<td>13 - 53</td>
<td>16</td>
</tr>
<tr>
<td>Stainless Steel</td>
<td>0.2</td>
<td>0.65 - 1</td>
<td>15</td>
</tr>
<tr>
<td>Kevlar</td>
<td>0.15</td>
<td>3.5</td>
<td>2</td>
</tr>
</tbody>
</table>
With all these exciting developments of carbon nanotubes, the numerical simulation techniques for this type of tiny objects remain a grey area to be further explored. Traditionally the molecular dynamics method is the obvious candidate, but more and more experimental and numerical results show that there are actually profound structural similarities between carbon nanotubes and continuum thin shell structures. Taking advantage of these similarities will lead to an efficient numerical framework that will benefit future numerical studies of large carbon nanotube systems.

In this thesis, the general continuum method that integrates the subdivision thin shell finite element method and the atomic hydrocarbon potential was developed, implemented and validated.
Chapter 2
IMPLEMENTATION OF KIRCHHOFF-LOVE THIN SHELL THEORY

The thin shell subdivision finite element method will serve as the numerical framework for the carbon nanotube simulation in this thesis. As a result, in this chapter, the implementation of the method will be covered and validated.

2.1 Kinematics of thin shells

As already discussed in section 1.2.2, the thin shell can be represented by the Cosserat surface i.e. the middle surface and its normal director. Shown in Fig.2.1 is a shell structure that undergoes the deformation \( F \) from the reference configuration to the current one. This process can be represented by

\[
\begin{align*}
\mathbf{r}(\theta^1, \theta^2, \theta^3) &= \mathbf{r}_{\text{middle surface}}(\theta^1, \theta^2) + \mathbf{r}_{\text{thickness}}(\theta^1, \theta^2) + \mathbf{a}_3(\theta^1, \theta^2), \quad -\frac{h}{2} \leq \theta^3 \leq \frac{h}{2} \\
\mathbf{r}(\theta^1, \theta^2, \theta^3) &= \mathbf{x}(\theta^1, \theta^2) + \mathbf{a}_3(\theta^1, \theta^2), \quad -\frac{h}{2} \leq \theta^3 \leq \frac{h}{2}
\end{align*}
\tag{2.1}
\]

Functions \( \mathbf{x}(\theta^1, \theta^2) \) and \( \mathbf{x}(\theta^1, \theta^2) \) furnish the parametric representations...
\[ a_{\alpha} = \tilde{x}_\alpha, \quad a_{\alpha} = x_{\alpha} \] (2.3)

where \( (\cdot)_{,\alpha} = \partial(\cdot) / \partial \theta_{\alpha} \) and \( \alpha \) runs from 1 to 2. The covariant components of the surface metric tensors are

\[ a_{\alpha\beta} = \tilde{a}_{\alpha} \cdot \tilde{a}_{\beta}, \quad a_{\alpha\beta} = a_{\alpha} \cdot a_{\beta} \] (2.4)

then the corresponding contravariant components are the inverse of Eqn.2.4, i.e. \( a^{\alpha\gamma}a_{\gamma\beta} = \delta^\alpha_\beta \), and \( a^{\alpha\gamma}a_{\gamma\beta} = \delta^\alpha_\beta \). The Jacobian for the embedded \((\theta^1, \theta^2)\) plane relative to the Euclidean space \((x^1, x^2, x^3)\) is \( d\Omega = \sqrt{a_{1}\theta^1 d\theta^2} = \|\tilde{a}_1 \times \)
\( \mathbf{a}_2 \| d\theta^1 d\theta^2 \). The shell director \( \mathbf{a}_3 \) for the reference configuration should be the unit vector normal to the middle surface as shown in Fig.2.1 and is given by

\[
\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\| \mathbf{a}_1 \times \mathbf{a}_2 \|}
\] (2.5)

How would the shell director be like in the deformed configuration? As discussed in the end of section 1.2.2, there are two assumptions in thin shell theory. Here we will adopt the first one, i.e. the Kirchhoff-Love assumptions, which states that during deformation the shell director remains straight and normal to the middle surface. Hence the shell director for the current configuration is

\[
\mathbf{a}_3 = \frac{\mathbf{a}_1 \times \mathbf{a}_2}{\| \mathbf{a}_1 \times \mathbf{a}_2 \|}
\] (2.6)

So far all the quantities related to the Cosserat surface have been derived using the direct approach, that is we didn’t begin with some general three-dimensional equation followed by imposing constraints about the shell. Instead we directly started out from the shell structure, i.e. the Cosserat surface and proceeded to obtain necessary metrics. Now by Eqn.2.1 and Eqn.2.2, we can derive the covariant base vectors for shell body in the reference and the current configurations
where \( a \) goes from 1 to 2. The corresponding covariant metrics of the shell body are simply the inner products of base vectors, \( \bar{g}_{ij} = \bar{g}_i \cdot \bar{g}_j \) and \( g_{ij} = g_i \cdot g_j \), where \( i \) and \( j \) run from 1 to 3. By definition the Green-Lagrange strain tensor can be found by \( E_{ij} = \frac{1}{2} (g_{ij} - \bar{g}_{ij}) \), to the first order in the shell thickness \( h \) it can be shown that the Green-Lagrange strain of the shell is of the following form

\[
E_{ij} = \alpha_{ij} + \theta^3 \beta_{a\beta}
\]

where \( \alpha_{ij} = \frac{1}{2} (a_i \cdot a_j - \bar{a}_i \cdot \bar{a}_j) \) and \( \beta_{a\beta} = a_a \cdot a_{3,\beta} - \bar{a}_a \cdot \bar{a}_{3,\beta} \) and \( i, j \) from 1 to 3; \( a, \beta \) from 1 to 2. The detailed derivation can be found in Appendix B. The in-plane components \( \alpha_{a\beta} \), or the membrane strains, measure the straining of the surface; the components \( \alpha_{a3} \) measure the shearing of the director \( \bar{a}_3 \); the component \( \alpha_{33} \) measures the stretching of the director; and the component \( \beta_{a\beta} \), the bending strains, measure the bending or change in curvature of the shell.

If only consider linear condition, then

\[
x(\theta^1, \theta^2) = \bar{x}(\theta^1, \theta^2) + u(\theta^1, \theta^2)
\]
be linearized as

\[ \alpha_{\alpha\beta} = \frac{1}{2} (\bar{a}_\alpha \cdot u_{\beta} + u_{\alpha} \cdot \bar{a}_\beta) \quad (2.11) \]

\[ \beta_{\alpha\beta} = \frac{d}{dc}[\beta(x + cu)]_{c=0} = -u_{\alpha\beta} \cdot \bar{a}_3 + \frac{1}{\sqrt{d}}[u_{1\alpha} \cdot (\bar{a}_{3\beta} \times \bar{a}_2) + u_{2\beta} \cdot (\bar{a}_1 \times \bar{a}_{\alpha\beta})] + \frac{\bar{a}_3 \cdot \bar{a}_{\alpha\beta}}{\sqrt{d}} [u_{1\alpha} \cdot (\bar{a}_2 \times \bar{a}_3) + u_{2\beta} \cdot (\bar{a}_3 \times \bar{a}_1)] \quad (2.12) \]

For detailed steps about linearizing \( \beta_{\alpha\beta} \) to Eqn.2.12, see Appendix C.

From linearized Eqn.2.11 and Eqn.2.12, it’s found that the only unknown is the displacement of middle surface \( u \), which will be the primary unknown in the following finite element implementation.

### 2.2 Procedure of subdivision method

#### 2.2.1 Loop’s triangular subdivision rule

As reviewed in section 1.2.4, Loop’s rule (Loop, 1987) is the most pronounced triangular subdivision scheme. It is in fact an approximating method that calculates the positions of new vertices from unrefined ones by easy linear relations. The masks for calculating the edge and the center vertex using the rule are shown in Fig.2.2, and Fig.2.3.

The subdivision matrix \( A \) stores the information about how the subdi-
Figure 2.2: Mask showing Loop's rule for calculating the edge vertex (hollow circle, $\eta_{rs}$), that is $\eta_{rs} = \frac{1}{8} p + \frac{3}{8} q + \frac{3}{8} s + \frac{1}{8} q$.

Figure 2.3: The new vertex of the old vertex (the center one) after one subdivision step is determined by a linear relation with weights indicated on the figure. $N$ represents the number of edges (or the valence) incident on the old vertex (the center one). In the mathematic form, $x_{1}^{k+1} = (1 - Nw)x_{1}^{k} + wx_{2}^{k} + \cdots + wx_{N+1}^{k}$ where $x_{1}^{k+1}$ is the new position of the old vertex, and $k$ represents the subdivision level. For an irregular vertex, $w = \frac{3}{8} \left[ \frac{5}{8} - \left( \frac{3}{8} + \frac{1}{4} \cos \frac{2\pi}{N} \right)^{2} \right]$, while for regular vertices $w = \frac{3}{8N}$ for $N > 3$; $w = \frac{3}{16}$ for $N = 3$.

vision operation should proceed in regard to a particular patch which could be regular or irregular\(^1\). It is crucial to establish correct matrix $A$ for each patch in the mesh. Let's exam a mesh surrounding an irregular patch (in green) as shown in Fig.2.4. One vertex in the green patch has seven bonds incident on it.

\(^1\)After one subdivision operation over the original mesh, it's guaranteed that each patch (triangle) will have at most one vertex which has the valence other than six. Irregular patch is the one that has one such vertex. Every vertex in a regular patch has exactly six valences.
or seven valences.

![Diagram of a triangle with vertices labeled 1°, 2°, 8°, and 10°.]

**Figure 2.4:** The green triangle $\triangle 1^0 2^0 8^0$ is the irregular patch since the vertex $1^0$ has 7 valences while the rest two have 6. In the current subdivision implementation, we only consider 1-neighbor ring of the patch. First set up the patch of interest, the green one in this case. The irregular vertex is labeled as $1^0$ from which five edges (plus the other two in the green triangle equal to 7) are drawn. Then move on to the next vertex $2^0$ in the patch to draw four edges plus two in the triangle equal to 6 which makes it regular vertex. Same procedure applies to the third vertex ($8^0$) in the triangle. Finally connecting all the free ends of the valences so the configuration of the above mesh is obtained excluding the red lines. New vertices and edges with respect to vertex 1. The superscript 0 represents the unrefined level, while the superscript 1 represents the refined level after one subdivision step. Parallel to the generation of the original mesh, the subdivision also begins from vertex $1^0$ then $2^0$ then $8^0$.

**Subdivision matrix generation step by step**

By referring to Fig.2.4, we can see that the application of Loop’s rule to $1^0$ will generate seven new edge vertices according to Fig.2.2, and the center (irregular) vertex $1^0$ will also assume a new position by the rule shown in Fig.2.3.
The numbering scheme is shown in Fig. 2.4.

\[ x_2^1 = \frac{x_0^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]
\[ x_3^1 = \frac{x_2^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]
\[ x_4^1 = \frac{x_2^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]
\[ x_5^1 = \frac{x_2^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]
\[ x_6^1 = \frac{x_0^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]
\[ x_7^1 = \frac{x_0^0 + x_3^0 + 3x_6^0 + 3x_7^0}{8} \]

Eqn. 2.13 according to Fig. 2.2 demonstrates relations between the new edge vertices and the unrefined ones. For the center vertex refinement, referring to Fig. 2.3, it is

\[ x_1^1 = (1 - Nw)x_1^0 + wx_2^0 + wx_3^0 + wx_4^0 + wx_5^0 + wx_6^0 + wx_7^0 + wx_8^0 \]  

If Eqn. 2.13 and Eqn. 2.14 are expressed in the matrix form, it becomes

\[
\begin{bmatrix}
  x_1^1 \\
  x_2^1 \\
  x_3^1 \\
  x_4^1 \\
  x_5^1 \\
  x_6^1 \\
  x_7^1 \\
  x_8^1
\end{bmatrix}
= \begin{bmatrix}
  (1 - Nw) & w & w & w & w & w & w \\
  3/8 & 3/8 & 1/8 & 0 & 0 & 0 & 1/8 \\
  3/8 & 1/8 & 3/8 & 1/8 & 0 & 0 & 0 \\
  3/8 & 0 & 1/8 & 3/8 & 1/8 & 0 & 0 \\
  3/8 & 0 & 0 & 1/8 & 3/8 & 1/8 & 0 \\
  3/8 & 0 & 0 & 0 & 1/8 & 3/8 & 1/8 \\
  3/8 & 0 & 0 & 0 & 0 & 1/8 & 3/8 \\
  3/8 & 1/8 & 0 & 0 & 0 & 1/8 & 3/8
\end{bmatrix}
\begin{bmatrix}
  x_1^0 \\
  x_2^0 \\
  x_3^0 \\
  x_4^0 \\
  x_5^0 \\
  x_6^0 \\
  x_7^0 \\
  x_8^0
\end{bmatrix}
\]  

Again by using rules described in Fig. 2.2 and Fig. 2.3, we will have the
Figure 2.5: New edge vertices generated anchoring vertex $2^0$. The important issue is to correctly register new vertices. For the sake of simplicity, let’s currently only consider new vertices contributing to the 1-neighbor of the irregular vertex $1^0$ by subdividing the neighborhood of vertex $2^0$ shown in the light yellow shadow, vertices $9^1, 10^1$ which is the new position of vertex $2^0$, and $11^1$.

The following relations, Eqn.2.16 for the new vertices corresponding to vertex $2^0$ by referring to Fig.2.5

\[
x_7^1 = \frac{x_0^0 + x_1^0 + 3x_2^0 + 3x_3^0}{8}
\]

\[
x_{11}^1 = \frac{x_0^0 + x_{10}^0 + 3x_8^0 + 3x_9^0}{8}
\]

\[
x_{10}^1 = (1 - 6 \times \frac{3}{8 \times 6}) x_0^0 + \frac{1}{16} x_1^0 + \frac{1}{16} x_3^0 + \frac{1}{16} x_{11}^0 + \frac{1}{16} x_{10}^0 + \frac{1}{16} x_9^0 + \frac{1}{16} x_8^0
\] (2.16)

The transformation equations for vertex $8^0$, refer to Fig.2.6, are as follows

\[
x_{13}^1 = \frac{x_0^0 + x_{12}^0 + 3x_7^0 + 3x_8^0}{8}
\]

\[
x_{12}^1 = (1 - 6 \times \frac{3}{8 \times 6}) x_8^0 + \frac{1}{16} x_1^0 + \frac{1}{16} x_7^0 + \frac{1}{16} x_{13}^0 + \frac{1}{16} x_{12}^0 + \frac{1}{16} x_9^0 + \frac{1}{16} x_2^0
\] (2.17)
Figure 2.6: New edge vertices generated anchoring vertex $8^0$. Similar to Fig.2.5, let's currently only consider new vertices contributing to the 1-neighbor of the irregular vertex $1^0$ by subdividing the neighborhood of vertex $8^0$ shown in the blue shadow, vertices $13^1, 12^1$ which is the new position of vertex $8^0$.

Now based on Eqn.2.15, 2.16, and 2.17, we could form a big subdivision matrix $A$ producing a set of refined vertices $(1^1, 2^1, 3^1, 4^1, 5^1, 6^1, 7^1, 8^1, 9^1, 10^1, 11^1, 12^1, 13^1)$ as shown in Fig.2.6. The detailed entries of this matrix $A$ are
Figure 2.7: Left-over edge vertices (14, 15, and 16) for vertex 2 in pink shadow. They need to be incorporated into the refinement also which constitutes the new entries in the matrix A.

Matrix A can be divided into four parts with the upper left being denoted as S,
the upper right as $O$, the lower left as $S_{11}$, and the lower right as $S_{12}$ following the fashion of Stam (1998). After finishing this basic subdivision matrix $A$, we are ready to consider an extended subdivision which accounts for the refinement of more vertices related to vertex $2^0$ and $8^0$. By referring to Fig.2.7 and Fig.2.8 and following the rules shown in Fig.2.2, we could have the following equations for building new vertices $14^1$ through $19^1$,

$$
\begin{align*}
    x_{14}^1 &= \frac{x_9^0 + x_{14}^0 + 3x_7^0 + 3x_8^0}{8} \\
    x_{15}^1 &= \frac{x_9^0 + x_{15}^0 + 3x_7^0 + 3x_{10}^0}{8} \\
    x_{16}^1 &= \frac{x_9^0 + x_{16}^0 + 3x_7^0 + 3x_{11}^0}{8}
\end{align*}
$$

(2.18)

Figure 2.8: Left-over edge vertices ($17^1$, $18^1$, and $19^1$) for vertex $8^0$ in purple shadow. They need to be incorporated into the refinement also which constitutes the new entries in the matrix $A$. 

42
\[ x_{17} = \frac{x_{1}^6 + x_{12}^6 + 3x_{12}^6 + 3x_{12}^6}{8} \]
\[ x_{18} = \frac{x_{1}^6 + x_{12}^6 + 3x_{12}^6 + 3x_{12}^6}{8} \]
\[ x_{19} = \frac{x_{1}^6 + x_{12}^6 + 3x_{12}^6 + 3x_{12}^6}{8} \]  

(2.19)  

Now we can add the six lines to the basic matrix \( A \). After rearrangements, we could get the extended subdivision matrix \( \tilde{A} \) as follows,

\[
\begin{bmatrix}
    3/8 & 3/8 & 1/8 & 0 & 0 & 0 & 0 & \frac{1}{8} \\
    3/8 & 3/8 & 1/8 & 0 & 0 & 0 & 0 & 0 \\
    3/8 & 3/8 & 1/8 & 0 & 0 & 0 & 0 & 0 \\
    3/8 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\
    3/8 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\
    3/8 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\
    3/8 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\
    3/8 & 0 & 0 & 0 & 1/8 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
    (1 - Nw) & w & w & w & w & w & w & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\begin{bmatrix}
    x_0^1 \\
    x_0^2 \\
    x_0^3 \\
    x_0^4 \\
    x_0^5 \\
    x_0^6 \\
    x_0^7 \\
    x_0^8 \\
\end{bmatrix}
\begin{bmatrix}
    x_0^0 \\
    x_0^1 \\
    x_0^2 \\
    x_0^3 \\
    x_0^4 \\
    x_0^5 \\
    x_0^6 \\
    x_0^7 \\
\end{bmatrix}
\]

This extended matrix \( \tilde{A} \) according to Stam (1998) when cast into block
form becomes

$$\bar{A} = \begin{bmatrix} S & O \\ S_{11} & S_{12} \\ S_{21} & S_{22} \end{bmatrix}$$

(2.20)

So up to this point, the derivation of the subdivision matrix $A$ and $\bar{A}$ about an patch involving one irregular vertex (with 7 valences) has been completed. It still however needs generate subdivision matrices for regular matrices and irregular matrices with different valences. It’s better write a programming subroutine to perform this task. If the number of bonds incident on the irregular node is denoted as $N$ then the dimension of $\bar{A}$ is $(N+12, N+6)$. The

![Diagram of mesh subdivision](image)

(a) The coarse mesh for a rectangular area. (b) The refined mesh after one subdivision operation.

**Figure 2.9:** Effects of subdivision matrix $\bar{A}$.

The effect of subdivision operation about an original mesh is shown in Fig. 2.9, which refines the mesh and isolates the irregular vertices so that every triangle
only has at most one irregular vertex.

Coordinate transformation between unit triangular patch and refined ones.

As discussed in the paper by Stam (1998), using Loop's subdivision finite element method actually means to interpolate the displacement over a regular patch with twelve spline basis functions, i.e. \( u_h(\bar{\theta}^1, \bar{\theta}^2) = \sum_{i=1}^{12} N_i(\bar{\theta}^1, \bar{\theta}^2) u_i \).

The expressions for base functions \( N_i \) can be found in appendix D. The patch can be considered as the surface parametrized over a unit triangle \( \Omega \) with two parameters \( \bar{\theta}^1, \bar{\theta}^2 \). This relation is written for the unit triangle as shown in Fig. 2.10.

The surface expressed in group language is as follows

\[
\Omega = \left\{ (\bar{\theta}^1, \bar{\theta}^2) | \bar{\theta}^1 \in [0,1] \cap \bar{\theta}^2 \in [0,1-\bar{\theta}^1] \right\}
\]  

\( \theta_2 = 1 - \theta_1 \)

**Figure 2.10:** Unit triangle (parameter domain) constructed by two parameters \( \theta_1 \) and \( \theta_2 \).
We seek a parametrization \( s(\tilde{\theta}^1, \tilde{\theta}^2) \) for triangular surface, the patch for all \( (\tilde{\theta}^1, \tilde{\theta}^2) \in \Omega \). As shown in fig.2.11, we can partition the parameter domain

\[
\begin{align*}
\text{Figure 2.11: The parameter domain is partitioned into an infinite set of triangular tiles. (Stam, 1998)}
\end{align*}
\]

into an infinite set of tiles \( \Omega^n_k \), with \( n \geq 1 \) and \( k = 1, 2, 3 \). Remember the right-angle corner is irregular hence we successively divide the triangle into four patches three of which are regular labelled as \( \Omega^1, \Omega^2 \), and \( \Omega^3 \). These three regular patches can be efficiently evaluated using \( u_k(\theta^1, \theta^2) = \sum_{l=1}^{12} N^l(\theta^1, \theta^2) u_l \). However the shape functions \( N^l(\theta^1, \theta^2) \) were calculated based upon the unit triangle shown in Fig.2.10, so it becomes necessary to map tile \( \Omega^n_k (\theta^1, \theta^2) \) in Fig.2.11 onto unit tile \( \Omega (\tilde{\theta}^1, \tilde{\theta}^2) \) in Fig.2.10. We have seen from Fig.2.10 that

\[
\begin{align*}
0 \leq \tilde{\theta}_1 & \leq 1 \\
0 \leq \tilde{\theta}_2 & \leq 1 - \tilde{\theta}_1
\end{align*}
\]
• Now first examine triangle $\Omega_1$ in Fig. 2.11, where

\[
\left(\frac{1}{2}\right)^n \leq \theta^1 \leq 2\left(\frac{1}{2}\right)^n
\]
\[
0 \leq \theta^2 \leq 2 \left(\frac{1}{2}\right)^n - \theta^1
\]

To transform coordinates, we make the following steps so that finally ranges of $\bar{\theta}^1$ and $\bar{\theta}^2$ will coincide with that of $\theta^1$ and $\theta^2$, for $\theta^1$

\[
2^n \times \left(\frac{1}{2}\right)^n \leq 2^n \times \theta^1 \leq 2^n \times 2 \left(\frac{1}{2}\right)^n
\]
\[\implies 1 \leq 2^n \theta^1 \leq 2 \quad (2.23)\]
\[\implies \quad 1 - 1 \leq 2^n \theta^1 - 1 \leq 2 - 1 \quad (2.24)\]
\[\implies \quad 0 \leq \frac{2^n \theta^1 - 1}{\bar{\theta}^1} \leq 1 \quad (2.25)\]

Hence we have obtained the first coordinate transformation rule $\bar{\theta}^1 = 2^n \theta^1 - 1$.

For $\theta^2$,

\[
0 \leq 2^n \times \theta^2 \leq 2^n \times 2 \left(\frac{1}{2}\right)^n - 2^n \times \theta^1
\]
\[
0 \leq 2^n \theta^2 \leq 2 - 2^n \theta^1 = 1 - (2^n \theta^1 - 1) \quad (2.27)\]
\[\implies \quad 0 \leq \frac{2^n \theta^2}{\bar{\theta}^2} \leq 1 - \bar{\theta}^1 \quad (2.28)\]

Hence the second transformation rule is $\bar{\theta}^2 = 2^n \theta^2$.

• We now look at triangle $\Omega_2$ as shown in Fig. 2.11, where
\[
0 \leq \theta^1 \leq \left(\frac{1}{2}\right)^n
\]
\[
\left(\frac{1}{2}\right)^n - \theta^1 \leq \theta^2 \leq \left(\frac{1}{2}\right)^n
\]

Since \( \Omega_2 \) is in the opposite orientation of the unit triangle, we made some tricky steps. For \( \theta^1 \),

\[
2^n \times 0 \leq 2^n \times \theta^1 \leq 2^n \times \left(\frac{1}{2}\right)^n
\]

\[
\Rightarrow
0 \leq 2^n \theta^1 \leq 1
\]

\[
\Rightarrow
-1 \leq -2^n \theta^1 \leq 0
\]

\[
\Rightarrow
-1 + 1 \leq -2^n \theta^1 + 1 \leq 0 + 1
\]

\[
\Rightarrow
0 \leq 1 - 2^n \theta^1 \leq 1
\]

Hence the first transformation rule for \( \Omega_2 \) is \( \theta^1 = 1 - 2^n \theta^1 \).

For \( \theta^2 \),

\[
-1 \times \left[ \left(\frac{1}{2}\right)^n - \theta^1 \right] \geq -1 \times \theta^2 \geq -1 \times \left(\frac{1}{2}\right)^n
\]

\[
\Rightarrow
-\left(\frac{1}{2}\right)^n \leq -\theta^2 \leq \theta^1 - \left(\frac{1}{2}\right)^n
\]

\[
2^n \times \left[ -\left(\frac{1}{2}\right)^n \right] \leq 2^n \times \left( -\theta^2 \right) \leq 2^n \times \left[ \theta^1 - \left(\frac{1}{2}\right)^n \right]
\]

\[
\Rightarrow
-1 \leq -2^n \theta^2 \leq 2^n \theta^1 - 1
\]
\[ 1 - 1 \leq 1 - 2^n \theta^2 \leq 1 + 2^n \theta^1 - 1 \quad (2.38) \]
\[ 0 \leq 1 - 2^n \theta^2 \leq 2^n \theta^1 \quad (2.39) \]

Since \( \bar{\theta}^1 = 1 - 2^n \theta^1 \), \( 2^n \theta^1 = 1 - \bar{\theta}^1 \). We have

\[ 0 \leq 1 - 2^n \theta^2 \leq 1 - \bar{\theta}^1 \quad (2.40) \]

Thus the second transformation rule for \( \Omega_2 \) is \( \bar{\theta}^2 = 1 - 2^n \theta^2 \).

• Now let's examine triangle \( \Omega_3 \) in Fig.2.11, where

\[ 0 \leq \theta^1 \leq \left( \frac{1}{2} \right)^n \]
\[ \left( \frac{1}{2} \right)^n \leq \theta^2 \leq 2 \left( \frac{1}{2} \right)^n - \theta^1 \]

For \( \theta^1 \)

\[ 2^n \times 0 \leq 2^n \times \theta^1 \leq 2^n \times \left( \frac{1}{2} \right)^n \quad (2.41) \]
\[ \Rightarrow \]
\[ 0 \leq \frac{2^n \theta^1}{\bar{\theta}^1} \leq 1 \quad (2.42) \]

The first coordinate transformation rule for \( \Omega_3 \) is \( \bar{\theta}^1 = 2^n \theta^1 \). For \( \theta^2 \),

\[ 2^n \times \left( \frac{1}{2} \right)^n \leq 2^n \times \theta^2 \leq 2^n \times 2 \left( \frac{1}{2} \right)^n - 2^n \times \theta^1 \quad (2.43) \]
\[ \Rightarrow \]
\[ 1 \leq 2^n \theta^2 \leq 2 - 2^n \theta^1 = 1 - (2^n \theta^1 - 1) \quad (2.44) \]
Again since $\tilde{\theta}^1 = 2^n \theta^1$, we'll have

$$0 \leq \frac{2^n \theta^2 - 1}{\tilde{\theta}^2} \leq 1 - \tilde{\theta}^1$$

(2.47)

Hence the second transformation rule for $\Omega_3$ is $\tilde{\theta}^2 = 2^n \theta^2 - 1$.

For the purpose of clarity, we put Fig.2.11 here again. To sum up the above derivations for coordinate transformations between refined triangles and unit triangle, we have the following rules denoted as $t_{n,\Omega_k}(\theta^1, \theta^2)$ where $n$ represents the subdivision levels, $n = 1, 2, 3, ...$ and $k$ represents three regular triangles generated at each subdivision step, $k = 1, 2, 3,$

\begin{figure}[h]
\centering
\includegraphics[width=0.3\textwidth]{figure.png}
\caption{The parameter domain is partitioned into an infinite set of triangular tiles.}
\end{figure}

50
\[ t_{n,\Omega_1}(\theta^1, \theta^2) = (2^n \theta^1 - 1, 2^n \theta^2) \]  
(2.48)

\[ t_{n,\Omega_2}(\theta^1, \theta^2) = (1 - 2^n \theta^1, 1 - 2^n \theta^2) \]  
(2.49)

\[ t_{n,\Omega_3}(\theta^1, \theta^2) = (2^n \theta^1, 2^n \theta^2 - 1) \]  
(2.50)

**Forming the picking matrix**

When dealing with irregular patches, to be able to use B-spline shape functions, it’s necessary to subdivide the original patch until the evaluation point is located within a regular subpatch. Then the B-spline shape functions could be evaluated on the twelve surrounding vertices. These twelve nodal points are formed by the picking matrix according to the order shown in Fig.2.13. In this section we will only consider 1-neighbor vertices generated after one subdivision as this is the case applicable to the current study.

![Figure 2.13: A single regular triangular patch defined by 12 control vertices.](image)

As shown in Fig.2.14, the newly-generated 1-neighbor control vertices
Three new regular patches (shaded ones) are formed by one subdivision step. For these regular patches, B-spline basis functions could be applied to twelve vertices to represent the surface.

about the original triangle can be listed in the matrix format,

\[ \mathbf{X} = (x_{1}, x_{2}, x_{3}, x_{4}, x_{5}, x_{6}, x_{7}, x_{8}, x_{9}, x_{10}, x_{11}, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}, x_{19}) \]

(2.51)

Eqn.2.51 includes all new vertices shown in Fig.2.14 with the total of 19. For the three regular patches, there have to be three subsets\(^2\) of twelve control vertices formed by extracting elements properly from \(\mathbf{X}^T\) in Eqn.2.51. When comparing the ordering schemes between Fig.2.14 and Fig.2.13, it's obvious that they are quite different. Hence our next task is to sort out subsets from Eqn.2.51 for each patch according to Fig.2.13 by resort to a so called picking matrix \(P\). To better explain the issue, we list 1-neighbor configuration with 12 control vertices for each regular patch with their respective labelling un-

\(^2\)Of course there will be overlaps among 3 subsets of control vertices.
changed shown in Fig. 2.15. Now we need to define the picking matrix $P_k$

![Subpatch Images](image)

**Figure 2.15**: 1-neighbor of subpatch 1, 2, and 3 and their numbering order.

where $k = 1, 2, 3$ representing subpatch 1, 2, and 3, such that ultimately we could have a vector $\mathbf{b}_1$ with its elements listed in exact conformity with the order shown in Fig. 2.13, i.e. taking subpatch 1 for an example Fig. 2.15(a), $\mathbf{b}_1^T = (x_3, x_1, x_{11}, x_2, x_8, x_{16}, x_{10}, x_9, x_{12}, x_{15}, x_{14}, x_{17})$. By doing this, we could correctly evaluate $s_1(\theta^1, \theta^2) = \sum_{l=1}^{12} [N^I(\theta^1, \theta^2)]^T b_l$. Each row of $P_k$ is filled with zeros except for a one in the column corresponding to the index shown in Fig. 2.15, in other words, again taking subpatch 1 for an example Fig. 2.15(a), the row stacking sequence is determined by the position of each vertex in Fig. 2.15(a) relative to Fig. 2.13 and the nontrivial (nonzero) element of each row is located by the index of corresponding vertex in Fig. 2.15(a). For example, in ordering vertex labelled as 3 in Fig. 2.15(a), it is the first vertex in Fig. 2.13 so the corresponding row should be the first one of matrix $P_1$, and since its index in Fig. 2.15(a) is 3.
therefore the only nonzero element (its value is one) of this row should be in the third column. For the full picking matrix for subpatch 2 see Appendix A. Again the picking matrices shown here is for the irregular patch with \( N = 7 \). For other irregular patches with \( N \neq 7 \) the dimension which is determined by \( P(12, N + 12) \) and the locations of 1s will be different, hence a programming subroutine is needed. The matrix equation for the expression for a surface is

\[
\mathbf{s}_k(\theta^1, \theta^2) = \bar{B}_k^T \cdot \bar{N}(\theta^1, \theta^2) \\
= (P_k \bar{X})^T \cdot \bar{N}(\theta^1, \theta^2) \\
= \bar{X}^T P_k^T \cdot \bar{N}(\theta^1, \theta^2) \quad (2.52)
\]

Eqn.2.52 can be substituted back into Eqn.2.55 and Eqn.2.56, so that both regular and irregular vertices could be evaluated using the twelve B-spline shape functions.

### 2.3 Validations for Kirchhoff-Love thin shell finite element method

In this section, a series of simulation results from the aforementioned subdivision finite element method are presented.
2.3.1 Finite element formula

The simplest energy functional for thin shell structures is of the following form due to Simo and Fox (1989)

\[ W(\alpha, \beta) = \frac{1}{2} \frac{Eh}{1 - v^2} H^{\alpha\beta\gamma\delta} \alpha_{\alpha\beta\gamma\delta} + \frac{1}{212(1 - v^2)} \frac{Eh^3}{12} H^{\alpha\beta\gamma\delta} \beta_{\alpha\beta\gamma\delta} \]  (2.53)

The specifications of terms in Eqn. 2.53, readers are referred to papers by Simo and Fox (1989); Cirak et al. (2000). After applying virtual energy principle, and using the displacement interpolation \( u(\theta^1, \theta^2) = \sum_{i=1}^{N^P} N^I(\theta^1, \theta^2) u_i \), the discretized equation is

\[ \mathbf{K}_h \mathbf{u}_h = \mathbf{f}_h \]  (2.54)

where \( \mathbf{K}_h \) is

\[ K^I_J = \sum_{k=1}^{NEL} \int_{\Omega_k} \left[ \frac{Eh}{1 - v^2} (M^I)^T H M^I + \frac{Eh^3}{12(1 - v^2)} (B^I)^T H B^I \right] d\Omega \equiv \sum_{k=1}^{NEL} K^I_J \]  (2.55)

and

\[ f^I_I = \sum_{k=1}^{NEL} \left[ \int_{\Omega_k} q N^I d\Omega + \int_{\Gamma_k \cap \Gamma} N N^I d\Gamma \right] \equiv \sum_{k=1}^{NEL} f^I_k \]  (2.56)

The left-hand side of Eqn. 2.54 represents the internal force in an elastic body and the right-hand side the external forces. The detailed expressions for each term can be found in the paper by Cirak et al. (2000). Eqn. 2.54 can be augmented to the form that describes the dynamic response of an elastic by
adding the inertia term as follows,

\[ M_h \ddot{u} + K_h \dot{u} = f_h \]  \hspace{1cm} (2.57)

where \( M_h \) is the element mass matrix, which is determined by

\[ M_h = \int_{\Omega_k} \rho N^1 N^1 d\Omega \]  \hspace{1cm} (2.58)

The calculation operates on element level then is distributed to the global level by usual assembly rules.

2.3.2 Elastic static responses

In this section, the subdivision finite element method is used to solve the elastic responses of a flat plate under the uniform pressure loading and a cylinder with two diametrically opposite concentrated forces, the configurations of which are shown in Fig. 2.16a and b respectively.

Both clamped and simply-supported boundary conditions are considered for the flat plate case by resort to the penalty method. Eqn. 2.54 is directly solved for the displacement by inverting the stiffness matrix \( K_h \). As shown in Fig. 2.16a, the length of the square plate is 100.0, the thickness 1.0. The Young's modulus and the Poisson ratio are \( 1.0 \times 10^7 \) and 0.0 respectively. A uniform pressure load of 1.0 is applied on the surface of the plate. The computed deformations of the plate for the clamped and the simply-support boundary con-
(a) The definition of the plate and the mesh used. (b) The definition of the cylinder and the mesh used.

Figure 2.16: Definitions for plate and cylinder.

(a) Clamped boundary condition. (b) Simply-supported boundary condition.

Figure 2.17: Simulation results for a flat square plate.
ditions are shown in Fig. 2.17.a and Fig. 2.17.b respectively. The maximum displacement is found at the center of the plate with clamped edges to be approximately 0.148 while the analytical solution being 0.151; and the maximum displacement for the simply-supported plate is about 0.487 with an error margin on the order of $o(10^{-4})$.

The cylinder under consideration has a height of 600.0, the cross-sectional diameter of 300.0, and the thickness 3.0. Young's modulus is $3.0 \times 10^6$, and Poisson ratio being 0.3. Both ends of the cylinder are totally fixed and two concentrated forces with the magnitude of 1.0 are positioned at two ends of the diameter line on the median cross-section plane which is 300 above the bottom of the cylinder. The correct response has been simulated in Fig. 2.18, where the com-

(a) Isometric view of the pinched cylinder. (b) Top view of the pinched cylinder.

**Figure 2.18**: Simulation results for a pinched cylinder.
puted displacement at the point of the concentrated force is about $1.93952 \times 10^{-5}$ comparing favorably to the analytical solution of $1.93488 \times 10^{-5}$.

2.3.3 Elastic dynamic responses

When the inertia effect becomes important, one has to consider adopting dynamic simulations. The governing equation is given by Eqn. 2.57. There are two types of methods for solving dynamic solid mechanics problems, that is the implicit and explicit formulations. The implicit method solves the problem by evaluating a nonlinear system of equations on the current time step, so that a complex and computationally expensive (massive core storage for stiffness matrix for large systems) algorithm has to be developed. While on the positive side, it does theoretically have no limit on how large a time step can be used so long as the physical correctness requirements are met, which is the major plus for this method. However many times, when highly nonlinear events such as contact, fracture and etc need to be simulated, the implicit method are outcast by its inherently complexity and stringent requirements on computer resources. Hence explicit dynamics method is popular in those scenarios because if the lumped mass matrix $M_h$ is used then no inversion of matrix for solving a linear equation system is needed, which greatly improves the easiness of programming and lowers computational cost for large problems. How-
ever the downside is that a very small time step needs to be used to maintain
stable integration process.

The algorithm of the explicit dynamic finite element method based on
Newmark central-difference scheme is shown in Listing 2.1. First the lumped
mass matrix stored in mass_lump(:) is computed by calling subroutine Massassem
on line100, which is of diagonal form so that its inversion is trivial. Then the sta-
bale time step is determined on line102 followed by the main Newmark integra-
tion loop starting from line104 to line122. In the loop, the displacement and ve-
locity fields are first calculated at the predictor step on line105 and line106 then
the boundary condition treatments adjust values on boundary nodes. The in-
ternal and external forces of the system are calculated in subroutine Getforce
on line111 and their sums are stored in resi_global(:), which is then used
in the corrector step on line112 to find the acceleration rate acc(:). The veloc-
ity vector can be updated on line116. Thus one Newmark integration iteration
completes. The transient results are written to files in the following lines at user
specified output frequency.

In the following, the explicit Newmark time integration scheme were
used to calculate several simple dynamic cases. The first case considered is a
square plate with four side clamped under impact loading. The side length is
0.254m, thickness 0.0127m. The material properties are as follows, the density
7.166 \times 10^4 N/m^3, the Young's modulus 0.6895 \times 10^{10} N/m^2, and the Poisson
ratio 0.3. The impact loading is applied uniformly over the entire plate surface
with the magnitude of 2.07 \times 10^6 N/m^2 at the beginning of simulation (t = 0)
and then stays on unchanged. The time variation of the deflection of the center
point of the square plate is depicted in Fig. 2.19 against the result by Sladek
et al. (2007) using the MLPG method. The deflection is normalized by the static
value 0.00819m and the time is scaled by 

\[ t_0 = \frac{t^2}{4\sqrt{\rho h/D}} = 1.35 \times 10^{-2} \text{ sec}, \]

where
Figure 2.19: Time variation of the central deflection of the clamped plate.

$L$ is the length of the plate, $\rho$ the material density, $h$ the plate thickness, and $D$ the plate flexural rigidity defined as $D = \frac{Eh^3}{12(1-\nu^2)}$ in which $E$ is the Young’s modulus and $\nu$ the Poisson ratio. The second case is a simply-supported square plate with the edge length of 2.438m, thickness of 0.00635m. Material properties are: density 2500$kg/m^3$, Young’s modulus $6.90 \times 10^{10} N/m^2$, Poisson ratio 0.25. The impact loading with the magnitude of $47.89 N/m^2$ is consider. The time variation of the central deflection is plotted in Fig. 2.20 which is compared with the results by Yang and Bhatti (1985).
Figure 2.20: Time variation of the central deflection of the simply-support plate.
Chapter 3
METHOD FOR CARBON NANOTUBE CALCULATIONS

The finite element method using Loop's subdivision scheme has been derived and validated for the cases of thin plate and cylindrical shell. The extension to the applications of carbon nanotubes requires constructing a constitutive relation of hyper-elastic potential type based on an appropriate atomic potential. To this end, it is necessary to devise a series of mathematical instruments that bridges the carbon lattice kinematics and macroscopic deformation measures. In this chapter, detailed account of this general continuum method will be given.

3.1 Linking atomic to macroscopic world

Tadmor et al. (1996) proposed a framework that established the relation between lattice deformations on atomic scale and nonlinear solid mechanics deformation measures on macroscopic scale. It's proved that this relation can only be through the deformation gradient $F$. Originally this method was designed for the study of defects in solids. But it's adopted here to provide as the
theoretical ground for the carbon nanotube calculations.

Figure 3.1: Schematics showing the linkage between atomic lattice and macroscopic finite element method.

The idea of this framework can be better seen from Fig. 3.1. For any macroscopic point on a continuum body, it is represented on atomic scale by a crystallite of radius $R_c$ with infinite number of atoms embedded. The crystalline lattice deforms according to the local continuum displacements. And the energy of the crystallite is determined from an appropriate atomic model. Since the energy is obtained directly from the atomic model, the key properties of the crystal, such as symmetry, are preserved. In finite element method, the continuum solid is discretized into many small elements, as shown in Fig.
3.1 b. The displacements in an element is interpolated from the values on the corresponding nodal points, and they become the only major unknowns to the numerical solution. For quasistatic problems studied in this thesis, the displacements are solved for directly by minimizing the total energy of the continuum solid, which is the sum of all contributions from each crystallite.

The bridge that links the crystallite deformation to the continuum one is the Cauchy-Born rule, for a good recent review see Ericksen (2008). Cauchy-Born rule is used to relate macroscopic deformation of crystals to changes in lattice vectors, $a = F A$, see Fig. 3.2, where $A$ and $a$ are the undeformed and deformed lattice vector respectively. As pointed out by Ericksen (2008), the lattice vectors underlying a continuum point deforms solely according to the local deformation gradient $F$ as long as the deformation is homogeneous and generally elastic. If an inhomogeneous event is sought, then the nonlocal for-
mulation needs to be used, which was discussed in the paper by Tadmor et al. (1999). But here in the current research, we consider all deformations in carbon nanotubes to be homogeneous and elastic only. For this reason, the choice of representative crystallite is immaterial to the calculation. We can always choose the one that is most convenient to the finite element formulation. For the honeycomb graphene sheet, the representative crystallite or the representative cell is chosen to be the one with the diamond shape shown at the center of Fig. 3.3, which has two carbon atoms (black dots) and three inequivalent bonds. The energy of this representative cell $E$ can be calculated from an appropriate atomic potential and the corresponding potential energy density $W$ is found by dividing $E$ with the area of the undeformed representative cell $S_0$. Because of the homogenization process assumed in our calculation, the mathematical formulae of potential energy density $W$ is applicable everywhere over the graphene sheet and can be used to compute energies contained in each finite element. Of
cause $W$ is a function of local deformation gradient $F$. Thus the correspondence between the macroscopic deformations and the atomic lattice deformations is established.

3.2 Revised Cauchy-Born rule for carbon monolayer

As discussed in the previous section, Cauchy-Born rule relates the atomic lattice vector to the macroscopic deformation through a simple relation, see Fig. 3.2. This is appropriate for bulk materials, such as the cubic lattice, however when it comes to atomic monolayers like the graphene sheet studied here, which in fact is a two-dimensional manifold embedded in a three-dimensional space, it can not be used directly. By referring to Fig. 3.4, the deformation gradient $F$ only maps the infinitesimal line segment between tangent spaces.

\[ a = F \cdot A \]

**Figure 3.4:** Direct use of Cauchy-Born rule leads to inaccurate results.
The lattice vector is in fact the chord of the carbon monolayer. To resolve this difficulty, Arroyo and Belytschko (2002, 2004) proposed to use the method of exponential map to approximate the true lattice vectors. Wang and Hu (2005) explored the applicability of using higher order Cauchy-Born rule, which was obtained by keeping an additional second order Taylor series expansion term of the deformation gradient $\mathbf{F}$. However the inclusion of a second order derivative of $\mathbf{F}$ could results in a much more complex mathematical formula in numerical implementation. Hence the exponential map is used in this thesis.

The effect of exponential map is to pull away a vector $\mathbf{v}$ in the tangent space to a chord $\mathbf{A}$ of the geodesic circle, as shown in Fig. 3.5. At point $P$ on the surface of the globe, there are a tangent vector $\mathbf{v}$ and the geodesic circle passing

![Figure 3.5: Schematics showing exponential map.](image-url)

69
through it. The exponential map $A = \exp_P(v)$ lays off a length equal to $||v||$ along the geodesic passing $P$ in the direction of $v$ to form a new vector $A$, that is the chord of the geodesic circle.

To relate to the current carbon monolayer problem, see, Fig. 3.6. On the left is the undeformed or the reference configuration where $A$ represents the bond vector connecting atom $X$ and $Y$ on the carbon sheet, $W$ or the tangent bond vector for convenience and on the right is the deformed or the current configuration where $a$ and $w$ are the counterparts of $A$ and $W$ in the reference

Figure 3.6: Schematics showing exponential map between two configurations.
configuration respectively. Since the deformation gradient \( F \) only maps entities in between the tangent spaces, that \( w = FW \), it is necessary to first use the inverse exponential map \( \exp_X^{-1} \) to map the chord vector \( A \) into the tangent space then after the deformation gradient operation use the exponential map \( \exp_X \) again to obtain the desired chord vector \( a \).

The above concept can be expressed in the following composite mapping relation

\[
a = \exp_{\phi(X)} \circ F(X) \circ \exp_X^{-1}(A) \tag{3.1}
\]

The order of composition operation is from the right most towards the left. Since carbon nanotubes are made, virtually by rolling up an originally flat graphene sheet into the cylindrical shape, it's reasonable and convenient to consider the flat carbon sheet as the reference configuration. By so doing, the exponential mapping Eqn. 3.1 can be simplified as

\[
a = \exp_{\phi(X)} \circ F(X) \circ (A) \tag{3.2}
\]

since the chord vector and the tangent vector are coincident on a flat sheet. The evaluation of exponential map needs the knowledge of geodesic circles which involved solving a system of two nonlinear ODE's. And the coefficients of the equations are Christoffel symbols. There isn't existing an analytic solution to it (do Carmo, 1976). Here an approximate method proposed by Arroyo and
Belytschko (2005) is adopted, which decouples the general deformation in two principal directions of curvature tensor, $V_1$ and $V_2$. Some local correction terms were added to obtain the approximate $a$. The method will be covered in the following sections.

3.2.1 Kinematics of graphene sheet

To analyze the deformation of carbon nanotubes, the nonlinear finite deformation formulae need to be used because:

1. a CNT is formed by rolling the flat carbon sheet into the cylinder configuration, which is a large deformation process;

2. the constitutive relation for CNT is derived from an atomic energy potential, which generally is nonlinear in its own right;

3. the instabilities of CNT is expected to occur during deformation.

Like the conventional finite deformation continuum mechanics, two configurations representing the undeformed (reference) state, $\Omega_0$ and the deformed (current) state $\Omega$, need to be identified, refer to Fig. 3.7 for the symbol uses and mapping relations. All quantities such as coordinates, surface basis vectors for the undeformed configuration are denoted by capitalized Latin letters, while those for the deformed configuration by lower case Latin letters. Both config-
Figure 3.7: Relations for the graphene sheet between undeformed and deformed configurations.

Conformations are referenced by a single Euclidean coordinate system \((i_1, i_2, i_3)\). A particle on \(\Omega_0\) can then be expressed as \(X = X^1i_1 + X^2i_2 + X^3i_3\), similarly for a particle on \(\Omega\), \(x = x^1i_1 + x^2i_2 + x^3i_3\). For the purpose of future finite element method applications, a reference configuration \(\bar{\Omega}\) is introduced. It is a two-dimensional slab located in the space spanned by \((\bar{I}_1, \bar{I}_2)\), a point on which is denoted by \(\bar{\xi} = \xi^1\bar{I}_1 + \xi^2\bar{I}_2\). The pair \((\xi^1, \xi^2)\) is actually the coordinates for the finite element base functions used for variable interpolations. Because of
the isoparametric representation used, every quantity regardless of whether it is on $\Omega_0$ or $\Omega$ can be expressed in terms of the parametric coordinates $(\xi^1, \xi^2)$ through base functions.

There are three types of mapping relations demonstrated in Fig. 3.7.

1. mapping entities between two tangent spaces of $\Omega_0$ and $\Omega$, i.e. $dx = FdX$ or $T\phi = T\phi_0 \circ T\phi_0^{-1}$;

2. mapping entities between $\Omega_0$ or $\Omega$ to the reference configuration $\tilde{\Omega}$, i.e.
   \[ X = \phi_0(\xi), \xi = \phi_0^{-1}(X) \text{ and } x = \phi(\tilde{\xi}), \tilde{\xi} = \phi_0^{-1}(x); \]

3. mapping entities between two configurations $\Omega_0$ and $\Omega$, i.e. $\phi(X) = \phi(\phi_0^{-1}(X))$.

The tangent spaces for configuration $\Omega_0$ and $\Omega$ are shown as the cross-hatched areas in Fig. 3.7, where any vector or the material line element belonging to these spaces of each configuration is related through the deformation gradient by $dx = FdX$. Each particle on either $\Omega_0$ or $\Omega$ are related to the parametric coordinates $(\xi^1, \xi^2)$ on $\tilde{\Omega}$ through the finite element interpolation scheme and is expressed as $X = \phi_0(\xi)$ and $x = \phi(\tilde{\xi})$. Their inverse mappings also prove to be useful, $\xi = \phi_0^{-1}(X)$ and $\tilde{\xi} = \phi_0^{-1}(x)$. In nonlinear finite element literature, they are sometimes called the pullback tensor, which will be used in future. By making use of the inverse mappings of type two, the direct mapping between
Ω₀ and Ω is \( \Phi(X) = \varphi(\varphi₀^{-1}(X)) \).

To describe the kinematics of carbon monolayer, besides the global Euclidean coordinate system, the convected coordinate system is also needed. Referring to Fig. 3.7, the basis vector \( \mathbf{G}_i \) spanning the tangent surface \( T_X\Omega₀ \) of \( \Omega₀ \) are

\[
\mathbf{G}_n = \frac{\partial \varphi₀^A}{\partial \xi^A} \mathbf{i}_A \tag{3.3}
\]

Components of Eqn. 3.3 are sometimes called the covariant components of the metric tensor and denoted as \( (T\varphi₀)^A_n \), since they can be used to calculate the element of area in the undeformed body

\[
d\Omega₀ = \det(T\varphi₀)dξ₁^1dξ₂^2 \tag{3.4}
\]

where \( \det \) represents the calculation of the determinant of a 2 × 2 matrix with the components described in Eqn. 3.3. The pullback tensor \([\mathbf{F}]_{CB₀}^2\) which is a two-point tensor and transforms physical quantities from the current configuration back to the reference configuration can also be derived from Eqn. 3.3 as

---

¹In the text that follows, the Greek letters, \( \alpha, \beta, \) etc., running from 1 to 2; the capitalized Latin letters, \( A, B, \) etc. running from 1 to 2; the lower case Latin letters, \( a, b, \) etc. running from 1 to 3. The Newton's summation convention applies to these indices. Other symbols do not have such implied summation rules.

²The calligraphic capitalized Latin letters, are used for denoting basis for different configurations, \( \mathcal{C} \) for the convected basis in \( \Omega; B₀ \) for the basis in \( \Omega₀; B \) for the basis in \( \Omega. \)
\[ [F]_{CB_0} = [T\varphi_0]^{-1}_{B_0B} \]  

(3.5)

In other words, the components of the *pullback* tensor are the components of the inverse matrix of the metric tensor \((T\varphi_0)^A_A\).

Similarly the basis vector \(g_a\) of the tangent plane \(T_x\Omega\) on the surface \(\Omega\) is given as

\[ g_a = \frac{\partial \varphi^a}{\partial \xi^a} i_a \]  

(3.6)

Note that the component of Eqn. 3.6 is on the basis of \(C\). The components of corresponding metric tensor are

\[ [g]_C = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix} \]  

(3.7)

where \(g_{\alpha\beta} = g_a^\alpha (g_\beta)^a\) is the inner product of \(g_a\) and \(g_\beta\).

The Green strain tensor is found through the definition \(C = F^T F\). It can be proved under convective coordinates, the components of Green strain tensor is \(C_j^i = G^{ik} g_{kj}\). The derivation can be found in appendix E. To express the Green strain tensor in the basis \(B_0\), by using the pullback tensor, Eqn. 3.5, we have

\[ C_{AB} = g_a\beta(T\varphi_0^{-1})^\alpha_A (T\varphi_0^{-1})^\beta_B \]  

(3.8)

The unit normal to the deformed configuration \(\Omega\) is

76
and the covariant components of the curvature tensor is

\[ k_{\alpha \beta} = n^a (g_{\alpha \beta})^a \]

(3.10)

where \( g_{\alpha \beta} = \frac{\partial g_\alpha}{\partial \xi^\beta} \). The quantities appeared in Eqn. 3.10 are all relative to the current configuration \( \Omega \), which is part of the solution, so it needs to be transformed to be relative to the reference configuration \( \Omega_0 \). To this end, the pullback tensor \( [T_{\Phi_0}^{-1}]_{\beta \alpha} \), determined in Eqn. 3.5 is use to pull back the quantities to the reference configuration. And the resulting pullback form of the curvature tensor is

\[ K_{AB} = k_{\alpha \beta} (T_{\Phi_0}^{-1})^\alpha_A (T_{\Phi_0}^{-1})^\beta_B \]

(3.11)

The principal curvature \( k_1, k_2 \) and the principal directions \( V_1, V_2 \) for the curvature tensor can be found by solving the following eigenvalue problem based on the reference configuration \( \Omega_0 \)

\[ K_{AB} V^B = kC_{AC} V^C \]

(3.12)

The solution to the above problem has been given by do Carmo (1976). The results are listed below for the completeness. The principal curvatures are

\[ k_{1,2} = H \pm \sqrt{H^2 - K}, \]

where \( K \), the Gaussian curvature is given as
and the mean curvature $H$ is given as

$$H = \frac{1}{2} \text{trace} \left( [C]^{-1} [K] [C]^{-1} \right) = \frac{\kappa_{11} \kappa_{22} - 2 \kappa_{12} \kappa_{12} + \kappa_{22} \kappa_{11}}{\kappa_{11} \kappa_{22} - \kappa_{12}^2} \quad (3.14)$$

By substituting the principal curvature $k_{1,2}$ back into Eqn. 3.12, the corresponding principal directions can be found as

$$V_{1,2} = \begin{bmatrix} -\kappa_{12} + C_{12} k_{1,2} \\ \kappa_{11} - C_{11} k_{1,2} \end{bmatrix} \quad (3.15)$$

then $V_{1,2}$ is normalized as

$$V_{1,2} = \frac{V_{1,2}}{\sqrt{V_{1,2}^T C V_{1,2}}} \quad (3.16)$$

### 3.2.2 Local approximation to exponential map for carbon sheet

As discussed at the beginning section of this chapter, the closed-form solution to the exponential map is not available and numerical methods involve solving a system of nonlinear ODE's with Christoffel symbols as coefficients, which makes the direct solution extremely difficult and inefficient for large scale problems. Arroyo and Belytschko (2002) proposed an approximation method to estimate the true exponential map of the bond vector. For con-
Since the reference configuration is a planar carbon sheet, the undeformed bond vector \( \mathbf{A} \) coincides with the tangent vector. The exponential map only needs to be performed once in the current configuration, see Fig. 3.6 for a quick reference. If we denote \( \mathbf{w} = F \circ \mathbf{A} \) as the tangent deformed bond vector, then the problem becomes to solving the equation \( \mathbf{a} = \exp_{\phi_X} \circ \mathbf{w} \). The method assumes that the curved surface can be represented locally in the principal direction \( \mathbf{V}_{1,2} \) by a cylinder with the radii to be \( 1/k_{1,2} \). The exponential map for a cylindrical surface has the closed-form solution, thus we could use this easy-to-calculate cylinder approximation to correct the tangent deformed bond vector, i.e. \( \mathbf{a} = \exp_{\phi_X} \circ \mathbf{w} \approx \mathbf{F} \circ \mathbf{A} + \triangle \mathbf{w}_1 + \triangle \mathbf{w}_2 \), where \( \triangle \mathbf{w}_{1,2} \) are the correction terms in the principal directions \( \mathbf{V}_{1,2} \). The final form of the deformed bond vector \( \mathbf{a} \) by approximate exponential map is

\[
\mathbf{a} = \begin{bmatrix} a^1 \\ a^2 \\ a^3 \end{bmatrix} = \begin{bmatrix} \frac{w^1 \sin(k_1 w^1)}{k_1 w^1} \\ \frac{w^2 \sin(k_2 w^2)}{k_2 w^2} \\ \frac{k_1 (w^1)^2 \sin^2(k_1 w^1)}{2} + \frac{k_2 (w^2)^2 \sin^2(k_2 w^2)}{2} \end{bmatrix}
\]

(3.17)

the terms \( w^{1,2} \) in Eqn. 3.17 are the components of the tangent deformed bond vector \( \mathbf{w} \) resolved in the two principal directions \( \mathbf{V}_{1,2} \) and they are given as.
\[
\mathbf{w} = \begin{bmatrix} w^1 \\ w^2 \end{bmatrix} = \begin{bmatrix} C_{AB}^{A}(V_1)^B \\ C_{AB}^{A}(V_2)^B \end{bmatrix}
\]

The length of the deformed bond vector is simply

\[ a = \|a\| \]

and the angle between any two bond vectors \(a\) and \(b\) can be found by

\[ \cos(\theta) = \frac{a \cdot b}{\|a\|\|b\|} \]

When applying the exponential Cauchy-Born rule to carbon sheets, there is another issue that deserves attention. By reference to Fig. 3.8, each lattice point in the crystal structure of carbon sheet is actually associated with two distinguishable material units (sublattice). Sublattice \(A, L_A\), represented by red dots interpenetrates another sublattice \(B, L_B\), represented by dark blue dots. If sublattice \(A\) is taken as the center lattice, then any red dots can be a center nuclear site. From the relative motion point of view, the deformation of the graphene sheet can be thought of as sublattice \(B\) (dark blue) moving away from sublattice \(A\) to the light blue dots. The Bravais vectors (red arrow-headed lines) \(B_1\) and \(B_2\) that connect two neighboring red dots serve as the basis vectors. Because it is a multilattice structure, use of Bravais vectors alone can not uniquely describe every atomic site. Hence the shift vector \(P\) connecting neighboring sublattice nuclear sites (red and blue) is needed. As pointed out by
Cousins (1978), every site on a given sublattice has the same symmetry but different sublattices may have different symmetries. And as long as there is one sublattice not possessing the inversion symmetry, which is the case for carbon sheets, the occurrence of relative displacement (represented by $\eta$ in Fig. 3.8) between two sublattices under a homogeneous deformation is inevitable. The relative displacement represents a rearrangement phenomenon on lattice level.
that leads to the internal energy equilibrium. Hence it is also called the inner
displacement.

Based on the above analysis of carbon lattice, the undeformed bond vec-
tor $A$ in the planar configuration can be given as,

\[ A_1 = P - B_2 + \eta = A_{01} + \eta \]  (3.21)
\[ A_2 = P - B_2 + \eta = A_{02} + \eta \]  (3.22)
\[ A_3 = P + \eta = A_{03} + \eta \]  (3.23)

where $A_{0i}$ is the original carbon bond vectors based on the hexagonal lattice
structure; while $A_i$ can be considered as the relaxed bond vectors because they
are modified by the inner displacement $\eta$ which minimizes the strain energy
density of the structure on the lattice level. Note that the formula for calculat-
ing deformed bond vector $a$, Eqn. 3.17 still holds, and $A_i$ should be substituted
instead. Fig. 3.9 shows the representative cell and bond structure for the un-
deformed carbon sheet before the inner displacement $\eta$ is determined. On the
right are three inequivalent carbon bonds for the representative cell and they
are separated apart by 120°. Angle $\theta_0$ formed by bond vector $A_{01}$ and the $x$ axis
or the rolling axis can be considered as the chiral angle. It's straightforward to
show that the three bond vectors $A_{0i}$ are of the following forms

82
Figure 3.9: Undeformed carbon bonds.

\[ A_{01} = \begin{bmatrix} A_0 \cos \theta_0 \\ A_0 \sin \theta_0 \end{bmatrix}, \quad A_{02} = \begin{bmatrix} A_0 \cos \left( \theta_0 + \frac{2\pi}{3} \right) \\ A_0 \sin \left( \theta_0 + \frac{2\pi}{3} \right) \end{bmatrix}, \quad A_{03} = \begin{bmatrix} A_0 \cos \left( \theta_0 - \frac{2\pi}{3} \right) \\ A_0 \sin \left( \theta_0 - \frac{2\pi}{3} \right) \end{bmatrix} \]

(3.24)

which are based on the two dimensional undeformed configuration. From the above formulae we can see that the inner displacement only affects the undeformed body and represent an in-plane adjustment for the solid. The method of calculating inner displacements will be discussed later.

So far the formulae for carbon bond lengths \(a_i(C, K, A_i)\) and bond angles \(\theta_i(C, K, A_j, A_k)\) have been derived, Eqn. 3.19, and 3.20. It is found that through the use of the general continuum method described here, the crystal lattice quantities, namely the bond length and bond angle can be expressed as functions of the strain measures for macroscopic continuum mechanics, such as the Green strain tensor \(C\) and the curvature tensor \(K\). If an appropriate interatomic po-
tential that accepts the input of bond length and angle was selected, then the atomic mechanics problems could be solved in the conventional macroscopic framework, such as the finite element method.

3.3 Interatomic potentials

In the field of molecular dynamics (MD), different types of interatomic potentials for various molecules have been proposed, validated and applied. The accuracy of a potential is largely determined by its ability to simulate the multibody interactions and quantum effects within a molecule system. Generally speaking the more multibody terms are included in the formula the higher accuracy will it have. However registering and tracking so many atoms for multibody interactions is computationally expensive and inefficient. As a result, all the multibody effects of the order higher than three are usually ignored. Even this it is still a haunting task for computer modelling using a three-body potential. The bond-order type of interatomic potentials have been proposed, notably by Tersoff (1988); Brenner (1990). The higher than two multibody terms are not used but the bond order terms which implicitly describe the angular dependence of interatomic forces are added so that the over-all pair-wise formulation is preserved. For carbon nanotube modelling the reactive empirical bond order (REBO) potential developed by Brenner (1990) has gained a wide
popularity. The REBO potential is an empirical many-body potential-energy expression developed for hydrocarbons that can model intermolecular chemical bonding in a variety of small hydrocarbon molecules as well as graphite and diamond lattices. REBO potential counts the short-ranged interactions. The van der Walls long-ranged interatomic force in carbon nanotube simulations plays an important role and is described by the Lennard-Jones potential. In this section the revised continuum version of both potentials will be presented.

3.3.1 REBO potential

The major assumption that we laid for this study is that only the homogeneous elastic deformations of CNTs is considered. And by the spirit of the general continuum method, the total interatomic energy can be found by evaluating it over a representative cell, the location of which is of no preference over the carbon sheet. By reference to Fig. 3.9 the three inequivalent bonds are picked up and redrawn here in Fig. 3.10 for a better explanation. The interatomic energy for this representative cell depends on the variations of the bond lengths and angles. The labelling rules are demonstrated in Fig. 3.10, where $a_1$ is the length of the bond vector formed by atom 1 and 2; $a_2$ and $a_3$ following the similar fashion. The angle formed by bond vectors $a_2$ and $a_3$ is denoted by $\theta_1$; the namings of $\theta_2$ and $\theta_3$ are also shown.
With the naming rules handy, the specialized form of the REBO potential function for carbon nanotube modelling is as follows

\[
E = \sum_{i=1}^{3} [V_R(a_i) - B_i(a_j, a_k, \theta_j, \theta_k)V_A(a_i)]
\]

(3.25)

The energy \( E \) contained in this representative cell is the sum by looping over three bonds and three angles as shown in Eqn. 3.25 where \( V_R(a_i) \) is the repulsive pair term and is given as
\[ V_R(a_i) = f_i(a_i) \frac{D_{CC}^{(e)}}{S_{CC} - 1} e^{-\sqrt{2s_{CC}}\beta_{CC}(a_i - R_{CC}^{(2)})} \]  \hspace{1cm} (3.26) \\

and \( V_A(a_i) \), the attractive pair term and is given as

\[ V_A(a_i) = f_i(a_i) \frac{D_{CC}^{(e)}}{S_{CC} - 1} e^{-\sqrt{2s_{CC}}\beta_{CC}(a_i - R_{CC}^{(2)})} \]  \hspace{1cm} (3.27) \\

and bond-order term \( B_i \) is

\[ B_i = \left[ 1 + G_C(\theta_i) f_j(a_j) + G_C(\theta_j) f_k(a_k) \right]^{-\delta} \]  \hspace{1cm} (3.28)

the bond angle function \( G_C(\theta_i) \) is

\[ G_C(\theta_i) = a_0 \left\{ 1 + \frac{c_0^2}{d_0^2} - \frac{c_0^2}{d_0^2 + (1 + \cos \theta_i)^2} \right\} \]  \hspace{1cm} (3.29)

the switching function that maintains a smooth transition near the cutoff range is given as

\[ f_i(a_i) = \begin{cases} 1 & \text{, } a_i < R_{CC}^{(1)} \\ \frac{1}{2} \left[ 1 + \cos \left( \frac{\pi(a_i - R_{CC}^{(1)})}{R_{CC}^{(2)} - R_{CC}^{(1)}} \right) \right] & \text{, } R_{CC}^{(1)} < a_i < R_{CC}^{(2)} \\ 0 & \text{, } a_i > R_{CC}^{(2)} \end{cases} \]  \hspace{1cm} (3.30)

The second set of parameters provided by Brenner (1990) is adopted and listed in Table 3.1. These parameters corresponds to the equilibrium bond length of \( \| A_0 \| = 1.45 \text{\AA} \). To evaluate the total energy of the carbon sheet, again we need to follow the assumption of homogeneity of the deformation. The in-
Table 3.1: Parameters for REBO potential

<table>
<thead>
<tr>
<th>parameter</th>
<th>value</th>
<th>parameter</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_{CC}^{(e)}$</td>
<td>1.39 Å</td>
<td>$D_{CC}^{(e)}$</td>
<td>6.0 eV</td>
</tr>
<tr>
<td>$\beta_{CC}$</td>
<td>2.1 Å$^{-1}$</td>
<td>$S_{CC}$</td>
<td>1.22</td>
</tr>
<tr>
<td>$\delta_{CC}$</td>
<td>0.5</td>
<td>$a_{CCC}$</td>
<td>0.0</td>
</tr>
<tr>
<td>$R_{CC}^{(1)}$</td>
<td>1.7 Å</td>
<td>$R_{CC}^{(2)}$</td>
<td>2.0 Å</td>
</tr>
<tr>
<td>$a_0$</td>
<td>0.000 208 13</td>
<td>$c_0^2$</td>
<td>330$^2$</td>
</tr>
<tr>
<td>$d_0^2$</td>
<td>3.5$^2$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.11: Apply REBO potential to representative cell.

teratomic energy density $W$ can be computed using the result from Eqn. 3.25 as follows

$$W = \frac{E}{S_0}$$  \hspace{1cm} (3.31)

where $S_0$ is the area of the representative cell as shown in Fig. 3.11 and is given as

$$S_0 = \frac{3\sqrt{3}}{2} \| A_{0f} \|^2$$  \hspace{1cm} (3.32)
Of course $W$ is the function of $C$, $K$, $A_i$ and the inner displacement $\eta$ and all of these quantities have been converted to be relative to the reference configuration $\Omega_0$ through the pullback operation. Hence the total energy can be found by integrating the energy density over the surface of $\Omega_0$ as follows

$$E_{\text{REBO}} = \int_{\Omega_0} W d\Omega_0$$  \hspace{1cm} (3.33)

### 3.3.2 Lennard-Jones potential

The van der Waals force is only active between two atoms that are not bonded and within the cutoff radius. If two representative cells are considered,

![Figure 3.12: Schematics showing Lennard Jones potential calculation over the representative cells.](image)

89
labeled as $I$ and $J$, as shown in Fig. 3.12. There are two carbon atoms in each cell, then the Lennard Jones potential energy can be calculated as follows

$$E_{LJ} = \sum_{I=1}^{4} \sum_{J=1, J \neq I}^{4} V(||r_{IJ}||)$$  \hspace{1cm} (3.34)

where $r_{IJ}$ is the vector that connects two nonbonded atoms located within different representative cells; $B_I$ is a group of atoms that are bonded to atom $I$. The computation is looped over two nonbonded atoms. However Eqn. 3.34 can only be carried out over discrete individual atoms. For the purpose of finding the total Lennard-Jones energy about the surface of carbon nanotubes, a continuum version of Eqn. 3.34 is needed. The argument about homogeneity is invoked again, that is, the carbon atoms are distributed homogeneously on the CNT surface and undergoes a homogeneous deformation. A notion of mean surface density for carbon atoms, $M$ could be introduced as $M = 2/S_0$. Hence the practical Lennard Jone potential formula is

$$E_{LJ} = \frac{4}{5} \int_{\Omega_0} \int_{\Omega_0 - B_I} V(||r_{IJ}(X_I, X_J)||) d\Omega_0 d\Omega_0$$  \hspace{1cm} (3.35)

where $X_I$ and $X_J$ represents the coordinates for an particle (not necessarily coincident with carbon atoms) on the continuum surface defined by the representative cell $I$ or $J$. As a result the total van der Waals potential energy between two interacting surfaces can be found through a double integration.
The exact van der Waals potential expression takes the following format:

\[ V = \begin{cases} 
0.0, & r_{ij} \leq r_{\text{small}} \\
 c_{3,k}(r_{ij} - r_k)^3 + c_{2,k}(r_{ij} - r_k)^2 + c_{1,k}(r_{ij} - r_k) + c_{0,k}, & r_{\text{small}} < r_{ij} < r_{\text{medium}} \\
 4\epsilon \left[ \left( \frac{\sigma}{r_{ij}} \right)^{12} - \left( \frac{\sigma}{r_{ij}} \right)^6 \right], & r_{\text{medium}} \leq r_{ij} \leq r_{\text{big}} 
\end{cases} \]

(3.36)

The parameters for Lennard-Jones potential are taken from Sinnott et al. (1998) and listed in Table 3.2. Parameters \( c_{n,k} \) and \( r_k \) are the coefficients of the cubic spline interpolation function, which can be found from the original paper of Sinnott et al. (1998). Eqn. 3.36 includes a third order polynomial as the switching function so that both the function and the gradient evaluations of the Lennard-Jones potential are smoother and reduce overshooting when nearing the lower bond of the cutoff radius. The curves showing how modified Lennard-Jones potential and its gradient vary with the changes in the distance is provided in Fig. 3.13.
Interaction distance (AA)

(a) Lennard-Jones potential.

(b) Gradient of Lennard-Jones potential.

Figure 3.13: Modified Lennard-Jones potential.
Chapter 4
NUMERICAL IMPLEMENTATION OF GENERAL CONTINUUM METHOD

4.1 Setup of the subdivision finite element method

Because of the similarity with the thin shell problems discussed in Chapter Two (curvatures are used as one of the strain measures), the equations derived for carbon nanotubes put a stringent continuity requirement on the possible finite element approximating scheme. As have been thoroughly explored in Chapter Two, the high-order subdivision finite element method can provide at least $C^1$ continuity in parametrizing surfaces. As a result it is chosen to be the numerical method for CNT modelling.

A nonlinear large deformation finite element formula for CNTs will be constructed. Two different configurations are established as shown in Fig. 4.1. The undeformed configuration is a two dimensional rectangular sheet defined in the Euclidean coordinate system $(X^1, X^2)$ and the deformed one is a cylindrical surface formed by rolling the undeformed rectangular sheet along the block arrow direction in the $(x^1, x^2, x^3)$ space. This initial deformation map can be
Figure 4.1: Mapping relations for finite element approximation.

established as follows

\[
\begin{align*}
\Phi^1 &= X^1 \\
\Phi^2 &= R_0 \cos \left( \frac{X^2}{R_0} \right) \\
\Phi^3 &= R_0 \sin \left( \frac{X^2}{R_0} \right)
\end{align*}
\]

(4.1)

where \( R_0 \) is the radius of carbon nanotube. This arrangement bears physical senses. The two dimensional planar carbon sheet represents the lowest carbon atomic energy state; and having a planar undeformed configuration reduces
some difficulties in numerical implementations. The two configurations are related through the triangular parametric domain \((\xi^1, \xi^2)\), see Fig. 4.1 for the demonstration, which is further related to the finite element base functions, see Appendix D. Both meshes on the two configurations are designed to have the same regular pattern, that is, every inner node has exactly six valence edges and the same indexing scheme. And for the rectangular undeformed mesh, a series of ghost elements are built around the original boundary edges (highlighted in Fig. 4.2) which makes every boundary node including the corner nodes have enough neighboring nodes to be regular. All of these make the mappings among different configurations much smoother. Since all elements are regular, the evaluation of the coordinates of a point within an element can be done by using the template shown in Fig. 4.3. The center element is in dark color where there are three sampling points (or the Gauss quadrature points when numerical integration is used). These sampling points are represented by the
parametric coordinates (in fact they are triangular area coordinates for the base functions used here) \((\zeta^1, \zeta^2)\). And the order of the indices for the nodes in the one neighbor of center element is also shown. The coordinates of the sampling points for an element in the undeformed and deformed configurations can be expressed as follows\(^1\)

\[
(q^e_0)^A(\zeta^1, \zeta^2) = \sum_{I=1}^{12} (q^e_0)^A_I N_I(\zeta^1, \zeta^2)
\]

\[\text{(4.2)}\]

\[
(q^e)^A(\zeta^1, \zeta^2) = \sum_{I=1}^{12} (q^e)^A_I N_I(\zeta^1, \zeta^2)
\]

\[\text{(4.3)}\]

Once Eqn. 4.2 and 4.3 are assembled for the entire mesh, the deformed config-

---

\(^1\)The meanings of sub-/super-scripts are as follows: superscript \(e\) signifies that the quantity is evaluated over an element; superscript \(A\) represents the \(A\)th Cartesian component of the quantity; subscript \(0\) means the quantity is about the undeformed configuration \(Q_0\); subscript \(I\) represents \(I\)th finite element base function \(N_I\) or \(I\)th node according to the indexing scheme described in Fig. 4.3.
uration \( \varphi \) can be found by solving for the nodal values of \( (\varphi^e)^f_i \), i.e. the nodal degrees of freedom for the discretized finite element problem.

4.2 Numerical representations of strain measures

In this section, the strain measure terms for carbon nanotube derived in Section 3.2 will be discretized using subdivision finite element base functions. And note that every numerical formula needs to be evaluated at every quadrature point of each element.

**Pullback tensor** \( (T \varphi^e)^A_0 \)

In Eqn. 3.3, we show that the metric tensor for the undeformed configuration \( \Omega_0 \) to be

\[
(T \varphi_0)^A = \frac{\partial \varphi^A_0}{\partial \zeta^A}
\]

if discretized over element \( e \), it becomes

\[
(T \varphi^e)^A = \begin{bmatrix}
\sum_{i=1}^{12}(\varphi^e_0)^1_i N_{i,1} & \sum_{i=1}^{12}(\varphi^e_0)^1_i N_{i,2} \\
\sum_{i=1}^{12}(\varphi^e_0)^2_i N_{i,1} & \sum_{i=1}^{12}(\varphi^e_0)^2_i N_{i,2}
\end{bmatrix}
\]

(4.4)

where \( N_{i,k} \) represents \( \partial N_i(\zeta^A) / \partial \zeta^A \). The pullback tensor \( P^e \) is the inverse of Eqn. 4.4.
\[(P^e)_{\alpha A} = \left[(T \varphi_0^e)^{A}_{\alpha}\right]^{-1}\]  

(4.5)

Since the undeformed configuration \(\Omega_0\) is known \textit{a priori} from the problem setting and keeps unchanged, the pullback tensor needs only to be calculated once at the start of the calculation and stored in an array for later use.

**Metric tensor \(g_{\alpha\beta}^e\) for deformed configuration \(\Omega\)**

The basis vector \(g_\alpha\) for \(\Omega\) is given in Eqn. 3.6 as

\[g_\alpha = \frac{\partial \varphi_i}{\partial \xi^\alpha} i_\alpha\]

The discretized form for element \(e\) is

\[
\begin{align*}
    g_{1}^e &= \sum_{i=1}^{12} (\varphi_i^e)^1_i N_{i,1} i_1 + \sum_{i=1}^{12} (\varphi_i^e)^2_i N_{i,2} i_1 + \sum_{i=1}^{12} (\varphi_i^e)^3_i N_{i,3} i_1 \\
    g_{2}^e &= \sum_{i=1}^{12} (\varphi_i^e)^1_i N_{i,2} i_1 + \sum_{i=1}^{12} (\varphi_i^e)^2_i N_{i,1} i_2 + \sum_{i=1}^{12} (\varphi_i^e)^3_i N_{i,3} i_2 \\
\end{align*}
\]

(4.6)

The components of the metric tensor \(g_{\alpha\beta}^e\) is given as

\[g_{\alpha\beta}^e = g_{\alpha}^e \cdot g_{\beta}^e\]

(4.7)

Eqn. 4.7 represents an inner product between two basis vectors and \(\alpha\) and \(\beta\) varies between 1 and 2.

98
The Green strain tensor is obtained by performing the pullback operation on the metric tensor $g_{a\beta}$ and is given in Eqn. 3.8 as

$$C_{AB} = g_{a\beta}(T\varphi^{-1}_0)^a_A(T\varphi^{-1}_0)^\beta_B$$

By use of Eqn. 4.5 and 4.7 over element $e$, we have the discretized form

$$C_{AB}^e = g_{11}^e (P^e)^1_A (P^e)^1_B + g_{12}^e (P^e)^1_A (P^e)^2_B$$

$$+ g_{21}^e (P^e)^2_A (P^e)^1_B + g_{22}^e (P^e)^2_A (P^e)^2_B$$

(4.8)

The covariant components of the curvature tensor is given in Eqn. 3.10

$$k_{a\beta} = n^a (g_{a\beta})^a$$

where $n^a$ is the component of the unit normal of surface $\Omega$ given in Eqn. 3.9 and $g_{a\beta}$ is the component of the derivative of the basis vector $g_a$.

$$g_{a,\beta}^e = \sum_{i=1}^{12} (\varphi^e)^1_i N_{i,\alpha}^a i_1 + \sum_{i=1}^{12} (\varphi^e)^2_i N_{i,\alpha}^a i_2 + \sum_{i=1}^{12} (\varphi^e)^3_i N_{i,\alpha}^a i_3$$

(4.9)

where $N_{i,\alpha}$ represents $\partial N_i/\partial x^\alpha \partial \xi^\beta$. The pullback of the curvature tensor $k_{a\beta}$ is
\[ K'_{AB} = k'_{11}(P^e)_A^1(P^e)_B^1 + k'_{12}(P^e)_A^1(P^e)_B^2 + k'_{21}(P^e)_A^2(P^e)_B^1 + k'_{22}(P^e)_A^2(P^e)_B^2 \] (4.10)

### 4.3 Energy computations

The contributing sources to the total energy of the CNT are: 1. the short-range interatomic energy described by REBO potential, \( E_{\text{REBO}} \); 2. the long-range van der Waals energy described by Lennard-Jones potential, \( E_{\text{LJ}} \); 3. the external energy, \( E_{\text{external}} \). In this section, the numerical method to calculate those energies are presented.

#### 4.3.1 REBO energy calculation

The CNT surface is subdivided into many small triangular elements in numerical simulations and the total REBO energy can be evaluated by adding up the contributions from each individual element. Starting from Eqn. 3.33, we have
\[ E_{\text{REBO}} = \int_{\Omega_0} W(C, K, \eta) d\Omega_0 \]  
\[ = \sum_{e=1}^{nele} \int_{\Omega_e} W(C, K, \eta) \det(\begin{pmatrix} T \phi_i^e \end{pmatrix}) d\Omega \]  
\[ \approx \sum_{e=1}^{nele} \sum_{i=1}^{nqp} W(C, K, \eta) \xi_i \det \begin{pmatrix} (T \phi_i^e) \end{pmatrix} \omega_i \]

where \( nele \) is the number elements in the mesh; \( nqp \) is the number quadrature points used and \( \omega_i \) are the corresponding weights for the numerical integration. The integration over the entire undeformed domain \( \Omega_0 \) is first broken down into being over individual finite element \( e \), then by using the Jacobian it is further converted to be on the triangular parametric domain \( \tilde{\Omega} \). At the final step, the numerical quadrature is carried out for the integration, where the integrand is evaluated at \( nqp \) quadrature points \( \xi_i \), multiplied by the weight \( \omega_i \).

In this thesis, three quadrature points were used for numerical integrations.

The derivative of \( E_{\text{REBO}} \) with respect to the degrees of freedom \( \varphi_I \) is the internal force for the discrete system. And it’s of key importance in the subsequent calculations. The detailed method of evaluating derivatives will be presented here.
\[
(F_{\text{internal}})_i = \frac{\partial E_{\text{REBO}}}{\partial \varphi_i} 
\]
\[
= \sum_{e=1}^{n_{\text{ele}} \cdot n_{\text{nap}}} \sum_{i=1}^{n_{\text{nap}}} \frac{\partial W(C, K, \eta)}{\partial \varphi_i} |_{\varphi_i} \det [(T\varphi_0^\text{e})]_{\varphi_i} \omega_i 
\]
\[
= \sum_{e=1}^{n_{\text{ele}} \cdot n_{\text{nap}}} \sum_{i=1}^{n_{\text{nap}}} \left[ \frac{\partial W}{\partial C} \frac{\partial C}{\partial \varphi_i} + \frac{\partial W}{\partial K} \frac{\partial K}{\partial \varphi_i} + \frac{\partial W}{\partial \eta} \frac{\partial \eta}{\partial \varphi_i} \right] |_{\varphi_i} \det [(T\varphi_0^\text{e})]_{\varphi_i} \omega_i 
\]

where the derivative of the energy density \( W \) with respect to the inner displacement \( \eta \), \( \partial W / \partial \eta \) could be removed by performing the inner relaxation as discussed in Section 3.2.2. The numerical method for this operation will be discussed later. If the inner relaxation has been achieved then the formula for the internal force calculation is

\[
(F_{\text{internal}})_i = \sum_{e=1}^{n_{\text{ele}} \cdot n_{\text{nap}}} \sum_{i=1}^{n_{\text{nap}}} \left[ \frac{\partial W}{\partial C} \frac{\partial C}{\partial \varphi_i} + \frac{\partial W}{\partial K} \frac{\partial K}{\partial \varphi_i} \right] \det [(T\varphi_0^\text{e})]_{\varphi_i} \omega_i 
\]

Upon examining the expression for REBO potential energy, Eqn. 3.25, we found that the energy density \( W = E/S_0 \) is explicitly the function of deformed bond lengths \( a_i \) and bond angles \( \theta_i \); while \( a_i \) and \( \theta_i \) are the functions of the Green strain tensor \( C \) and the curvature tensor \( K \), see Section 3.2 for details. Hence by chain rule, the derivative terms in Eqn. 4.13 can be calculated as follows,
Derivatives of REBO energy density with respect to the Green strain tensor and the curvature tensor

It can be shown that

$$
\frac{\partial W}{\partial C} = \sum_{i=1}^{3} \left( \frac{\partial W a_i}{\partial a_i \ C} + \frac{\partial W \theta_i}{\partial \theta_i \ C} \right)
$$

(4.14)

and

$$
\frac{\partial W}{\partial K} = \sum_{i=1}^{3} \left( \frac{\partial W a_i}{\partial a_i \ K} + \frac{\partial W \theta_i}{\partial \theta_i \ K} \right)
$$

(4.15)

In the above equations, repetitive indices don't imply a summation and \((i,j,k)\) is an even permutation of the triplet \((1,2,3)\). The related derivatives can be found as

$$
\frac{\partial W}{\partial a_i} = \frac{1}{S_0} \left[ \frac{dV_R(a_i)}{d a_i} - B_i \frac{dV_A(a_i)}{d a_i} - V_A(a_i) \frac{\partial B_i}{\partial a_i} - V_A(a_k) \frac{\partial B_k}{\partial a_i} \right]
$$

and

$$
\frac{\partial W}{\partial \theta_i} = \frac{1}{S_0} \left[ -V_A(a_i) \frac{\partial B_i}{\partial \theta_i} - V_A(a_k) \frac{\partial B_k}{\partial \theta_i} \right]
$$

In the above equations, repetitive indices don't imply a summation and \((i,j,k)\) is an even permutation of the triplet \((1,2,3)\). The related derivatives can be found as

$$
\frac{dV_R(a_i)}{d a_i} = \frac{D^{(e)}_{CC}}{S_{CC} - 1} e^{-\sqrt{2S_{CC}\beta_{CC}(a_i - R^{(e)}_{CC})}} \frac{df_i(a_i)}{d a_i} - \frac{D^{(e)}_{CC}}{S_{CC} - 1} \sqrt{2S_{CC}\beta_{CC}f_i(a_i)} e^{-\sqrt{2S_{CC}\beta_{CC}(a_i - R^{(e)}_{CC})}}
$$
\[
\frac{dV_A(a_i)}{da_i} = \frac{D_{CC}^{(e)} S_{CC}}{S_{CC} - 1} e^{-\sqrt{2/\lambda_{CC}} \beta_{CC}(a_i - \lambda_{CC})} \frac{df_1(a_i)}{da_i} - \frac{D_{CC}^{(e)} S_{CC}}{S_{CC} - 1} \sqrt{2/\lambda_{CC}} \beta_{CC} f_1(a_i) e^{-\sqrt{2/\lambda_{CC}} \beta_{CC}(a_i - \lambda_{CC})}
\]

The derivative of bond order term \( B_1 \) with respect to the bond lengths and angles are

\[
\frac{\partial B_1}{\partial a_1} = 0
\]

\[
\frac{\partial B_1}{\partial a_2} = -\delta [1 + G_C(\theta_3) f_2(a_2) + G_C(\theta_2) f_3(a_3)]^{-\delta - 1} G_C(\theta_3) \frac{df_2(a_2)}{da_2}
\]

\[
\frac{\partial B_1}{\partial a_3} = -\delta [1 + G_C(\theta_3) f_2(a_2) + G_C(\theta_2) f_3(a_3)]^{-\delta - 1} G_C(\theta_2) \frac{df_3(a_3)}{da_3}
\]

\[
\frac{\partial B_1}{\partial \theta_1} = 0
\]

\[
\frac{\partial B_1}{\partial \theta_2} = -\delta [1 + G_C(\theta_3) f_2(a_2) + G_C(\theta_2) f_3(a_3)]^{-\delta - 1} f_3(a_3) \frac{dG_C(\theta_2)}{d\theta_2}
\]

\[
\frac{\partial B_1}{\partial \theta_3} = -\delta [1 + G_C(\theta_3) f_2(a_2) + G_C(\theta_2) f_3(a_3)]^{-\delta - 1} f_2(a_2) \frac{dG_C(\theta_3)}{d\theta_3}
\]

The derivative of bond order term \( B_2 \) with respect to the bond lengths and
angles are

\[ \frac{\partial B_2}{\partial a_1} = -\delta [1 + G_C(\theta_3)f_1(a_1) + G_C(\theta_1)f_3(a_3)]^{-\delta-1} G_C(\theta_3) \frac{df_1(a_1)}{da_1} \]

\[ \frac{\partial B_2}{\partial a_2} = 0 \]

\[ \frac{\partial B_2}{\partial a_3} = -\delta [1 + G_C(\theta_3)f_1(a_1) + G_C(\theta_1)f_3(a_3)]^{-\delta-1} G_C(\theta_1) \frac{df_3(a_3)}{da_3} \]

\[ \frac{\partial B_2}{\partial \theta_1} = -\delta [1 + G_C(\theta_3)f_1(a_1) + G_C(\theta_1)f_3(a_3)]^{-\delta-1} f_3(a_3) \frac{dG_C(\theta_1)}{d\theta_1} \]

\[ \frac{\partial B_2}{\partial \theta_2} = 0 \]

\[ \frac{\partial B_2}{\partial \theta_3} = -\delta [1 + G_C(\theta_3)f_1(a_1) + G_C(\theta_1)f_3(a_3)]^{-\delta-1} f_1(a_1) \frac{dG_C(\theta_3)}{d\theta_3} \]

The derivative of bond order term \( B_3 \) with respect to the bond lengths and angles are

\[ \frac{\partial B_3}{\partial a_1} = -\delta [1 + G_C(\theta_2)f_1(a_1) + G_C(\theta_1)f_2(a_2)]^{-\delta-1} G_C(\theta_2) \frac{df_1(a_1)}{da_1} \]

\[ \frac{\partial B_3}{\partial a_2} = -\delta [1 + G_C(\theta_2)f_1(a_1) + G_C(\theta_1)f_2(a_2)]^{-\delta-1} G_C(\theta_1) \frac{df_2(a_2)}{da_2} \]

\[ \frac{\partial B_3}{\partial a_3} = 0 \]
The derivative of bond angle function with respect to $\theta_i$ is

$$
\frac{dG_c(\theta_i)}{d\theta_i} = \frac{-2a_0 c^2 \sin(\theta_i)(1 + \cos(\theta_i))}{[d^2 + (1 + \cos \theta_i)^2]^2}
$$

and the derivative of the switching function with respect to $a_i$ is

$$
\frac{df_i(a_i)}{da_i} = \begin{cases} 
0 & , a_i < R_c^{(1)} \\
-\frac{\pi}{R_c^{(2)} - R_c^{(1)}} \sin \left( \frac{\pi(a_i - R_c^{(1)})}{R_c^{(2)} - R_c^{(1)}} \right) & , R_c^{(1)} < a_i < R_c^{(2)} \\
0 & , a_i > R_c^{(2)}
\end{cases}
$$

The dependencies of the bond length $a_i$ and the bond angle $\theta_i$ on the Green strain tensor and the curvature tensor can be traced back in Eqn. 3.17 and 3.20. The derivatives $\partial a_i / \partial C$ and $\partial a_i / \partial K$ can be derived by first differentiating Eqn. 3.17 with respect to $C$ and $K$. It is verified that the resulting formulae for $\partial a_i / \partial C$ and $\partial a_i / \partial K$ share the same mathematical syntax so that for convenience we use a wildcard quantity $\mathcal{P}$ to represent either $C$ or $K$.
\[
\frac{\partial \mathbf{a}}{\partial \mathbf{P}} = \begin{bmatrix}
\frac{\partial a_1}{\partial \mathbf{P}} \\
\frac{\partial a_2}{\partial \mathbf{P}} \\
\frac{\partial a_3}{\partial \mathbf{P}}
\end{bmatrix} = \begin{bmatrix}
S_1 \frac{\partial w_1^1}{\partial \mathbf{P}} + w_1 S_1' \left( w_1 \frac{\partial k_1}{\partial \mathbf{P}} + k_1 \frac{\partial w_1}{\partial \mathbf{P}} \right) \\
S_2 \frac{\partial w_2}{\partial \mathbf{P}} + w_2 S_2' \left( w_2 \frac{\partial k_2}{\partial \mathbf{P}} + k_2 \frac{\partial w_2}{\partial \mathbf{P}} \right) \\
\frac{1}{2} \left[ w_1^2 S_{12}^2 \left( w_1 \frac{\partial k_1}{\partial \mathbf{P}} + 2k_1 \frac{\partial w_1}{\partial \mathbf{P}} \right) + k_1 (w_1^1)^2 S_{12}' \left( w_1 \frac{\partial k_1}{\partial \mathbf{P}} + k_1 \frac{\partial w_1}{\partial \mathbf{P}} \right) \right] + \\
\frac{1}{2} \left[ w_2^2 S_{22}^2 \left( w_2 \frac{\partial k_2}{\partial \mathbf{P}} + 2k_2 \frac{\partial w_2}{\partial \mathbf{P}} \right) + k_2 (w_2^1)^2 S_{22}' \left( w_2 \frac{\partial k_2}{\partial \mathbf{P}} + k_2 \frac{\partial w_2}{\partial \mathbf{P}} \right) \right]
\end{bmatrix}
\]

(4.16)

where some operators are defined as

\[ S_n = S(k_n w^n), \quad S_{n2} = S(k_n w^n / 2), \quad S'_n = S'(k_n w^n), \quad S'_{n2} = S'(k_n w^n / 2), \quad S''_n = S''(k_n w^n), \quad S''_{n2} = S''(k_n w^n / 2), \quad \text{and} \quad S = \sin x / x, \quad S' = (\cos x - S) / x, \quad \text{and} \quad S'' = -(\sin x + 2S') / x, \quad n \text{ goes from } 1 \text{ to } 2. \]

For the three types of bond lengths and angles \( i = 1, 2, 3 \)

\[
\frac{\partial a_i}{\partial \mathbf{P}} = \frac{1}{a_i} \sum_{k=1}^{3} (a_i)^k \frac{\partial (a_i)^k}{\partial \mathbf{P}}
\]

and

\[
\frac{\partial \theta_i}{\partial \mathbf{P}} = -\frac{1}{a_i a_k \sin \theta_i} \left[ \sum_{c=1}^{3} (a_j)^c \frac{\partial (a_k)^c}{\partial \mathbf{P}} + \sum_{c=1}^{3} (a_k)^c \frac{\partial (a_i)^c}{\partial \mathbf{P}} - \cos \theta_i \left( a_i \frac{\partial a_k}{\partial \mathbf{P}} + a_k \frac{\partial a_i}{\partial \mathbf{P}} \right) \right]
\]

In the above formulae, \( a_i \) means the length of bond vector with type \( i \), i.e. \( a_i = \|a_i\| \); \((a_i)^k\) means the \( k \)-th component of bond vector with type \( i \), i.e. \( (a_i)^k = a_i^k \).

The derivative of tangent deformed bond vector \( \mathbf{w} \) with respect to \( \mathbf{c} \) and \( \mathbf{K} \) are given as
The derivatives of the principal directions are

\[
\frac{\partial w^m}{\partial \kappa} = C_{AB} A^A \frac{\partial (V_n)^B}{\partial \kappa} + \frac{1}{2} (A \otimes V_n + V_n \otimes A)
\]

and

\[
\frac{\partial w^n}{\partial \kappa} = C_{AB} A^A \frac{\partial (V_n)^B}{\partial \kappa}
\]

The derivatives of the principal directions are

\[
\frac{\partial V_n}{\partial \kappa} = \frac{1}{2(k_n - k_m)} \left( V_m \otimes (V_n \otimes V_m + V_m \otimes V_n) \right)
\]

and

\[
\frac{\partial V_n}{\partial \nu} = \frac{1}{2} V_n \otimes V_n - k_n \frac{\partial V_n}{\partial \kappa}
\]

In the above formulae, the indices \( m, n, A \) and \( B \) all go from 1 to 2, and \( A \) is the undeformed carbon bond vector, and \( V_n \) is the principal direction of the curvature tensor; the symbol \( \otimes \) represents the dyadic product operation about two vectors.

Derivatives of the Green strain tensor and the curvature tensor with respect to the degrees of freedom

Since the Green strain tensor is the pullback of the metric tensor \( g_{\alpha \beta} \), by reference to Eqn. 4.7, it can be proved that the derivatives of the metric tensor with respect to the degrees of freedom is

\[
\frac{\partial g_{\alpha \beta}}{\partial \phi_I} = N_{i, \alpha} g_{\beta} + N_{i, \beta} g_{\alpha}
\]

where the subscript \( I \) is the nodal ID. Then the Green strain tensor derivatives are
The derivative of the curvature tensor $k_{a\beta}$ with respect to the degrees of freedom is

$$\frac{\partial k_{a\beta}}{\partial \varphi_i} = \sum_{i=1}^{3} (g_{a,\beta})_i \frac{\partial n^i}{\partial \varphi_i} + N_{I,a\beta} n$$

and

$$\frac{\partial n^i}{\partial \varphi_i} = \frac{1}{\|g_1 \times g_2\|} \left[ i_i \times (N_{I,12}g_1 - N_{I,11}g_2) - n_i \frac{\partial \|g_1 \times g_2\|}{\partial \varphi_i} \right]$$

and

$$\frac{\partial \|g_1 \times g_2\|}{\partial \varphi_i} = \frac{1}{\|g_1 \times g_2\|} \left[ (N_{I,18}g_{22} - N_{I,12}g_{12})g_1 - (N_{I,18}g_{11} - N_{I,12}g_{12})g_2 \right]$$

where $i_i$ is the Cartesian unit vector and $\times$ represents the cross product. The derivative for the pullback curvature tensor is

$$\frac{\partial K_{AB}}{\partial \varphi_i} = (p^e)_A (p^e)_B \frac{\partial k_{a\beta}}{\partial \varphi_i}$$

Finally the derivatives of REBO energy density function with respect to the degrees of freedom, $\partial E_{\text{REBO}}/\partial \varphi_i$ can be computed by plugging all the boxed formulae into Eqn. 4.13. Normally this operation is performed on the element level, obtaining a $3 \times 1$ vector of $(F^e_{\text{internal}})_I$ at each node then this elemental internal force vector is added to the global force vector.
4.3.2 Lennard-Jones energy calculation

In the numerical evaluation of van der Waals forces for the continuum system, the interacting distances are calculated for the pairs of eligible quadrature points that are not bonded and lie within the valid cutoff ranges on the finite element mesh.

From Eqn. 3.35, the numerical integration formula is given as

\[
E_{LJ} = \frac{4}{\sigma_0^6} \int_{\Omega_0} \int_{\Omega_f} V(||r_{ij}(X_i, X_j)||) d\Omega_0 d\Omega_f
\]

where

\[
V(||r_{ij}(X_i, X_j)||) = \frac{1}{\sigma_0^6} \frac{dV}{d\epsilon} \frac{dV}{d\epsilon} N_i \det(T \phi_0^i) \det(T \phi_0^j) \omega_i \omega_j r_{ij}^{ef}
\]

The nonbonded force is between the two interacting elements, \( e \) and \( f \) obeying Newton's third law. It can be calculated as follows

\[
(F_{\text{nonbonded}})^e_f = \sum_{i=1}^{\text{nape}} \sum_{j=1}^{\text{nape}} \frac{1}{r_{ij}^{ef} \, \epsilon} \frac{dV}{d\epsilon} N_i \det(T \phi_0^i) \det(T \phi_0^f) \omega_i \omega_f r_{ij}^{ef}
\]

where \((F_{\text{nonbonded}})^e_f\) represents the nonbonded force component at node I on element \( e \). The nonbonded force at node I on element \( f \) is the opposite of it on element \( e \), i.e. \((F_{\text{nonbonded}})^e_f = -(F_{\text{nonbonded}})^f_e\)
4.3.3 External energy calculation

When an external force field is applied to the carbon tube, the force can be treated as the body force, $B$ with the unit of the force per unit undeformed area. Then the external energy $E_{\text{external}}$ is

$$E_{\text{external}} = \int_{\Omega_0} B \cdot \Phi d\Omega_0$$  \hspace{1cm} (4.19)

4.4 Energy minimizations

The total energy of the system is $E_{\text{total}} = E_{\text{REBO}} + E_{\text{LJ}} - E_{\text{external}}$. For a system having applied conserved forces (like the carbon nanotube cases described here) only, a stable configuration of the system will be the minimizer of its total energy, which provides a clean-cut methodology for this highly nonlinear problem.

4.4.1 Inner displacement relaxation

As have been discussed in Section 3.2.2, for the multilattice structure like graphene sheet, the inner displacement $\eta$ plays an important role in equalizing the internal energy on the lattice level. A direct appearance of the inner displacement effect can be found in Eqn. 4.12 which calculates the internal force vector. At that time we simply ignored the term $\frac{\partial W}{\partial \eta} \frac{\partial \eta}{\partial \phi}$ completely by saying it
will vanish by the inner relaxation. In this section, the detailed account of the method for the inner relaxation will be given.

Recall the REBO energy density is $W(C, \kappa, \eta)$. The inner relaxation numerically means to find the minimizer $\eta$ to $W$ under the given strain measures, $C$ and $\kappa$, i.e. to solve for $\eta$ so that

\begin{equation}
\frac{\partial W}{\partial \eta} = 0 \tag{4.20}
\end{equation}

and

\begin{equation}
\text{det} \left( \frac{\partial^2 W}{\partial \eta^2} \right) > 0 \tag{4.21}
\end{equation}

Once the minimizer $\eta$ is found, $\frac{\partial W}{\partial \eta}$ will be uniformly zero; the term $\frac{\partial W}{\partial \eta} \frac{\partial \eta}{\partial \varphi}$ can be removed from the REBO energy density. This is deemed as the inner energy minimization. Quasi-Newton's method is implemented because the value of inner displacement $\eta$ that minimizes $W$ is expected not to be very far from its current value and Quasi-Newton's method converges at a quadratic rate in this situation (Friedland et al., 1987).

A portion of Fortran code is presented in Listing 4.1. For an initial inner displacement $\eta$, the first and second directive of $W$, $\partial W / \partial \eta$ and $\partial^2 W / \partial \eta^2$ are evaluated; the inverse matrix of $\partial^2 W / \partial \eta^2$ is calculated. The updated inner displacement $\eta$ is found as, Line 115,
Listing 4.1: Quasi Newton's method for inner relaxation

\begin{verbatim}
do i = 1, NELE \! loop over every element
    ...
do j = 1, NQP \! loop over every quadrature point
    ...
        !assign inner displacement values from last converged \! step
        Eta_New(:,:) = Eta_Old(:,)
    ...
        !calculate 1st and 2nd derivative of W with respect to \! inner displacement, dWdEta, ddWddEta
        call dStrEDendEta(..., Eta_New, dWdEta, ...)
call ddStrEDenddEta(..., Eta_New, ddWddEta, ...)
    ...
        !calculate the increment of inner displacement
        dEta = Eta(..., dWdEta, ddWddEta, ...)
        !update inner displacement
        Eta_New = Eta_Old + dEta
        Eta_Old = Eta_New
        !evaluate new dWdEta
        call dStrEDendEta(..., Eta_New, dWdEta, ...)
    ...
    !calculate magnitude of dWdEta
    res = sqrt(abs(dWdEta(1)**2 + dWdEta(2)**2))
    ...
if (res .gt. tol) goto enddo /i

enddo /i
enddo /j
\end{verbatim}

\[
\Delta \eta^k = - \left[ \left( \frac{\partial^2 W}{\partial \eta^2} \right)^{-1} \frac{\partial W}{\partial \eta} \right]^k
\]

(4.22)

and

\[
\eta^{k+1} = \eta^k + \Delta \eta^k
\]

(4.23)

The \( \partial W / \partial \eta \) is then evaluated again with the newly found \( \eta \). If \( \| \partial W / \partial \eta \| \geq \) .tol., the iteration continues at Line 110, otherwise exit with the minimizer \( \eta \).
Normally the number of iterations ranges from two to four steps.

The inner displacement $\eta$ enters expressions for undeformed bond vectors at Eqn. 3.21 through 3.23. The derivatives $\partial W / \partial \eta$ and $\partial^2 W / \partial \eta^2$ are given as

$$\frac{\partial W}{\partial \eta} = \sum_{i=1}^{6} \frac{\partial W}{\partial P_i} \frac{\partial P_i}{\partial \eta}$$  \hspace{1cm} (4.24)

and

$$\frac{\partial^2 W}{\partial \eta^2} = \sum_{i=1}^{6} \left[ \frac{\partial W}{\partial P_i} \frac{\partial^2 P_i}{\partial \eta^2} + \frac{\partial^2 W}{\partial P_i \partial \eta} \right] \hspace{1cm} (4.25)$$

$$\sum_{i<j<6} \frac{\partial^2 W}{\partial P_i \partial P_j} \left( \frac{\partial P_i}{\partial \eta} \frac{\partial P_j}{\partial \eta} + \frac{\partial P_i}{\partial \eta} \frac{\partial P_j}{\partial \eta} \right)$$  \hspace{1cm} (4.26)

where $P_i$ represents three inequivalent bond lengths and angles $(a_1, a_2, a_3, \theta_1, \theta_2, \theta_3)$.

And the derivatives of them are given below

\[
\frac{\partial a}{\partial \eta} = \begin{bmatrix}
\frac{\partial a_1}{\partial \eta} \\
\frac{\partial a_2}{\partial \eta} \\
\frac{\partial a_3}{\partial \eta}
\end{bmatrix} = \begin{bmatrix}
(S_1 + k_1 w^1 S_1') \frac{\partial w^1}{\partial \eta} \\
(S_2 + k_2 w^2 S_2') \frac{\partial w^2}{\partial \eta} \\
k_1 w^1 S_{12} \left( S_{12} + \frac{1}{2} k_1 w^1 S_{12}' \right) \frac{\partial w^1}{\partial \eta} + \\
k_2 w^2 S_{22} \left( S_{22} + \frac{1}{2} k_2 w^2 S_{22}' \right) \frac{\partial w^2}{\partial \eta}
\end{bmatrix}
\]

where $\partial w^w_{\eta} = CV_n$ with $n$ being 1 or 2. And the operator $S_1, S_2$, etc. are defined below Eqn. 4.16. And
\[
\frac{\partial a_i}{\partial \eta} = \frac{1}{a_i} \sum_{k=1}^{3} (a_i)^k \frac{\partial (a_i)^k}{\partial \eta}
\]

and

\[
\frac{\partial \theta_i}{\partial \eta} = \frac{-1}{a_i a_k \sin \theta_i} \left[ \sum_{c=1}^{3} (a_j)^c \frac{\partial (a_k)^c}{\partial \eta} + \sum_{c=1}^{3} (a_k)^c \frac{\partial (a_j)^c}{\partial \eta} - \cos \theta_i \left( a_j \frac{\partial a_k}{\partial \eta} + a_k \frac{\partial a_j}{\partial \eta} \right) \right]
\]

where \((i, j, k)\) follows the even permutation of \((1, 2, 3)\).

The second derivatives of \(a_i\) and \(\theta_i\) with respect to \(\eta\) are

\[
\frac{\partial^2 a}{\partial \eta^2} = \begin{bmatrix}
  k_1 \left( 2S_1'' + k_1 w^1 S_1'' \right) \frac{\partial w^1}{\partial \eta} \otimes \frac{\partial w^1}{\partial \eta} \\
  k_2 \left( 2S_2'' + k_2 w^2 S_2'' \right) \frac{\partial w^2}{\partial \eta} \otimes \frac{\partial w^2}{\partial \eta} \\
  k_1 \left[ \left( S_{12} + \frac{1}{2} k_1 w^1 S_{12}' \right)^2 + k_1 w^1 S_{12}' \left( S_{12}' + \frac{1}{2} k_1 w^1 S_{12}'' \right) \right] \frac{\partial w^1}{\partial \eta} \otimes \frac{\partial w^1}{\partial \eta} + \\
  k_2 \left[ \left( S_{22} + \frac{1}{2} k_2 w^2 S_{22}' \right)^2 + k_2 w^2 S_{22}' \left( S_{22}' + \frac{1}{2} k_2 w^2 S_{22}'' \right) \right] \frac{\partial w^2}{\partial \eta} \otimes \frac{\partial w^2}{\partial \eta}
\end{bmatrix}
\]

The second derivatives of bond length \(a_i\) and angle \(\theta_i\) are

\[
\frac{\partial^2 a_i}{\partial \eta^2} = \frac{1}{a_i} \left[ \sum_{k=1}^{3} \frac{\partial (a_i)^k}{\partial \eta} \otimes \frac{\partial (a_i)^k}{\partial \eta} + \sum_{k=1}^{3} (a_i)^k \frac{\partial^2 (a_i)^k}{\partial \eta^2} - \frac{\partial a_i}{\partial \eta} \otimes \frac{\partial a_i}{\partial \eta} \right]
\]

and

\[
\frac{\partial^2 \theta_i}{\partial \eta^2} = \frac{-1}{\sin \theta_i} \left[ \frac{\partial^2 \cos \theta_i}{\partial \eta^2} + \cos \theta_i \frac{\partial \theta_i}{\partial \eta} \otimes \frac{\partial \theta_i}{\partial \eta} \right]
\]

where
\[
\frac{\partial^2 \cos \theta_i}{\partial \eta^2} = \frac{1}{a_j a_k} \left[ \frac{\partial^2 (a_j \cdot a_k)}{\partial \eta^2} - \left( \frac{\partial \cos \theta_i}{\partial \eta} \otimes (a_j \frac{\partial a_k}{\partial \eta} + a_k \frac{\partial a_j}{\partial \eta}) + (a_j \frac{\partial a_k}{\partial \eta} + a_k \frac{\partial a_j}{\partial \eta}) \otimes \frac{\partial \cos \theta_i}{\partial \eta} \right) \right]
\]

and

\[
\frac{\partial^2 (a_j \cdot a_k)}{\partial \eta^2} = \sum_{c=1}^{3} (a_j)^c \frac{\partial^2 (a_k)^c}{\partial \eta^2} + \sum_{c=1}^{3} (a_k)^c \frac{\partial^2 (a_j)^c}{\partial \eta^2} + \sum_{c=1}^{3} \left( \frac{\partial (a_j)^c}{\partial \eta} \otimes \frac{\partial (a_k)^c}{\partial \eta} + \frac{\partial (a_k)^c}{\partial \eta} \otimes \frac{\partial (a_j)^c}{\partial \eta} \right)
\]

where \((i, j, k)\) follows the even permutation of \((1, 2, 3)\).

### 4.4.2 Global energy minimization

The global energy minimization involves minimizing the total energy

\[E_{\text{total}} = E_{\text{REBO}} + E_{\text{LJ}} - E_{\text{external}}\]

aiming at finding the equilibrium configuration for that specific loading condition. An algorithm called L-BFGS-B proposed by Byrd et al. (1995); Zhu et al. (1994) is used for this purpose. It is a limited memory algorithm specialized for solving large nonlinear optimization problems subject to simple bounds on the variables, as defined in Eqn. 4.27

\[
\min f(x) \quad \text{(4.27)}
\]

subject to

\[l \leq x \leq u\]
where \( l \) and \( u \) are the lower and upper bound of variable \( x \). Three minimization methods were implemented: direct primal, primal CG, and dual. At each iteration a limited memory BFGS approximation to the Hessian matrix is updated, so only the gradient of the objective function is required while the second order derivatives (the Hessian) are not.

4.5 Structure of the code

A Fortran computer program has been created following the above mentioned algorithm. It’s tested and validated with various published results. An overview of the code structure is provide below for reference.

1. Read in problem definitions: \((n_1, n_2)\) for CNT; length \( L \); perimeter \( W \); the number of elements;
2. Generate a rectangular mesh with the size of \( L \times W \) on the undeformed configuration;
3. Analyze the mesh; generate ghost elements and build one neighbor list to keep track of each node in orderly sequence see Fig. 4.3;
4. Calculate pullback tensor and stored in a global array for quick access;
5. Generate a cylindrical mesh on the deformed configuration with length of \( L \) and radius of \( W/2\pi \);
6. Repeat step 3;
7. INITIAL GLOBAL MINIMIZATION Minimize the total energy \( E_{\text{total}} \) on the deformed configuration using LBFGS-B method without applying any loadings or constraints;
8. Update the mesh with the one obtained from step 7;

9. **LOADING STEP GLOBAL MINIMIZATION** Apply the desired loading in small increments and properly constrain the carbon nanotube; using LBFGS-B method to minimize the total energy $E_{\text{total}}$ on the equilibrium configuration from the last loading step

10. Update the mesh

11. If the loading has been fully applied then exit or else go back to step 9

12. Exit

In both global minimization steps 7 and 9, the code is written in the reverse-communication mode where the subroutines for evaluating the total energy and its gradient are called repeatedly. The inner displacement relaxation is performed every time either one of them is called by the LBFGS-B driver.
Chapter 5
VALIDATIONS AND NUMERICAL EXAMPLES

5.1 Two dimensional carbon atomic ring

In the first validation case, a two dimensional ring made of carbon atoms interacting with the substrate was modelled using the general continuum method, and the results were compared with those obtained by molecular dynamics calculations.

Assume we are considering a (20, 0) zig-zag carbon nanotube and we are only interested in its transverse deformations. A cut normal to the tube’s axial...
direction will produce a circle with twenty equally distributed carbon atoms. And the carbon ring is placed a little bit over 2Å above the rigid carbon substrate with 41 atoms, which makes the two objects positioned within the valid van der Waals interaction range. A point loading $P$ is applied at the top atom of the ring.

To solve this problem using molecular dynamics method, a total of 183 degrees of freedom need to be solved for, see Fig. 5.1. If the general continuum method is used, since the finite element nodes are not necessarily coincident with the atomic sites, the smaller number of nodes could be used. Fig. 5.2 demonstrates a finite element mesh that's used for the modelling where there are 12 finite element nodes on the ring and 17 nodes on the rigid carbon substrate. The reduction of the number of degrees of freedom becomes much more manifest as the size of problem grows larger especially in the three dimensional
cases, as we will see later.

![Graph](image)

(a) Initial configuration.

(b) After the relaxation.

**Figure 5.3:** van der Waals force between the ring and the substrate.

van der Waals force although contributing very little to the total energy of the carbon nanotube system, it’s of crucial effect in determining deformations in the transverse direction and the interactions with other objects. In Fig.
5.3, the van der Waals forces between the ring and the substrate are plotted for the two different configurations. The search for the active interacting pairs is conducted between the quadrature points of each element in the ring and those of each element in the substrate. Fig. 5.3 (a) shows the van der Waals force distribution after the meshes are initially generated. It is quite obvious that the forces are not in equilibrium, since all the force vectors are pointing upward. This is reasonable because the initial mesh of the ring are simply the isometric transformation from the straight line to the circular configuration where the lengths of element are preserved, in other words there will be no strains when the deformation happens. This is not physical as a result the initial relaxation of the system is necessary which allows the system to rearrange itself to adopt a more stable and equilibrium configuration.

It can be seen from Fig. 5.3 (b) that after the initial relaxation, the carbon ring under the influence of van der Waals forces becomes flattened on the bottom. More areas of the ring take part in the van der Waals interactions with the substrate. The central part of the flattened region receives repulsive forces that try to push away the ring while the side parts having attractive forces which try to hold the ring in place. Under these two counteracting mechanisms, an equilibrium configuration could be found through the minimization of the total energy of the ring and the substrate.
The case of a point loading $P$ applied at the top atom of the ring, see Fig. 5.1 and 5.2, is calculated using both the general continuum method described in this thesis and the molecular dynamics code, GULP (General Utility Lattice Program) developed by Gale and Rohl (2003). And their results are plotted in Fig. 5.4 with one overlapping the other, where the black dots are the results obtained from molecular dynamics while the continuous red line is from the general continuum method. It shows that the prediction for the carbon ring deformation by both methods are almost identical. And the van der Waals forces are shown in arrows. They are all pointing upward to balance the load coming from the top of the ring.
5.2 Elastic properties of carbon nanotubes

In the first validation test for this section, the REBO strain energy relative to that of the unstrained planar graphene sheet per unit atom for the zigzag and armchair type of carbon nanotubes are calculated with different tube radii, i.e.

\[ \frac{E_{\text{REBO}}_{\text{CNT}}}{n} - \frac{E_{\text{REBO}}_{\text{graphene sheet}}}{n}, \]

where \( E_{\text{REBO}}_{\text{graphene sheet}} / n = -7.3756eV \) and \( n \) is the number of carbon atoms, see Eqn. 1.8 for details.

The smallest CNT used for the numerical test is (5,0) zigzag and (3,3) armchair carbon nanotubes with the corresponding radius range going from about 2\( \AA \) to nearly 10\( \AA \). And a length of 40\( \AA \) is used for every tube. A cylindrical mesh is built for each case to meet the radius and length specifications using the simple mapping relations described in Eqn. 4.1. The mesh size is intentionally kept constant for the tubes with different sizes. There are ten elements around the circumferential direction and twenty along the tubule axis direction. We can do this is because in the general continuum method, the finite element node does not have to be stuck with the underlying carbon atom. As a result it's the user's freedom to decide the type and density of the mesh. Although the number of carbon atoms increases with the radius, the computa-
Figure 5.5: Variation of Carbon nanotube strain energy (relative to graphene sheet) with respect to radii.

The results are plotted against the data calculated by Robertson et al. (1992) using the same interatomic potential. It can be seen that the relative energy v.s. radius curves for both zigzag and armchair correlate very well with
Robertson et al. (1992)'s results. It's found that with the increase of the tube radius, the relative energy per unit atom decreases and when the tube becomes very large, the value approaches to zero, which means the energy of large carbon nanotube is close to that of a graphene sheet. And it is also found that for both the zigzag and the armchair carbon nanotubes, the energy decreases with the increase of the tubule radius and the dependence is fitted to be $E \propto 1/R^2$, no influence of lattice structure involved.

Next we consider a (5,5) carbon nanotube subjected to the tensile load. The numerical setup is shown in Fig. 5.6. One end of the tube is totally constrained while the other end is provided with a prescribed displacement. It simulates the situation in the actual tensile test where one end of the specimen is firmly gripped by an apparatus and the other is pulled out in the axial direc-
tion. The tube is to be stretched by 15% of its original length. The stretching ratio is divided up into a series of small increments; hence at every loading step, the nodes on the right end will be moved along the axial direction by certain amount of distance governed by the current increment. Then the structure is optimized by minimizing the total energy for this new configuration and during the optimization both ends are fixed. The strain energy per unit atom is computed at different level of strains and plotted against the result for the same problem using the molecular dynamics method by Robertson et al. (1992). It shows that the results obtained by two methods agree well in pre-

\[ \text{Strain Energy (eV/atom)} \]

\[ \text{Strain} \]

\[ 0.02 \quad 0.04 \quad 0.06 \quad 0.08 \quad 0.1 \quad 0.12 \quad 0.14 \]

Figure 5.7: Variation of Carbon nanotube strain energy with respect to strain.
dicting the quadratic growth of strain energy with the increase of the strain. Fig. 5.8 provides the snapshots of the deformed configurations at two different instances of the tensile test. In both cases, the left ends of the tube are fixed and the tensile load is applied at the right end. Fig. 5.8 (a) represents the situation at the early stage of the test, but the necking is obviously noticeable; while the second figure shows how the carbon nanotube looks like when it's stretched 15% longer. The necking region has moved towards where the tensile load is applied. The simulation shows that the carbon nanotube is actually quite ductile which is on the contrary to the usual perception of brittleness of graphite materials. This observation is backed by the report from Huang et al. (2006a), where they claimed that theoretical maximum tensile strain of a single-walled carbon nanotube is almost 20%. At high temperatures, individual CNTs can become nearly 280% longer.

Finally the estimation of Young's modulus of single walled carbon nanotube is explored. The concept of Young's modulus is naturally related to the three dimensional continuum body for correlating stress and strain. Carbon nanotubes are made of graphene monolayers which is actually a two-dimensional manifold embedded in the three-dimensional space. So the wall thickness is not well defined. Different authors using different methods choose widely scattered wall thickness $h$ which leads to quite different estimates of
(a) An intermediate snapshot.

(b) Final deformed configuration.

Figure 5.8: Stretched carbon nanotube.
Young’s modulus for carbon nanotube, as cited in the paper by Huang et al. (2006b), the reported values ranging from 0.974 TPa to 5.07 TPa. The graphene interlayer distance of 3.4 Å was generally accepted as the effective wall thickness. To be comparable with others, here it is used as the wall thickness. The tensile test was numerically used to determine the Young’s modulus of carbon nanotube under the condition of infinitesimal deformation. One end of the tube is completely constrained while the other end is under a uniform atomic force. By several trial runs, we can estimate the appropriate values for the force, then the tube is optimized and will be in a new equilibrium state with a new length, which leads to the determination of strain. With the wall thickness being 3.4 Å, the normal stress could be found. A series of zigzag carbon nanotubes with different radii are calculated and the results are plotted in Fig. 5.9, where triangular symbols are for the values from Arroyo and Belytschko (2002), and the square ones from the current method. The lines are simply for guiding purpose.

5.3 Buckling of carbon nanotubes

As seen in the previous section, carbon nanotubes are very ductile and can endure rather large deformations without failure. It can develop into different morphological patterns accompanied by abrupt energy release and sin-
Figure 5.9: The Young's modulus based on wall thickness of 3.4\(\text{Å}\) compared with the values from (Arroyo and Belytschko, 2002).

A (7,7), 60\(\text{Å}\) long carbon nanotube subjected to the axial compression load is modelled for its buckling patterns. The tube was initially relaxed before any loading is applied. Then every nodal point on one end of the tube is shifted along the axial direction by a small distance at every loading step while the other end is constrained. The results from the current method is compared with those by molecular dynamics method. For the atomic model, the carbon nanotube has 670 carbon atoms which amounts to 2,010 degrees of freedom in the molecular dynamics simulation; while for the general continuum method, the
mesh is designed to have 10 nodes along the circumference and 21 along the longitudinal direction, which gives the size of the problem at 630 degrees of freedom, about 70% less than the molecular dynamics simulation.

![Fig. 5.10: Dramatic change in carbon nanotube's morphological patterns.](image)

Fig. 5.10 demonstrates a dramatic shape change of carbon nanotube at the strain level in the neighborhood of 5% where the buckling starts to happen. Fig. 5.10 (a) shows the mesh used and the configuration of the surface before it buckles. There are two light necking regions near the endings. When the tube
is further compressed by the amount of 0.3%, a completely different buckled configuration is adopted by the carbon nanotube as shown in Fig. 5.10 (b). The key features for carbon nanotube buckling are apparently visible, such as a pair of symmetric fins and the neckings above and below the fins. Fig. 5.11 gives two different viewing perspectives of the buckling of the tube. It shows that there are three necking regions with one perpendicular to the other two. After the initial buckling, with the compression being continuously applied at small
(a) The deformed mesh for the second structural instability. (b) The smooth surface for the second structural instability generated by subdivision algorithm.

Figure 5.12: The onset of the second instability.

increments, the tube will keep the straight tubule axis and the morphological pattern resembles what have been shown above. At about 12% of strain, the straight axis configuration will be destroyed and the tube buckles sideway and the symmetric geometry is lost. These phenomenon can be seen in Fig. 5.12 and again on the left is the actual mesh used for the calculation; on the right is the limit surface by subdivision algorithm which was discussed in Chap-
ter 2. A comparison between the results from the current method with those

from the molecular dynamics method, shown in Fig. 5.13 indicates that the key features, i.e. the fins and the adoption of the asymmetric configuration in the later stage of compression are all predicted properly. The strain energy versus strain is shown in Fig. 5.14 compared with that from Yakobson et al. (1996). It is found that before the first buckling, the strain energy grows nearly at the quadratic rate with the increase of the strain. At around 5% strain value, the tube buckles as shown in Fig. 5.11 where the stored strain energy releases and a new configuration other than cylindrical shape is adopted for this energy level. Continued compression causes the strain energy to grow almost linearly with the strain until the second instability event happens at around 12% of strain value, as shown in Fig. 5.12. In Yakobson et al. (1996)'s calculation, there were
two other buckled configurations not present in our calculation. The reasons are probably because of the nature of the problem. In both methods, the energy minimization were used (conjugate-gradient for Yakobson et al. (1996)'s case and LBFGS-B for the current one). It is a highly nonlinear optimization problem, the calculated minimum configuration may be obtained following different equilibrium paths depending on the step size, convergence tolerance and etc. But from the above comparisons, both methods agrees well.
5.4 Twisting of carbon nanotube

A (13,0), 230Å long carbon nanotube is subjected to the twisting load as shown in Fig. 5.15 which is implemented by turning the nodes on two ends of the tube a small angle in the opposite directions about $x$ axis. On the atomic model, the tube should have 2,750 carbon atoms, which correspond to 8,248 degrees of freedom. The mesh size used for the current method is 12 nodes on the circumference and 51 along the longitudinal direction, which is 612 nodes or 1,836 degrees of freedom, about only 22.3% of the size of problem solved by molecular dynamics method. To see the nonlinear behavior of carbon nanotube, it will be twisted well beyond the linear range. In this case the tube will
be turned up to 516°. Fig. 5.16 (a) shows what the tube looks like before the initial instability occurs. At around 154°, the structure yields and the twisting of surface is obvious in Fig. 5.16 (b). The twisted configuration is compared with the one obtained from the molecular dynamics, shown in Fig. 5.16 (c). Both results show that under twisting, the carbon nanotube surface changes from cylindrical shape to the flattened and helix spiral but still maintains a straight axis and keeps the symmetry.

Continued twisting will cause the carbon nanotube to buckle into side and to lose symmetry about the axial axis. This severe deformation is shown in Fig. 5.17, where the second graph is obtained from the molecular dynamics. Fig. 5.18 shows the strain energy variation with the change of twisting angles. The red line is from the current calculation and the green line with black dots from Yakobson et al. (1996). They agreed well in the quadratic region; the difference becomes noticeable for large angle twistings, where a finer mesh is needed. But the overall trend between two methods are aligned.
(a) The carbon nanotube right before the onset of the first instability happens.

(b) Twisting effects are noticeable at angle of 154 °.

(c) The molecular dynamics modelling of the same problem at angle of 154° (Yakobson et al., 1996).

Figure 5.16: The onset of the first instability and comparison with molecular dynamics simulation.
(a) The carbon nanotube buckles sideway.

(b) The carbon nanotube buckles sideway by molecular dynamics method (Yakobson et al., 1996).

Figure 5.17: The side buckling of twisted carbon nanotubes and comparison with molecular dynamics simulation.
Figure 5.18: Strain energy vs. twisting angles.
Chapter 6
CONCLUSIONS AND FUTURE PLAN

In this thesis work, a Fortran program was developed for modelling mechanical behaviors of carbon nanotube. The central idea is that carbon nanotubes resemble the continuum thin shell structures and the modified Cauchy-Born rule can be used to link the atomic scale events to the macroscale events. We combined the newly proposed thin shell subdivision finite element framework with Brenner’s bond-order atomic potential for hydrocarbons (REBO) and derived and implemented the general continuum method that faithfully reflects the fact that carbon nanotube is a rolled-up carbon monolayer without thickness and preserve its atomic/crystallite characteristics.

The numerical validation cases show that the method can correctly predict key mechanical features of carbon nanotube and because of the finite element method used the computational cost is significantly lower than those of corresponding molecular dynamics method. Based on the work of this thesis, much more complex carbon nanotube structures could be studied.

Although the problem size could be reduced by using this method, the computational cost is still high for complex settings of carbon nanotubes be-
cause of the double optimizations involved in the algorithm (inner displacement relaxation and global energy minimization). As one important task in the future plan, the parallelization of the code is necessary. And more challenging computational cases needs to be tackled such as multitubes intercrossing within a matrix; study of superelastic carbon nanotubes; study of dynamic behavior of carbon nanotubes; even the study of a nanotube of different material by using an appropriate atomic potential (the potential form is not restricted to empirical ones; quantum mechanics based are also possible, see Tadmor et al. (1999)).
Appendices
Appendix A

Picking matrix

Figure A.1: The picking matrix for subpatch 1.
Figure A.2: The picking matrix for subpatch 2.

Figure A.3: The picking matrix for subpatch 3.
Appendix B

Green-Lagrange strain tensor of the shell

Since \( E_{ij} = \frac{1}{2} (g_{ij} - \bar{g}_{ij}) \), if expanding the tensors and throwing out all the terms that contain \((\theta^3)^2\) then we will have

\[
\begin{align*}
\bar{g}_{11} &= \mathbf{a}_1 \cdot \mathbf{a}_1 + 2\theta^3 \mathbf{a}_1 \cdot \mathbf{a}_{3,1} \\
\bar{g}_{12} &= \mathbf{a}_1 \cdot \mathbf{a}_2 + 2\theta^3 \mathbf{a}_1 \cdot \mathbf{a}_{3,2}
\end{align*}
\]

In Eqn.B.2, we've made use of these relations
1. since \( \mathbf{a}_2 \cdot \mathbf{a}_3 = 0 \), \( \mathbf{a}_2 \cdot \mathbf{a}_{3,1} = -\mathbf{a}_3 \cdot \mathbf{a}_{2,1} \); by the same fashion \( \mathbf{a}_1 \cdot \mathbf{a}_{3,2} = -\mathbf{a}_3 \cdot \mathbf{a}_{1,2} \)
2. \( \mathbf{a}_{2,1} = \mathbf{a}_{1,2} \)
3. hence \( \bar{g}_{12} = \mathbf{a}_1 \cdot \mathbf{a}_2 - 2\theta^3 \mathbf{a}_3 \cdot \mathbf{a}_{1,2} \)

For the rest, \( \bar{g}_{21} = \bar{g}_{12}, \bar{g}_{22} = \mathbf{a}_2 \cdot \mathbf{a}_2 + 2\theta^3 \mathbf{a}_2 \cdot \mathbf{a}_{3,2}, \bar{g}_{23} = \mathbf{a}_2 \cdot \mathbf{a}_3, \bar{g}_{31} = \bar{g}_{13}, \bar{g}_{32} = \bar{g}_{23}, \) and \( \bar{g}_{33} = \mathbf{a}_3 \cdot \mathbf{a}_3 \). Now if we use Kirchhoff-Love assumption then \( \mathbf{a}_a \cdot \mathbf{a}_3 = 0 \) and \( \|\mathbf{a}_3\| = 1 \). Thus above expressions could be further simplified...
and if we further put all of them into the matrix form,

\[ g_{ij} = \begin{bmatrix} a_1 \cdot a_1 - 2\theta^3 a_{1,1} \cdot a_3 & a_1 \cdot a_2 - 2\theta^3 a_{1,2} \cdot a_3 & 0 \\ a_2 \cdot a_1 - 2\theta^3 a_{2,1} \cdot a_3 & a_2 \cdot a_2 - 2\theta^3 a_{2,2} \cdot a_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

Following the same method, we can derive \( \tilde{g}_{ij} \) for the reference configuration,

\[ \tilde{g}_{ij} = \begin{bmatrix} \tilde{a}_1 \cdot \tilde{a}_1 - 2\theta^3 \tilde{a}_{1,1} \cdot \tilde{a}_3 & \tilde{a}_1 \cdot \tilde{a}_2 - 2\theta^3 \tilde{a}_{1,2} \cdot \tilde{a}_3 & 0 \\ \tilde{a}_2 \cdot \tilde{a}_1 - 2\theta^3 \tilde{a}_{2,1} \cdot \tilde{a}_3 & \tilde{a}_2 \cdot \tilde{a}_2 - 2\theta^3 \tilde{a}_{2,2} \cdot \tilde{a}_3 & 0 \\ 0 & 0 & 1 \end{bmatrix} \]

\( E_{ij} = \frac{1}{2}(g_{ij} - \tilde{g}_{ij}) \), hence the matrix form for \( (g_{ij} - \tilde{g}_{ij}) \) will be

\[ \begin{bmatrix} a_1 \cdot a_1 - \tilde{a}_1 \cdot \tilde{a}_1 + 2\theta^3 (\tilde{a}_{1,1} \cdot \tilde{a}_3 - a_{1,1} \cdot a_3) & a_1 \cdot a_2 - \tilde{a}_1 \cdot \tilde{a}_2 + 2\theta^3 (\tilde{a}_{1,2} \cdot \tilde{a}_3 - a_{1,2} \cdot a_3) & 0 \\ a_2 \cdot a_1 - \tilde{a}_2 \cdot \tilde{a}_1 + 2\theta^3 (\tilde{a}_{2,1} \cdot \tilde{a}_3 - a_{2,1} \cdot a_3) & a_2 \cdot a_2 - \tilde{a}_2 \cdot \tilde{a}_2 + 2\theta^3 (\tilde{a}_{2,2} \cdot \tilde{a}_3 - a_{2,2} \cdot a_3) & 0 \\ 0 & 0 & 0 \end{bmatrix} \]

So, \( E_{ij} = \frac{1}{2}(a_i \cdot a_j - \tilde{a}_i \cdot \tilde{a}_j) + \theta^3 (a_\alpha \cdot a_{3,\beta} - \tilde{a}_\alpha \cdot \tilde{a}_{3,\beta}) = a_{ij} + \theta^3 \beta_{\alpha\beta} \), where \( i \) and \( j \) go from 1 to 3 and \( \alpha \) and \( \beta \) from 1 to 2.
Appendix C

Linearizing the bending strains

To linearize the bending strains $\beta_{\alpha\beta} = \bar{a}_{\alpha,\beta} \cdot \bar{a}_3 - a_{\alpha,\beta} \cdot a_3$, Newton-type linearization method was used

$$L[\beta(\bar{x} + u)] = \beta(\bar{x}) + \frac{d}{de} [\beta(\bar{x} + eu)]_{e=0} \cdot u$$  \hspace{1cm} (C.1)

$\bar{a}_{\alpha,\beta}$ and $\bar{a}_3$ are already in known states so no further actions will be taken for these two terms.

$$a_{\alpha,\beta} = \frac{\partial^2 x}{\partial \theta^\alpha \partial \theta^\beta}$$  \hspace{1cm} (C.2)

$$a_3 = \frac{a_1 \times a_2}{||a_1 \times a_2||} = \frac{\frac{\partial x_i}{\partial \theta^1} \times \frac{\partial x_j}{\partial \theta^2}}{||\frac{\partial x_i}{\partial \theta^1} \times \frac{\partial x_j}{\partial \theta^2}||}$$  \hspace{1cm} (C.3)
Now refer to eqn.C.1, $\beta(\bar{x}) = 0$, so $L[\beta(\bar{x} + \mathbf{u})] = \frac{d}{d\epsilon}[\beta(\bar{x} + \epsilon\mathbf{u})]_{\epsilon=0} \cdot \mathbf{u}$. When substituting $x = \bar{x} + \epsilon\mathbf{u}$ into $\beta(x)$, we'll get

$$\beta(\bar{x} + \epsilon\mathbf{u}) = \bar{a}_{h,\beta} \cdot \bar{a}_3 - \frac{\partial^2 (\bar{x} + \epsilon\mathbf{u})}{\partial \theta^a \partial \theta^\beta} \cdot \frac{\partial (\bar{x} + \epsilon\mathbf{u})}{\partial \theta^1} \times \frac{\partial (\bar{x} + \epsilon\mathbf{u})}{\partial \theta^2} \| \frac{\partial (\bar{x} + \epsilon\mathbf{u})}{\partial \theta^1} \times \frac{\partial (\bar{x} + \epsilon\mathbf{u})}{\partial \theta^2} \|$$

(C.4)

When eqn.C.4 is fully expanded, we will have the following

$$\beta(\bar{x} + \epsilon\mathbf{u}) = \bar{a}_{h,\beta} \cdot \bar{a}_3 - \left( \frac{\partial^2 \bar{x}}{\partial \theta^a \partial \theta^\beta} + \epsilon \frac{\partial^2 \mathbf{u}}{\partial \theta^a \partial \theta^\beta} \right) \cdot \left[ \frac{\partial \bar{x}}{\partial \theta^1} \times \frac{\partial \bar{x}}{\partial \theta^2} + \epsilon \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} + \epsilon^2 \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} \right]$$

(C.5)

We denote the denominator of the last factor in eqn.C.5 by $\sqrt{\bar{a}}$ and noting that $\sqrt{\bar{a}}|_{\epsilon=0} = \sqrt{\bar{a}}$, where $\sqrt{\bar{a}} \equiv \| \bar{a}_1 \times \bar{a}_2 \|$. Now considering $\frac{d}{d\epsilon}[\beta(\bar{x} + \epsilon\mathbf{u})]_{\epsilon=0}$ to
eqn.C.5, term \( A \) yields to zero. Hence

\[
- \frac{d}{de} \left[ \beta(\mathbf{x} + e\mathbf{u}) \right]_{e=0} = \frac{\partial^2 \mathbf{u}}{\partial \theta^a \partial \theta^b} \cdot \frac{\partial x}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} + \frac{\partial^2 \mathbf{x}}{\partial \theta^a \partial \theta^b} \cdot \frac{\partial u}{\partial \theta^a} + \frac{\partial^2 \mathbf{x}}{\partial \theta^a \partial \theta^b} \cdot \frac{\partial u}{\partial \theta^a} \\
\left( \frac{\partial u_{\alpha \beta}}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} + \frac{\partial x}{\partial \theta^a} \times \frac{\partial u_{\alpha \beta}}{\partial \theta^b} \right) \cdot \left( \frac{\partial u}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) - \left( \frac{\partial u_{\alpha \beta}}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) \cdot \left( \frac{\partial u}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) \cdot \left( \frac{\partial u_{\alpha \beta}}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) \cdot \left( \frac{\partial u}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) \cdot \left( \frac{\partial u}{\partial \theta^a} \times \frac{\partial x}{\partial \theta^b} \right) \\
= u_{\alpha \beta} \cdot \mathbf{a}_3 + \frac{1}{\sqrt{a}} \left[ -u_{1,1} \cdot (\mathbf{a}_{\alpha \beta} \times \mathbf{a}_2) - u_{2,2} \cdot (\mathbf{a}_1 \times \mathbf{a}_{\alpha \beta}) \right] \cdot \frac{\mathbf{a}_3 \cdot \mathbf{a}_{\alpha \beta}}{\sqrt{a}} \cdot \frac{\partial \sqrt{a}}{\partial e} \bigg|_{e=0} \\
\text{(C.6)}
\]

It needs to differentiate the denominator \( \sqrt{a} \) in eqn.C.5 by first expressing it in indicial form using permutation symbols. Hence in the following section, I will present detailed steps as to how to evaluate part \( H \) shown in eqn.C.6.

\[
\sqrt{a}^2 = \left\| \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} + e^2 \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} \right\|^2 \\
= \left( \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} + e^2 \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} \right) \cdot \mathbf{A} \\
\left( \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} + e \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} + e^2 \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} \right) \cdot \mathbf{A} \\
\text{(C.7)}
\]
By using the following shorthands

\[
\begin{align*}
\frac{\partial \mathbf{x}}{\partial \theta^1} &= \mathbf{\hat{a}}_1 & \frac{\partial \mathbf{x}}{\partial \theta^2} &= \mathbf{\hat{a}}_2 \\
\frac{\partial \mathbf{u}}{\partial \theta^1} &= \mathbf{u}_1 & \frac{\partial \mathbf{u}}{\partial \theta^2} &= \mathbf{u}_2
\end{align*}
\]  
(C.8)  
(C.9)

and using indicial notation, we'll have

\[
\begin{align*}
\varepsilon \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} &= \mathbf{\hat{a}}_1 \times \mathbf{\hat{a}}_2 = (\mathbf{\hat{a}}_1)_i (\mathbf{\hat{a}}_2)_j \varepsilon_{ijk} \mathbf{e}_k = a_i b_j \varepsilon_{ijk} \mathbf{e}_k \\
\varepsilon \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{x}}{\partial \theta^2} &= \mathbf{\varepsilon u}_1 \times \mathbf{\hat{a}}_2 = \mathbf{\varepsilon u}_1 \times (\mathbf{\hat{a}}_2) = \varepsilon_{(u_1)_i} (\mathbf{\hat{a}}_2)_j \varepsilon_{ijk} \mathbf{e}_k = \varepsilon c_i b_j \varepsilon_{ijk} \mathbf{e}_k \\
\varepsilon \frac{\partial \mathbf{x}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} &= \mathbf{\varepsilon \hat{a}}_1 \times \mathbf{u}_2 = \mathbf{\varepsilon \hat{a}}_1 \times (\mathbf{u}_2) = \varepsilon_{(\mathbf{\hat{a}}_1)_i} (\mathbf{u}_2)_j \varepsilon_{ijk} \mathbf{e}_k = \varepsilon a_i d_j \varepsilon_{ijk} \mathbf{e}_k \\
\varepsilon^2 \frac{\partial \mathbf{u}}{\partial \theta^1} \times \frac{\partial \mathbf{u}}{\partial \theta^2} &= \varepsilon^2 \mathbf{u}_1 \times \mathbf{u}_2 = \varepsilon^2 (\mathbf{u}_1)_i (\mathbf{u}_2)_j \varepsilon_{ijk} \mathbf{e}_k = \varepsilon^2 c_i d_j \varepsilon_{ijk} \mathbf{e}_k
\end{align*}
\]  
(C.10)  
(C.11)  
(C.12)  
(C.13)

From eqn.C.7, \((\sqrt{a})^2 = \mathbf{A} \cdot \mathbf{A}\) where \(\mathbf{A}\) represents vector and \(\mathbf{A}\) can be recast into eqn.C.14 using eqn.C.10 through eqn.C.13

\[
\begin{align*}
\|\mathbf{A}\|^2 &= \mathbf{A} \cdot \mathbf{A} = \varepsilon_{ijk} \mathbf{e}_k (a_i b_j + \varepsilon c_i b_j + \varepsilon a_i d_j + \varepsilon^2 c_i d_j) \\
&= \varepsilon_{lmn} \mathbf{e}_n (a_i b_m + \varepsilon c_i b_m + \varepsilon a_i d_m + \varepsilon^2 c_i d_m) \\
&= \varepsilon_{ijk} \varepsilon_{lmn} \delta_{kn} (a_i b_j + \varepsilon c_i b_j + \varepsilon a_i d_j + \varepsilon^2 c_i d_j) (a_l b_m + \varepsilon c_l b_m + \varepsilon a_l d_m + \varepsilon^2 c_l d_m)
\end{align*}
\]  
(C.14)

152
Hence

$$\|A\|^2 = \frac{(\delta_{ii}\delta_{jm} - \delta_{im}\delta_{ji})(a_ib_j + e\epsilon c_i b_j + e\epsilon a_id_j + e^2c_id_j)(a_ib_m + e\epsilon c_i b_m + e\epsilon a_id_m + e^2c_id_m)}{B}$$ (C.15)

In other words, $\|A\| = \sqrt{B}$. Term $H$ in eqn.C.6 i.e. $\frac{\partial \sqrt{a}}{\partial \epsilon} |_{\epsilon=0}$ is what we are pursuing. Since $(\sqrt{a})^2 = A \cdot A = \|A\|^2 = B$, $\sqrt{a} = \sqrt{B}$

$$\frac{\partial \sqrt{a}}{\partial \epsilon} = \frac{\partial \sqrt{B}}{\partial \epsilon} =$$

$$\frac{1}{2} B^{-\frac{1}{2}} \frac{dB}{de} = \frac{1}{2} B^{-\frac{1}{2}} (\delta_{ii}\delta_{jm} - \delta_{im}\delta_{ji})[(c_ib_j + a_id_j + 2e\epsilon c_id_j)(a_ib_m + e\epsilon c_i b_m + e\epsilon a_id_m + e^2c_id_m) +$$

$$(a_ib_j + e\epsilon c_i b_j + e\epsilon a_id_j + e^2c_id_j)(c_ib_m + a_id_m + 2e\epsilon c_id_m)]$$ (C.16)
Now we evaluate eqn.C.16 at $\epsilon = 0$. We’ll tackle this term by term. First let’s look at term * in eqn.C.16. It becomes when $\epsilon = 0$ is set, referring to eqn.C.15:

\[
\frac{1}{2}[(\delta_{ij}\delta_{jm} - \delta_{im}\delta_{jl})a_ia_jb_ib_m]^{-\frac{1}{2}} = \frac{1}{2}[a_ia_1\delta_{ii}b_ib_mBm\delta_{jm} - a_1b_mb_m\delta_{im}a_1b_j\delta_{jl}]^{-\frac{1}{2}} \\
= \frac{1}{2}[(a \cdot a)(b \cdot b) - (a \cdot b)(a \cdot b)]^{-\frac{1}{2}} = \frac{1}{2}[(a \times b) \cdot (a \times b)]^{-\frac{1}{2}}.
\]

(C.17)

Hence eqn.C.17 tells us that term *

\[
\frac{1}{2}B^{\frac{1}{2}}|_{\epsilon=0} = \frac{1}{2} \frac{1}{\|a \times b\|}
\]

(C.18)

Now we can replace $a$ and $b$ by eqn.C.10

\[
\frac{1}{2}B^{\frac{1}{2}}|_{\epsilon=0} = \frac{1}{2} \frac{1}{\|a_1 \times a_2\|}
\]

(C.19)

---

1The relation used $(a \times b) \cdot (a \times b) = (a \cdot a)(b \cdot b) - (a \cdot b)(a \cdot b)$ is derived as follows,

\[
(a \times b) \cdot (a \times b) = a_ib_je_{ijk}a_jb_me_{ilm}e_k \cdot e_n \\
= a_ia_1b_1e_{ijk}a_1b_1e_{ilm}e_k \cdot e_n \\
= a_ia_1b_1b_mBm\delta_{jm} - a_1b_mb_mBm\delta_{jl} \\
= a_i\delta_{ij}\delta_{jm} - a_1b_mBm\delta_{jl} \\
= (a \cdot a)(b \cdot b) - (a \cdot b)(a \cdot b)
\]
Now we follow the notation on (Cirak et al., 2000, eqn.(7))

\[
\frac{1}{2} B^{-\frac{1}{2}} |_{\varepsilon=0} = \frac{1}{2} \frac{1}{\sqrt{\varepsilon}}
\]

(C.20)

The evaluation of the remaining terms of eqn.C.16 begins with first remove all terms having \( \varepsilon \) or \( \varepsilon^2 \) since we are evaluating at \( \varepsilon = 0 \), that is, \( \frac{\partial \sqrt{\varepsilon}}{\partial \varepsilon} |_{\varepsilon=0} = \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \left( \delta_{ij} \delta_{jm} - \delta_{im} \delta_{jl} \right) \left( (c_i b_j + a_i d_j) a_i b_m + a_i b_j (c_i b_m + a_i d_m) \right) \). When it is expanded, it becomes

\[
\frac{\partial \sqrt{\varepsilon}}{\partial \varepsilon} |_{\varepsilon=0} = \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \left[ \delta_{ij} \delta_{jm} (a_i b_m c_i b_j + a_i d_j a_i b_m) - \delta_{im} \delta_{jl} (a_i b_m c_i b_j + a_i d_j a_i b_m) \\
+ \delta_{ij} \delta_{jm} (a_i b_j c_i b_m + a_i b_j a_i d_m) - \delta_{im} \delta_{jl} (a_i b_j c_i b_m + a_i b_j a_i d_m) \right] \\
= \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \left[ c_i a_i \delta_{ij} b_j b_m \delta_{jm} + a_i a_i \delta_{ij} a_i b_m \delta_{jm} - c_i b_m \delta_{im} b_j a_i \delta_{jl} - a_i b_m \delta_{im} b_j a_i \delta_{jl} \right] \\
+ \frac{1}{2} \frac{1}{\sqrt{\varepsilon}} \left[ (c \cdot a) (b \cdot b) + (a \cdot a) (d \cdot b) - (c \cdot b) (b \cdot a) - (a \cdot b) (d \cdot a) \\
+ (a \cdot c) (b \cdot b) + (a \cdot a) (b \cdot d) - (a \cdot b) (b \cdot c) - (a \cdot d) (b \cdot a) \right] \\
= \frac{1}{\sqrt{\varepsilon}} \left[ (a \cdot c) (b \cdot b) + (a \cdot a) (b \cdot d) - (a \cdot b) (b \cdot c) - (a \cdot d) (b \cdot a) \right] \\
= \frac{1}{\sqrt{\varepsilon}} \left[ (c \cdot a) b \cdot b - (c \cdot b) a \cdot b + (d \cdot b) a \cdot a - (d \cdot a) b \cdot a \right] \\
= \frac{1}{\sqrt{\varepsilon}} \left[ (c \cdot a) b - (c \cdot b) a \right] \cdot b + \left[ (d \cdot b) a - (d \cdot a) b \right] \cdot a \\
= \frac{1}{\sqrt{\varepsilon}} \left[ c \times (b \times a) \cdot b + d \times (a \times b) \cdot a \right]
\]

Hence we have the compact form

155
$$\frac{\partial \sqrt{a}}{\partial \epsilon} \bigg|_{\epsilon=0} = \frac{1}{\sqrt{a}} \left[ c \times (b \times a) \cdot b + d \times (a \times b) \cdot a \right] \quad (C.21)$$

Now we replace $a$, $b$, $c$, and $d$ with eqn.C.10 through eqn.C.13.

$$\frac{\partial \sqrt{a}}{\partial \epsilon} \bigg|_{\epsilon=0} = \frac{1}{\sqrt{a}} \left[ -u_1 \times (\bar{a}_1 \times \bar{a}_2) \cdot \bar{a}_2 + u_2 \times (\bar{a}_1 \times \bar{a}_2) \cdot \bar{a}_2 \right] \quad (C.22)$$

$$= \frac{1}{\sqrt{a}} \left[ -(u_1 \times \bar{a}_3) \cdot \bar{a}_2 \sqrt{a} + (u_1 \times \bar{a}_3) \cdot \bar{a}_2 \sqrt{a} \right]$$

$$= \frac{1}{\sqrt{a}} \left[ -(\bar{a}_3 \times \bar{a}_2) \cdot u_1 \sqrt{a} + (\bar{a}_3 \times \bar{a}_1) \cdot u_2 \sqrt{a} \right]$$

$$= \frac{1}{\sqrt{a}} \left[ u_1 \cdot (\bar{a}_2 \times \bar{a}_3) + u_2 \cdot (\bar{a}_3 \times \bar{a}_1) \right] \sqrt{a}$$

$$= u_1 \cdot (\bar{a}_2 \times \bar{a}_3) + u_2 \cdot (\bar{a}_3 \times \bar{a}_1)$$

Eqn.C.22 shows that

$$\frac{\partial \sqrt{a}}{\partial \epsilon} \bigg|_{\epsilon=0} = u_1 \cdot (\bar{a}_2 \times \bar{a}_3) + u_2 \cdot (\bar{a}_3 \times \bar{a}_1) \quad (C.23)$$

and we can see that it's been totally linearized with $u$ being the only variable for the current state. Now substituting eqn.C.23 back into eqn.C.6 will lead to the following equation,

$$\beta_{\alpha \beta} = \frac{d}{d\epsilon} [\beta(\bar{x} + \epsilon u)]_{\epsilon=0} = -u_{\alpha \beta} \cdot \bar{a}_3 + \frac{1}{\sqrt{a}} \left[ u_1 \cdot (\bar{a}_{\alpha \beta} \times \bar{a}_2) + u_2 \cdot (\bar{a}_1 \times \bar{a}_{\alpha \beta}) \right] +$$

$$\frac{\bar{a}_3 \cdot \bar{a}_{\alpha \beta}}{\sqrt{a}} \left[ u_1 \cdot (\bar{a}_2 \times \bar{a}_3) + u_2 \cdot (\bar{a}_3 \times \bar{a}_1) \right] \quad (C.24)$$

156
To sum up, the above Newton linearization implementation linearizes the originally highly nonlinear bending strains $\beta_a = a_{a,\beta} \cdot \bar{a}_3 - a_{a,\beta} \cdot a_3$ down to a linear form as shown in eqn.C.24, which proves the equation appearing on (Cirak et al., 2000, eqn.22).
Appendix D

B-spline base functions

\[ N_1 = \frac{1}{12} \left[ (\theta^3)^4 + 2(\theta^3)^3(\theta^1) \right] \quad (D.1) \]
\[ N_2 = \frac{1}{12} \left[ (\theta^3)^4 + 2(\theta^3)^3(\theta^2) \right] \quad (D.2) \]
\[ N_3 = \frac{1}{12} \left[ (\theta^3)^4 + 2(\theta^3)^3(\theta^2) + 6(\theta^3)^3(\theta^1) + 6(\theta^3)^2(\theta^1)(\theta^2) + 12(\theta^3)^2(\theta^1)^2 + 6(\theta^3)(\theta^1)^2(\theta^2) + 6(\theta^3)(\theta^1)^3 + 2(\theta^1)^3(\theta^2) + (\theta^1)^4 \right] \quad (D.3) \]
\[ N_4 = \frac{1}{12} \left[ 6(\theta^3)^4 + 24(\theta^3)^3(\theta^2) + 24(\theta^3)^2(\theta^2)^2 + 8(\theta^3)(\theta^2)^3 + (\theta^2)^4 + 24(\theta^3)^3(\theta^1) + 60(\theta^3)^2(\theta^1)(\theta^2) + 36(\theta^3)(\theta^1)^2(\theta^2) + 6(\theta^1)(\theta^2)^3 + 24(\theta^3)^2(\theta^1)^2 + 36(\theta^3)(\theta^1)^3 + 12(\theta^1)^2(\theta^2)^2 + 8(\theta^3)(\theta^1)^3 + 6(\theta^1)^3(\theta^2) + (\theta^1)^4 \right] \quad (D.4) \]
\[ N_5 = \frac{1}{12} \left[ (\theta^3)^4 + 6(\theta^3)^3(\theta^2) + 12(\theta^3)^2(\theta^2)^2 + 6(\theta^3)(\theta^2)^3 + (\theta^2)^4 + 2(\theta^3)^3(\theta^1) + 6(\theta^3)^2(\theta^1)(\theta^2) + 6(\theta^3)(\theta^1)^2(\theta^2) + 2(\theta^1)(\theta^2)^3 \right] \quad (D.5) \]
\[ N_6 = \frac{1}{12} \left[ 2(\theta^3)(\theta^1)^3 + (\theta^1)^4 \right] \quad (D.6) \]
\[ N_7 = \frac{1}{12} \left[ (\theta^3)^4 + 6(\theta^3)^3(\theta^2) + 12(\theta^3)^2(\theta^2)^2 + 6(\theta^3)(\theta^2)^3 + (\theta^2)^4 + 8(\theta^3)^3(\theta^1) + 36(\theta^3)^2(\theta^1)(\theta^2) + 36(\theta^3)(\theta^1)^2(\theta^2) + 8(\theta^1)(\theta^2)^3 + 24(\theta^3)^2(\theta^1)^2 + 60(\theta^3)(\theta^1)^3 + 24(\theta^1)^2(\theta^2)^2 + 24(\theta^3)(\theta^1)^3 + 24(\theta^1)^3(\theta^2) + 6(\theta^1)^4 \right] \quad (D.7) \]
\[
N_8 = \frac{1}{12} \left[ (\theta^3)^4 + 8(\theta^3)^3(\theta^2) + 24(\theta^3)^2(\theta^2)^2 + 24(\theta^3)(\theta^2)^3 + 6(\theta^2)^4 + 6(\theta^3)^3(\theta^1) + 36(\theta^3)^2(\theta^1)(\theta^2) + 60(\theta^3)(\theta^1)(\theta^2)^2 + 24(\theta^1)(\theta^2)^3 + 12(\theta^3)^2(\theta^1)^2 + 36(\theta^3)(\theta^1)^2(\theta^2) + 24(\theta^1)^2(\theta^2)^2 + 6(\theta^3)(\theta^1)^3 + 8(\theta^1)^3(\theta^2) + (\theta^1)^4 \right] \quad (D.8)
\]
\[
N_9 = \frac{1}{12} \left[ 2(\theta^3)(\theta^2)^3 + (\theta^2)^4 \right] \quad (D.9)
\]
\[
N_{10} = \frac{1}{12} \left[ 2(\theta^1)^3(\theta^2) + (\theta^1)^4 \right] \quad (D.10)
\]
\[
N_{11} = \frac{1}{12} \left[ 2(\theta^3)(\theta^2)^3 + (\theta^2)^4 + 6(\theta^3)(\theta^1)(\theta^2)^2 + 6(\theta^1)(\theta^2)^3 + 6(\theta^3)(\theta^1)^2(\theta^2) + 12(\theta^1)^2(\theta^2)^2 + 2(\theta^3)(\theta^1)^3 + 6(\theta^1)^3(\theta^2) + (\theta^1)^4 \right] \quad (D.11)
\]
\[
N_{12} = \frac{1}{12} \left[ (\theta^2)^4 + 2(\theta^1)(\theta^2)^3 \right] \quad (D.12)
\]

In all the above base functions, \(\theta^1, \theta^2,\) and \(\theta^3\) meet \(\theta^1 + \theta^2 + \theta^3 = 1.\)
Appendix E

Derivation for Green strain tensor

Look at the deformation gradients $F$ and $F^T$ as two maps: $F : T\Omega_0 \rightarrow T\Omega$ and $F^T : T\Omega \rightarrow T\Omega_0$, see Fig.3.6. Let $W$ be a vector belonging to $T\Omega_0$ and $w$ belonging to $T\Omega$. Then the entity $FW \cdot w = F^T w \cdot W$ since they are referring to the length of the same material line. When expanded, we have

$$F^k_i W^l w^j g_{kl} = (F^T)^l_i w^j W^l G_{AB}$$

(E.1)

Since $W$ and $w$ are arbitrary, Eqn.E.1 reduces to

$$F^k_i g_{kl} = (F^T)^l_i G_{AB}$$

(E.2)

$F$ can be determined since there in only one unknown in Eqn.E.2 and it is

$(F^T)^l_k = g_{kl} G^{IH} F^l_H$, hence $C^l_I = G^{Ij} g_{kj}$. 

160
Bibliography


