#### Multi-Scale Modeling of Mechanical Properties of Single Wall Carbon Nanotube (SWCNT) Networks

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in

Materials Science and Engineering

#### Ankit Gupta

B.Tech., Materials Science and Engineering, Indian Institute of Technology Kanpur M.S., Materials Science and Engineering, Carnegie Mellon University

> Carnegie Mellon University Pittsburgh, PA

> > August, 2017

© Ankit Gupta, 2017 All Rights Reserved

#### Acknowledgments

I would like to thank my advisor Prof. Elizabeth Holm who gave me the opportunity to work on this interesting and challenging research topic. Her continuous support and guidance helped me a lot during this research and writing of this thesis. I would also like to thank my committee members, Prof. Anthony Rollett, Prof. Mohammad F. Islam, Prof. Jacobo Bielak and Prof. Kaushik Dayal for their insightful and constructive feedback that shaped this work. I also express my gratitude towards my group mates Brian, Jonathan, Philip, Ankita, Ian, Andrew and Anna for their valuable input and fruitful discussions. I also acknowledge Rundong Jiang who carried out a study included in this work. Finally I thank my parents, family and friends for their moral support and patience during all this time. This work is supported by NSF DMREF grant no 1335417 and through a Bradford and Diane Smith Fellowship in Engineering.

#### Abstract

## Multi-Scale Modeling of Mechanical Properties of Single Wall Carbon Nanotube (SWCNT) Networks

Single wall carbon nanotubes (SWCNTs) show a variety of unparalleled properties such as high electrical and thermal conductivity, high specific surface area (SSA) and a large stiffness under axial loads. One of the major challenges in tapping the vast potential of SWCNTs is to fabricate nanotube based macrostructures that retain the unique properties of nanotubes. Pristine SWCNT aerogels are highly porous, isotropic structures of nanotubes mediated via van der Waals (VDW) interactions at junctions. The mechanical behavior of such aerogels is examined in several experimental studies. However, it is necessary to supplement these studies with insights from simulations in order to develop a fundamental understanding of deformation behavior of SWCNT aerogels. In this study, the mechanical behavior of SWCNT networks is studied using a multi-scale modeling approach. The mechanics of an individual nanotube and interactions between few nanotubes are modeled using molecular dynamics (MD) simulations. The results from atomistic simulations are used to inform meso-scale and continuum scale finite element (FE) models. The deformation mechanism of pristine SWCNT networks under large compressive strain is deduced from insights offered by meso-scale simulations. It is found that the elasticity of such networks is governed by the bending deformation of nanotubes while the plastic deformation is governed by the VDW interactions between nanotubes. The stress response of the material in the elastic regime is dictated by the VDW stresses on nanotubes while in the plastic regime, both the VDW and axial deformation stresses on nanotubes drive the overall stress response. In this study, the elastic behavior of a random SWCNT network with any set of junction stiffness and network density is also investigated using FE simulations. It is found that the elastic deformation of such networks can be governed either by the deformation of the nanotubes (bending, axial compression) or deformation of the junctions. The junction stiffness and the network density determine the network deformation mode. The results of the FE study are also applicable to any stiff fiber network.

## Contents

$\mathbf{Li}$	st of	Figures	vii
$\mathbf{Li}$	st of	Tables	xi
1	Intr	oduction	1
	1.1	Research and Motivation	2
	1.2	Literature Review	6
		1.2.1 Network Structure	7
		1.2.2 Nature of Filaments	8
		1.2.3 Nature of Cross-links	9
		1.2.4 Small Deformation	9
		1.2.4.1 Flexible Filaments	9
		1.2.4.2 Semi-flexible Filaments	10
		1.2.5 Large Deformation	12
	1.3	Multi-Scale Modeling Approach	13
	1.4	Hypothesis	14
	1.5	Document organization	15
າ	мп	Simulations: CNT junctions	17
4	2 1	Introduction	18
	2.1 0.0	Method	10 20
	2.2 9.3		20
	2.5	Conclusion	20
	2.4		24
3	FEN	I: Method	25
	3.1	Network Geometry	26
		3.1.1 Placing and modifying nanotubes	26
		3.1.2 Generating junctions between nanotubes	27
		3.1.3 Generating beam elements	29
		3.1.4 Checking for network connectedness	30
	3.2	Finite Element Model	31
		3.2.1 Elements	32

	$3.3 \\ 3.4$	3.2.1.1 Beam Elements 32   3.2.1.2 Spring elements 35   3.2.2 Assembly of element matrices 37   3.2.3 Specifying boundary conditions 38   3.2.4 Solving the system of linear equations 46   3.2.5 Post-processing 42   3.2.5.1 Elastic modulus of the network 42   3.2.5.2 Elastic energy of the network 44   3.2.5.3 Non affinity measure 46   Affine theory 47 47   Conclusion 48 48
4	FEI	I: Results 49
	4.1	Introduction
	4.2	Network Geometry
	4.3	FEM: 2D analysis
		4.3.1 Variation of elastic modulus with unit cell size
		4.3.2 Parameter Map
		4.3.3 Non-affine deformation $\ldots \ldots \ldots$
	4.4	$FEM: 3D \text{ analysis} \dots \dots$
		4.4.1 Variation of elastic modulus with unit cell size
	15	4.4.2 Parameter Map
	4.5	Non linear mechanics: Junction fracture
	4.7	Conclusion
_		
5	Me	b-scale Simulations 81
	0.1	Introduction
		5.1.1 Dead Spring Model 5.1.2 Distinct Element Model 84
		5.1.2 Distinct Dictient Wodel 5.1.3 Meso-scopic Force Field Model 87
		5.1.4 Comparison between bead-spring, distinct element and meso-scopic
		force field model
	5.2	Method
		5.2.1 Axial Deformation of Nanotubes
		5.2.2 Bending Deformation of Nanotubes
		5.2.3 van der Waals Interactions between Nanotubes
	•	5.2.4 Implementation in LAMMPS
	5.3	$\begin{array}{cccccccccccccccccccccccccccccccccccc$
		5.3.1 Self-assembly of a UNT Network
		5.3.1.2 12-6 LJ potential (large $\sigma$ )

			5.3.1.3	10-4 LJ pot	ential			 • • •	 	 . 121
		5.3.2	Mechanic	al Behavior	of a CNT	' netwo	rk	 	 	 . 126
		5.3.3	Revisiting	g hypothesis				 	 	 . 137
	5.4	Conclus	sion					 •••	 	 . 140
6	Sun	nmary a	and Futu	ıre Work						142
	6.1	Summa	ry					 	 	 . 143
	6.2	Future	Work					 •••	 	 . 145
Bi	bliog	raphy								147
$\mathbf{A}$	$\operatorname{List}$	of imp	ortant v	ariables						156
в	LAN	AMPS	Input Se	$\operatorname{cript}$						160
$\mathbf{C}$	Elas	tic Mo	dulus C	urves						163
D	Elas	tic Ene	ergy Cur	ves						165
$\mathbf{E}$	Mod	lified so	ource co	de: Bendir	ıg buckli	ng				167
$\mathbf{F}$	Mod	lified so	ource co	de: LJ pot	ential					172

vi

# List of Figures

1.1 1.2	SWCNT aerogels: structure and property	3 5
1.3	Multi-scale Modeling Approach	14
2.1	Crossed nanotube configuration $\ldots \ldots \ldots$	19
2.2	Results of parallel nanotube configuration	21
2.3	Results of crossed nanotube configuration	21
2.4	Results of perpendicular nanotube configuration	22
3.1	Basic algorithm for generating a random network	26
3.2	Modifying nanotube and generating junctions	27
3.3	Example of an unconnected network	30
3.4	Random network of nanotubes	32
3.5	Two node elastic beam element	34
3.6	Different junction types in CNT network	37
3.7	Boundaries of the unit cell	40
3.8	Reaction forces at boundaries of the unit cell	43
3.9	Configuration of a single beam in affine theory	47
4.1	Variation of number of junctions per nanotube with 2D network density	51
4.2	Variation of number of junctions per nanotube in 3D network	52
4.3	Data collapse for number of junctions per nanotube in 3D network	52
4.4	Variation of network elastic modulus with system size in 2D network	54
4.5	Parameter map of variation of network elastic modulus in 2D network	54
4.6	Variation of elastic modulus of 2D network with junction axial stiffness	55
4.7	Variation of network elastic modulus with network density in 2D network .	56
4.8	Variation of elastic energy distribution with network density in 2D network	58
4.9	Variation of non affinity measures with network density in 2D network	59
4.10	Variation of non affinity measure with distance between nodes in 2D network	60
4.11	Elastic energy distribution in deformed 2D networks with stiff crosslinks	62
4.12	Elastic energy distribution in deformed 2D networks with compliant crosslinks	63
4.13	Variation of elastic modulus of 3D network with system size $L$	64

4.14	Variation of elastic modulus of 3D network with junction torsional stiffness	65
4.15	Parameter maps of elastic modulus and deformation mechanisms in 3D	
	network	66
4.16	Variation of elastic modulus of 3D network and deformation mechanisms	
	with junction axial stiffness in x direction	67
4.17	Variation of elastic modulus of 3D network with network density	68
4.18	Variation of elastic modulus of 3D network and deformation mechanisms	
	with junction axial stiffness in z direction	70
4.19	Variation of elastic modulus of 3D network with network density	72
4.20	Variation of elastic modulus of SWCNT network with density	74
4.21	Variation of elastic energy distribution in SWCNT network with density .	74
4.22	Distribution of the elastic energy in SWCNT network	75
4.23	Variation of non-affinity measures with network density in SWCNT network	75
4.24	Stress-strain curves of 3D network at different junction axial stiffness	76
4.25	% fractured junctions in 3D network v/s applied strain	77
4.26	Elastic energy distribution in 3D network v/s applied strain	77
4.27	Stress-strain curves of 3D network at different junction stiffness	77
4.28	% fractured junctions in 3D network v/s applied strain	77
4.29	Stress-strain curves of 3D network at different network densities	78
4.30	% fractured junctions in 3D network v/s applied strain	79
4.31	Number of junctions in 3D network v/s applied strain	79
5.1	Bead-Spring representation of a CNT	83
5.2	Corrugation in LJ potential	84
5.3	Distinct element representation of a CNT	85
5.4	Parallel bond model for elasticity of a CNT	85
5.5	Corrected LJ potential	86
5.6	MFF representation of a CNT	87
5.7	TEM image of semi-conducting CNT film	89
5.8	Meso-scale simulation techniques comparison	90
5.9	Stress-Strain curves for axial deformation of single CNT	92
5.10	Configuration of a nanotube during bending simulation	93
5.11	Results of atomistic simulations of bending of nanotubes	93
5.12	Buckling kink formed along the nanotube surface under compression	94
5.13	Results of bead spring model of bending of nanotube	95
5.14	VDW interaction energy between two nanotubes as a function of distance $d$	96
5.15	Configurations of two carbon nanotubes system	96
5.16	Results of sliding simulation of two nanotubes	97
5.17	Results of sliding simulation of two nanotubes	98
5.18	VDW interaction energy between two nanotubes as a function of distance d	99
5.19	Calculation of 10-4 LJ interaction between particles $i$ and $j$	100
5.20	Results of sliding simulation of two nanotubes	100

5.21	VDW interaction energy between two nanotubes as a function of distance a	101
5.22	VDW energy between two nanotubes as a function of distance $d$	103
5.23	Results of sliding simulation of two nanotubes	103
5.24	Configuration of three carbon nanotubes system	104
5.25	Snapshots of 3 CNTs configurations as obtained from atomistic simulation	105
5.26	Potential energy of 3 CNTs system as a function of angle	106
5.27	Snapshots of 3 CNTs configurations as obtained from bead spring simulations	\$107
5.28	Potential energy of 3 CNTs system as a function of time	108
5.29	Random network of 1 micron long nanotubes	111
5.30	Variation of potential energy of CNT network during self-assembly	112
5.31	Variation of components of potential energy of CNT network during self-	
	assembly	113
5.32	Final configuration of self assembled nanotube network	114
5.33	Bundling and entanglements present in CNT network	114
5.34	Micrographs of CNT aerogel	115
5.35	Distribution of VDW energy among nanotubes in CNT network	115
5.36	Variation of potential energy of CNT network during self-assembly	116
5.37	Variation of components of potential energy of CNT network during self-	
	assembly	117
5.38	Final configuration of self assembled nanotube network	118
5.39	Bundling and entanglements present in CNT network	118
5.40	Distribution of VDW energy among nanotubes in CNT network	119
5.41	Ordered mesh network of carbon nanotubes	120
5.42	Variation of potential energy of CNT network during self-assembly	120
5.43	Configuration of self assembled carbon nanotube network	121
5.44	Random network of 400 nm long nanotubes	122
5.45	Variation of potential energy of CNT network during self-assembly	123
5.46	Variation of components of potential energy of CNT network during self-	
	assembly	123
5.47	Final configuration of self assembled nanotube network	124
5.48	Bundling and entanglements present in CNT network	124
5.49	Distribution of VDW energy among nanotubes in CNT network	125
5.50	Stress-strain curve of CNT network obtained from 12-6 LJ potential	126
5.51	Stress-strain curve of a CNT aerogel	127
5.52	Variation of stress components of CNT network with strain	128
5.53	Stress-strain curve of CNT network obtained from 12-6 LJ potential with	
	large $\sigma$	128
5.54	Variation of potential energy with strain	129
5.55	Variation of potential energy components of CNT network with strain	130
5.56	Variation of stress components of CNT network with strain	131
5.57	Distribution of orientation angle of nanotubes with respect to the compres-	
	sive axis in CNT network as function of strain	132

5.58	Distribution of average and total axial stress over orientation angle in CNT
	network as function of strain
5.59	Distribution of average and total VDW stress over orientation angle in
	CNT network as function of strain
5.60	Deformation of a single nanotube at finite strains in CNT network 134
5.61	Top and front views of deformed CNT network
5.62	Distribution of bending stresses among nanotubes in CNT network 137
5.63	Governing deformation mechanisms of a CNT network

## List of Tables

2.1	Properties of SWCNT junctions governed by VDW interactions 23
3.1	List of values of nanotube parameters
$5.1 \\ 5.2$	LJ potentials parameters
B.1	Commands used in input script for LAMMPS

# Chapter 1

## Introduction

### **1.1** Research and Motivation

Carbon nanotubes are macromolecules of  $sp^2$  hybridized carbon with a high aspect ratio. Single wall carbon nanotubes (SWCNTs) show a variety of unparalleled properties. They have high electrical and thermal conductivity,[1] high specific surface area (SSA)[1] and a low density.[1] Due to having the highest known values of the spring constant between C-C bonds (500-1000 N/m),[2] SWCNTs provide very high stiffness to an axial load or small bending deformations, which manifests in high elastic moduli. At larger strains, nanotubes deform by buckling due to their hollow shell-like structure.[2] However, these deformations are reversible and involve no bond breaking. Nanotubes deform plastically at very high strains (several dozen percent depending on the strain rate) by forming defects that involve bond rotation.[2]

One of the major challenges in tapping the vast potential of nanotubes is to fabricate nanotube based macrostructures that retain the unique properties of nanotubes. To this date, these structures include vertically aligned CNT arrays via direct growth,[3] long fibers or sheets yarned from a CNT forest,[4] thin films or buckypapers,[5] and CNT based aerogels.[6] Aerogels in general are lightweight materials with a high SSA. The properties of aerogels depend upon the intrinsic properties of their constituent materials.

CNT based aerogels are highly porous, isotropic structures with random filamentous networks of nanotubes cross-linked via van der Waals interactions at junctions.[6] These interactions can be tailored in a number of ways such as covalent crosslinking, graphene coating or fused junctions as shown in Fig 1.1a.[7] CNT aerogels have an open cell structure with pore sizes ranging from 2-50 nm.[7]

A straightforward method to manufacture CNT aerogels is the single step CVD process. It consists of feeding a carbon source into a furnace operating at high temperature containing a catalyst precursor and quartz substrate in the presence of a carrier gas.[8] However this method produces aerogels consisting of a mixture of SWNCTs and multiwalled carbon nanotubes (MWCNTs). Since MWCNTs have much lower SSA compared to SWCNTs,[1] the SSA of aerogels produced using this method is comparatively low (300 to 400 m<sup>2</sup> gm<sup>-1</sup>).[8] In contrast, aerogels can also be created from an aqueous gel precursor by freeze drying and critical point drying (CPD).[7] In this process SWCNTs are



(Taken from Refs. [7] & [9])

Figure 1.1: a) Free standing aerogels with tunable junctions b) Stress-strain curve for uncoated aerogel c) Stress-strain curves for graphene coated and uncoated aerogels.

suspended in water using a surfactant. Suspensions are sonicated, centrifuged to sediment nanotube aggregates, and then poured into various shaped molds. These suspensions form hydrogels due to van der Waals interaction between nanotubes. These hydrogels are then converted to aerogels by removing background liquid using CPD or free drying without collapsing the network.[7] This method offers flexibility to tune aerogel shapes and sizes and control over CNT type, length, diameter, electronic properties and concentration.[6]

SWCNT aerogels fabricated from aqueous gel precursors have a low density (7.3-13.1 mg mL<sup>-1</sup>) and a high SSA (1291 m<sup>2</sup> gm<sup>-1</sup> at a density of 7.3 mg mL<sup>-1</sup>).[7] They form electrically percolating networks at very low volume fractions of nanotubes.[6] These aerogels have high electrical conductivity (250 S m<sup>-1</sup> at a density of 10 mg mL<sup>-1</sup>).[7] Figure 1.1b shows a stress-strain curve for SWCNT based aerogels under compression.[7] Aerogels can be compressed to a strain of > 90% because of their high porosity.[8, 7] This curve shows three characteristic regimes: a linear elastic regime (strain < 9%), a plateau regime (9% < strain < 60%) and a densification regime (strain > 60%). This curve has similar characteristics to that observed for open cell foams.[10] It is speculated that the initial linear elasticity may be caused by bending of nanotubes[10, 11, 7] or rotation of nanotubes about the nodes between nanotubes.[9] The plateau region can be associated with the collapse of cells due to elastic buckling of the nanotubes[10, 11, 7] and/or breaking/forming of nodes.[7] In the densification regime, when opposing cell walls touch each other, deformation is due to the compression of nanotubes, giving rise to a sharp increase in stress.[10] The measured elastic modulus for the linear elastic regime is higher than other aerogels at similar density.[7] However CNT aerogels undergo plastic deformation for strains > 9% or even in the linear elastic region after 2-3 cycles under cyclic loading, particularly at strains close to 9%.[7] The reason for the observed plastic deformation in the plateau regime is attributed to strain induced alignment of nanotubes because of increased van der Waals interaction between nanotubes on compression. And since there is no restorative force at original nodes, this alignment is irreversible.[9, 12, 7] The reason for plastic deformation under cyclic loading could be the development of selforganized folded patterns in nanotubes due to buckling after a few cycles.[11]

Aerogels can be made superelastic by coating junctions with a few layers of graphene nanoplates.[9] Figure 1.1 shows stress-strain behavior for uncoated and graphene coated aerogels during loading-unloading cycles under compression.[9] This curve also exhibits the same behavior as for open cell foams with three distinct regimes. However, graphene coated aerogels can recover their original shapes even when compressed to strains > 80%. The coating also increases the elastic modulus as well the compressive strength of the material.[9] Hysteresis loops in the loading unloading cycles can be attributed to energy dissipation due to friction between flowing air and nanotube walls.[8, 10, 9] There is no degradation in mechanical properties of coated aerogels after 2000 loading-unloading cycles at 60% strain and  $10^6$  cycles at 2% strain.[9] The superelastic behavior, increase in elastic modulus and compressive strength, and resistance to fatigue in graphene coated aerogels can be attributed to the strengthening of existing junctions that hinders the free rotation of nanotubes about the junctions and thus provides a restorative force large enough to destabilize additional junctions formed during large compression.[9] Figure 1.2 summarizes the different mechanisms of deformation proposed for CNT based aerogels.

Lightweight materials that are thermally and electrically conducting, highly elastic and resilient under cyclic loading can have a variety of applications. Aerogels produced



Figure 1.2: Plausible mechanisms of deformation in CNT aerogels.

from aqueous gel precursors by CPD and freeze drying can be back-filled with a polymeric fluid producing a polymeric nanocomposite. [6] Such nanocomposites will have an internal structure with well dispersed, homogeneously distributed nanotubes without the need of mechanical mixing.[13] Apart from this, other advantages are: low percolation threshold, large number density of fillers per filler volume, extensive interfacial area per volume of filler, short distances between fillers comparable to the radius of gyration of polymeric chain.[13] CNT based aerogels can also be used in energy storage devices in batteries, electrodes, and supercapacitors. In addition to high electrical conductivity and current density of individual nanotubes [1], a network of nanotubes such as one found in aerogels offers a combination of micropores (<2 nm) inherent to nanotubes [14] and mesopores (2-50 nm) formed by entanglement of CNTs.[7] This bimodal distribution of pore sizes is believed to be a property of an ideal electrode in electrochemical capacitors because actual energy storage occurs in micropores with large surface area while mesopores provide fast transport of electrolyte to and from the micropores. [14] CNT based aerogels are a promising candidates for environmental applications such as sorption. CNTs are considered as superior sorbents for gases because of their high specific surface area coupled with excellent van der Waals interaction, [1] and they can also be used for a wide range of organic and inorganic contaminants. CNT based aerogels in a densified state can absorb up to 180 times their own weight with excellent selectivity and recyclability.[8] Because CNT aerogels are hydrophobic and can be floated on water, they can be used in large-area spill cleanup applications.[8] CNT aerogel sheets are used in artificial muscles that provide giant elongations upon electrical actuation.[15] The structure and mechanical properties of cancellous bone is similar to a cellular material.[10] CNT based aerogels having very low density and superior mechanical properties can provide a good substitute for damaged or fractured bone.[10] These resilient and super-elastic aerogels can also be used in energy-absorbing and damping applications.[10, 11]

CNT aerogels have a vast potential in terms of applications. In order to realize this potential, a rigorous understanding of the mechanical behavior of CNT aerogels is required. The mechanical behavior of such aerogels is examined in several experimental studies. [7, 9] However, these studies generally measure the properties of the bulk due to the absence of an in-situ characterization technique that can offer insight into the deformation process unfolding at the nanoscale. Therefore, it is needed to adjunct these experimental studies with insights from simulations in order to develop a fundamental understanding of deformation of CNT aerogels. In some cases, the simulations can even guide the experiments towards rationally modifying the structure of aerogels so that their mechanical properties can be enhanced. But to date there is not a single numerical model that can provide a fundamental understanding of mechanics of CNT networks. This can be in part due to the complexity of deformation mechanisms involved. Most important among these are the VDW interactions between CNTs that differentiate CNT based aerogels from traditional cellular materials. Therefore, the goal of this study is to develop a numerical model that can capture the complex physics associated with the deformation of a CNT network and can also quantify its mechanical behavior as a function of network and junction properties.

## 1.2 Literature Review

It has been well established that the mechanics of any network depends broadly on the mechanical behavior of its individual filaments and cross-links and on network properties such as density of filaments, cross-links and the underlying network structure. [16, 17, 18, 19, 20, 21, 22, 23, 24, 25] It has also been suggested that networks might behave very differently under small and large deformations due to a number of reasons. For more than a half century, much of the work in the field of network mechanics was focused on materials such as papers and rubbers. But recently that trend has slightly shifted towards mechanics of cell, tissue and artificial biomaterials. Although the main focus of this review will be the class of materials in which carbon nanotubes fall, nevertheless we will briefly cover each of the important parameters that dictates mechanical behavior of networks.

#### **1.2.1** Network Structure

A network structure can be described based on certain parameters such as network density, distribution of filament segment lengths, and orientation. Network density can be characterized both by the density of cross-links as well as the density of filaments. The mean filament segment length  $l_c$  is related to filament density  $\rho_l$  by equation 1.1 for a 2D network.[26] Here,  $\rho_l$  is the total filament length per unit area and  $l_c$  is the average length between two crossings in the network.

$$l_c = \frac{\pi}{2\rho_l} \tag{1.1}$$

If we assign a cross-link at each filament crossing, then cross-link density  $\rho_n$  can be related to filament density  $\rho_l$  using equation 1.2.

$$\rho_n = \frac{\rho_l^2}{\pi} \tag{1.2}$$

Equations 1.3 and 1.4 give the corresponding relations in a 3D network.[27]

$$l_c = \frac{2}{\pi D \rho_l} \tag{1.3}$$

$$\rho_n = \frac{\pi D \rho_l^2}{4} \tag{1.4}$$

Segment lengths are Poisson distributed with a very broad spread [26] Another important result of some studies is that fluctuations in the filament density in a network depend upon the probing length scale. The fluctuations increase as length scale decreases. In one study, the authors found that the variation of the auto-correlation function of the density with probing length scale has a power law dependence with two distinct scaling regimes. [28] This outcome greatly affects the degree of homogeneity in the system in terms of strain distribution (affinity) as discussed later. Two other network parameters discussed in several papers are geometric and rigidity percolation. Geometric percolation is the density at which a continuous path spanning the whole network first appears. Rigidity percolation is when the network acquires finite stiffness. In networks with stiff cross-links, these two events are the same. Otherwise geometric and rigidity percolation occur at two separate filament densities,  $\rho_{\rm l} = 5.71/l$  and 6.71/l respectively in a 2D network. [29, 24] Here l is the length of the filament. Average filament orientation dictates the degree of anisotropy in the mechanical properties of the network. Preferential alignment of filaments in the network may introduce some anisotropy in the mechanical properties.[16]

#### **1.2.2** Nature of Filaments

The mechanical behavior of filaments can be categorized based on persistence length. For semi-flexible filaments (persistence length of the order of filament length), filament bending and stretching are major modes of deformation.[21] A further distinction can be made based upon the filament aspect ratio. For long slender beams, shear deformation can be neglected and Euler-Bernoulli theory can be used.[16, 30] For short, stubby beams, Timoshenko theory includes a correction due to the shear deformation mode.[16, 30] Given the broad distribution of segment lengths in a random network there might be some short segments.[26] Therefore, it is desirable to use Timoshenko beam theory for the entire network.[16] In the case of flexible filaments with relatively smaller persistence length, entropic stretching also becomes important.[21] The constitutive equations for such filaments can be represented by entropic spring models such as the freely rotating chain model[31] or worm like chain model.[32] If the filaments have small persistence length, they tend to coil up along their contour length. Upon applying strain at the ends, uncoiling of the filament occurs with a decrease in its entropy. This results in a thermodynamic force that depends both on the amount of stretch as well as the temperature of the system. SWCNTs generally have large aspect ratio and persistence length at room temperature.[2, 1] Therefore, both shear deformation and entropic stretching can be neglected.[2] SWCNT aerogels can thus be treated as semi-flexible networks.

#### **1.2.3** Nature of Cross-links

Two major distinct classes of cross-links are bonded and non-bonded. Bonded crosslinks can be further classified on the basis of whether they allow for the angle and distance between filaments to change. In some cases, cross-links may have some inherent structure and their own constitutive behavior. In cellular cytoskeletons for example, elasticity is governed by entropic stretching of actin cross-links that constitute a new class of crosslinks.[25, 33] In the case of non-bonded cross-links, filaments may interact via an excluded volume constraint or inter-fiber friction, and, cross-link density might change with deformation via formation/breaking of contacts.[16] In our study, we model van der Waals interactions between nanotubes as springs that resists both change in angle and distance between nanotubes in the small strain limit. In the case of large deformation, cross-links were allowed to break when strained beyond maximum length.

#### **1.2.4** Small Deformation

#### 1.2.4.1 Flexible Filaments

Early network deformation models were based on the assumption of an affine deformation of the network, i.e, filaments were assumed to deform based on a far field strain. Filaments were assumed to have all possible orientations and total stress is obtained by averaging over all orientations. It was found that the network modulus varies linearly with filament density. [34, 35, 36] Later in some models, it was taken into consideration that filaments may reduce their energy via non-affine deformation locally. [37, 38] In such networks, the modulus was found to depend on both network density as well as the coordination number at the cross-links. In [17], the authors considered all four mechanisms: axial stretching, bending, and entropic stretching. They found that from a purely affine perspective, increasing the average filament segment length transitions the overall mode of deformation from axial stretching to bending to thermal stretching. However, this prediction is valid only for regular cellular architecture. On increasing randomness in the network, the bending regime completely disappears. The behavior of the fibrous architecture was also affected by the nature of the cross-links. But the main conclusion of this study was that the networks with flexible filaments are more sensitive to randomness and polydispersity than the ones with semi-flexible filaments. They attributed this to the fact that entropic stretching stiffness varies as  $l^{-4}$  while mechanical stretching stiffness varies as  $l^{-1}$ .

#### 1.2.4.2 Semi-flexible Filaments

Several studies have been performed on deformation of networks with semi-flexible filaments with either freely rotating (pins) or stiff (welds) cross-links. The mechanical behavior of such networks is characterized by three distinct elastic regimes with different scaling between modulus with network density. The first regime is observed at a critical network density equivalent to either geometric or rigidity percolation depending upon the nature of the cross-links.[24, 29, 39, 21] Both the elastic modulus and the shear modulus scale as a power law of network density with an exponent of 3.0 [20, 21, 24] At the critical point, the moduli transition from a zero to a non-zero value continuously.[21, 24] Most of the elastic energy of the network is stored in bending of filaments in this regime.[20, 21]

On increasing the network density further, another distinct regime is observed. The system energy is still dominated by the bending mode, but a different scaling behavior is observed.[40, 41, 21, 24] The modulus of the network increases linearly with bending modulus of the filaments.[20, 21]The scaling exponent of the modulus with network density was found to be 6.67 in this regime.[24]

Upon further increasing the network density, the modulus of the network becomes independent of the bending modulus of the filaments and scales linearly with density.[20, 21]The elastic response of the network in this regime is governed by axial deformation of the filaments. [42, 43, 44, 45, 46, 47]

Another major distinction between bending and stretching dominated regimes is that deformation of the network in the former is highly non-affine. Non-affinity in the system decreases with an increase in network density. [16, 20, 21] However, it never vanishes, and the modulus of the network will always be smaller than the one predicted from the affine theory. [16, 22] Non-affinity in the system also depends upon the probing length scale. In the bending dominated regime, non-affinity increases without bound with decreasing length scale, while in the stretching dominated regime, it saturates with decreasing length scale.[21] The cross-over from the non-affine to affine regime is dictated by a dimensionless parameter  $\lambda$ .[20, 21] For  $\lambda \ll l$ , deformation is affine while for  $\lambda \gg l$  deformation is nonaffine. Here l is the length of the filament while  $\lambda$  is defined as  $l_{\rm c}(l_{\rm c}/l_{\rm b})^{\rm z}$ .  $l_{\rm c}$  as mentioned above is the average segment length, while  $l_{\rm b}$  is the square root of the ratio of bending to axial stiffness of the filament. The value of z in some numerical studies is calculated as 2/5 [20, 21] while it has been established as 1/3 in some empirical studies. [20] In another study the transition density,  $\rho_{\rm cross}$ , was found to vary as  $\rho_{\rm cross} * l \sim (r/l)^{-1/2.83}$ .[24] Here r is the radius of the filament. In [22], non affinity in the system is also measured as a function of length scale for different network density and bending stiffness. However, they found two scaling regimes for variation of non-affinity measure with probing length scale at all network densities. Distinction between the two regimes only vanished at very low filament bending stiffness. In another study [23], deformation of a three-dimensional network of semi-flexible filaments was considered. In this study, instead of varying the density of filaments, the density of cross-links was varied. They found two distinct regimes. At low cross-link density, they reported both shear and Young's modulus varied as density squared and the system's elastic energy was stored in bending mode. Upon increasing the cross-link density, they found that scaling exponent decreased from 2 to 1, and a transition from bending to stretching occurred. They measured non-affinity in the system via two measures, root mean square deviation (RMS) of simulated positions from those of perfectly affine deformations and a standard correlation function between the displaced and initial

positions. They found that RMS decreases and correlation increases on increasing the cross-link density, consistent with the fact that non-affinity in the system increases with density.

#### 1.2.5 Large Deformation

Intense strain hardening is observed in networks of both flexible and semi-flexible filaments under large deformation.[48] This might be due to the preferential alignment of filaments during deformation and also axial pulling of undulations in case of thermal filaments.[49, 50, 20, 24] In [19], two-dimensional networks of semi-flexible filaments with stiff cross-links were studied under shear. They observed a bending dominated regime at small strain followed by a transition into a stretching dominated regime characterized by a large network stiffness. This strain hardening was attributed to filament orientation along the direction of the strain. Undulations in the filaments simply delayed the transition from bending to stretching. On increasing the network density, the network became stiffer, and strain hardening was observed at a comparatively smaller value of strain. The degree of non affinity in the system was also found to decrease with strain as a result of filament re-orientation in the direction of strain.

Allowing the nodes to break led to strain softening and avalanches of cross-links failure. In [18], large deformation of three dimensional networks of stiff elastic filaments under tension was modeled. Three different types of cross-links were studied: In the first case, cross-links were permanent, in nature, unbreakable, and new ones can't be formed. The elastic modulus of the network in this case increased with strain and intense strain hardening was observed due to the orientation of filaments along the direction of applied strain, thereby increasing the contribution of axial stretching to the total elastic energy. This deformation was reversible. In the second case, cross-links were allowed to break when strained beyond maximum length. In this case, single cross-link failures nucleated small avalanches of cross-link failures, causing sharp drops in the stress-strain curve resulting in saw-tooth patterns. As the strain was increased, strong strain softening was observed. With increasing cross-link density, higher initial stiffness was observed but cross-links started breaking at smaller strains. Deformation of the network is irreversible, with stiffness vanishing when the networks were completely broken. In the third case, new cross-links were allowed to form if two filaments come closer than a certain distance. The number of cross-links in this case increased with the strain. Strain hardening was observed as in the case of permanent cross-links; however, in this case the deformation was irreversible, with elastic energy and number of cross-links further increasing during de-straining.

In [25, 51, 52, 53], the deformation of two and three-dimensional networks of rigid filaments with compliant cross-links was modeled under shear. These cross-links represented the highly compliant protein cross-links in cytoskeleton, [33] and were modeled as wormlike chains. Under small strains, the shear modulus was found to vary as the square of total cross-link density. In the non-linear regime observed at larger strains, the modulus of the network scaled with stress with a power law dependence of 3/2, and non-affinity increased with strain. Taking into account stretching as well as bending was found to have no effect on the non-linear stiffening regime.

In the case of non-bonded networks, hysteresis is observed between loading and unloading due to filament rearrangement and sliding.[54] In such networks, strain rate also plays an important role. Large strain can be achieved at same level of stress for slower strain rate. This is attributed to viscous effects at cross-links.[54] The elastic modulus of such networks under uniaxial compression seems to scale with density as a power law.[55]

## 1.3 Multi-Scale Modeling Approach

In this study, we model the mechanical behavior of CNT aerogels using a multi-scale modeling approach. Molecular dynamics (MD) simulations are used at the atomic scale to model the mechanical properties of individual nanotubes and to study the VDW interactions between a bunch of nanotubes. However these simulations become computationally expensive for a large system such as a network of nanotubes. Therefore the results from atomistic simulation are used to inform a continuum scale finite element model that is capable of performing faster simulation even for a large network of nanotubes. The finite element model is used to carry out high throughput simulations characterizing the elastic behavior of a CNT network in terms of the junction and network properties. But this



Figure 1.3: Multi-scale Modeling Approach

model fails at modeling VDW interactions between nanotubes in the limit of large strain. This model also cannot capture the entanglements between nanotubes present in a CNT aerogel. Therefore in order to bridge the gap between atomistic and continuum scale we employ a meso-scale model that can capture the complex physics associated with CNT aerogels while avoiding the large computation cost of traditional MD simulation. It is used to model self assembly of a CNT network and its large deformation behavior. Our multi scale modeling approach is summarized in Figure 1.3.

## 1.4 Hypothesis

1. In a stiff random fiber network, changing junction properties can change the elastic deformation mode.

2. A meso-scale simulation can model the large deformation behavior of a CNT network accurately including the three deformation regimes.

## **1.5** Document organization

In chapter 2, VDW interactions between CNTs are modeled using molecular dynamics (MD) simulations. The specific cases of translation, sliding and rotation of a pair of carbon nanotubes are studied. The resistance offered against deformation by the CNT junction is calculated. In addition, the fracture strength of these junctions is obtained.

In chapter 3, the method for generating a random network of nanotubes for 2D and 3D cases is discussed in the beginning. This section covers the algorithms for generating junctions between the nanotubes and checking for network connectedness in detail. This discussion is followed by the method for developing the finite element model in order to study the network deformation at the continuum scale. This section talks about the stiffness matrices of beam and spring elements, coordinate transformation, assembly of elements, loading boundary conditions, solving the system of linear equations, and post-processing operations. The affine theory for 2D network deformation is mentioned at the end.

In chapter 4, the results obtained from the finite element simulations of deformation of 2D and 3D networks are discussed. The first section of this chapter talks about the variation in the number of junctions with network density in 2D and 3D networks. In the next section, the FEM results for a 2D network are discussed, including characterizing the network deformation as a function of network structure and junction parameters. These results are also compared with the predictions from affine theory, and the amount of nonaffine deformation in the network is calculated. Similar parameter maps are developed for a 3D network, as well and a set of rules that dictates the network deformation mode. This chapter ends with a brief discussion about the non-linear mechanical behavior of CNT networks in the event of junction fracture.

In chapter 5, a meso-scale model is developed for studying the self-assembly and large deformation behavior of a pristine CNT network. In the first section of this chapter, the common meso-scale modeling approaches are discussed. In the second section, the formulation and implementation of a bead spring model for simulating a CNT network is discussed. This model is fitted using the results from atomistic MD simulations for individual nanotubes. In the last section, the results of the self assembly and mechanical testing of CNT networks are discussed. These results are also compared with the experimental results.

In chapter 6, the summary of this work including the important findings as well as the future scope of this work are discussed.

## Chapter 2

# **MD Simulations: CNT junctions**

## 2.1 Introduction

One of the major challenges in tapping the vast potential of CNTs is to fabricate nanotube-based macrostructures that capitalize on the unique properties of nanotubes. Interactions between nanotubes play a very important role in shaping mechanical properties of these macrostructures. Nanotubes interact via weak van der Waals (VDW) forces. The VDW potential between two atoms as a function of distance x can be modeled by the Lennard-Jones potential [56] as shown in equation 2.1.

$$u(x) = -\frac{A}{x^6} + \frac{B}{x^{12}} \tag{2.1}$$

Here A and B are attractive and repulsive constants respectively. This model assumes that electron density is uniformly distributed over the whole surface of a nanotube instead of being localized at each individual atom. [56] Using this approximation, the potential  $\Phi$ between two parallel nanotubes is given by equation 2.2. [56]

$$\phi = \nu^2 \int u(x)d\sum_1 d\sum_2 \tag{2.2}$$

where,  $\nu$  is the mean surface density of carbon atoms (equal to  $4/3(3)^{1/2}a^2$  where a is the lattice constant for graphene) and  $d\Sigma_1$  and  $d\Sigma_2$  are the surface elements for two nanotubes.

By inserting equation 2.1 into equation 2.2 and integrating, potential  $\Phi$  between two parallel nanotubes can be represented by equation 2.3.[57]

$$\phi = \nu^2 \frac{\pi^2 \sqrt{r_1 r_2}}{32 d^{7/2} \sqrt{r_1 + r_2}} \left( -5A + \frac{2431B}{2048d^6} \right)$$
(2.3)

where, d is the distance between the nanotubes.

The equilibrium distance  $d_o$  between two nanotubes is achieved when  $\delta \Phi / \delta d = 0$ .



Figure 2.1: Top and front view of crossed nanotube configuration of radii  $r_1$  and  $r_2$ , distance d and angle  $\gamma$  between them

$$d_o = \frac{1}{4} \left(\frac{92378}{35}\right)^{1/6} \left(\frac{B}{A}\right)^{1/6} \tag{2.4}$$

The energy well depth  $\Phi_o$  per unit length can be obtained by plugging the value of  $d_o$  into equation 2.3.[57]

$$\phi_o = -\nu^2 \frac{240\pi^2}{19} \left(\frac{35}{92378}\right)^{7/12} \frac{A^{19/12}}{B^{7/12}} \sqrt{\frac{2r_1 r_2}{r_1 + r_2}}$$
(2.5)

Figure 2.1 shows two nanotubes with radii  $r_1$  and  $r_2$  at a distance d and making an angle  $\gamma$  with respect to each other. The VDW potential  $\Phi$  between nanotubes in a crossed configuration is given by equation 2.6.[58, 57]

$$\phi = \frac{\nu^2 \pi^2 \sqrt{r_1 r_2}}{d^3 \sin\gamma} \left[ -A\left(\frac{1}{3} + \frac{\alpha d}{48}\right) + \frac{2B}{45d^6} \right]$$
(2.6)

Here, a is taken as  $1/r_1 + 1/r_2$ .

The VDW potential between two nanotubes also varies with angle  $\gamma$  such that the potential energy is minimum when the two tubes are in a parallel configuration, since this allows for maximum overlap of area.

### 2.2 Method

The nanotube configuration shown in figure 2.1 is modeled by molecular dynamics simulation using the LAMMPS package.[59, 60] The first step involves developing the atomic model. Carbon atoms belonging to different nanotubes are assigned as type 1 and 2. In order to represent the sp<sup>2</sup> bonding of C atoms in a nanotube, the AIREBO potential is applied to type 1-1 and 2-2 atomic bonds.[61] The Lennard-Jones (lj/cut) potential is applied to capture the VDW interactions between atoms of different type 1-2.[59, 60] Nanotube atomic coordinates are read from an input data file. Fixed boundary conditions are used in the simulation. The energy of the system is minimized prior to beginning the simulation using molecular statics.

In the first simulation, nanotubes are rotated in opposite direction at each time step by a small angle  $\gamma$ . The simulation is run for 50000 time-steps with each time-step being equal to 0.001 ps. The total energy of the system is plotted as a function of the angle between the nanotubes at every 200 time steps. In the second simulation, the angle between the nanotubes is taken to be zero (parallel nanotubes) and distance d between the nanotubes is varied to find the total energy as function of d. In the third simulation, the angle between nanotubes is taken to be ninety degrees (perpendicular nanotubes) and one nanotube is made to slide against the other to determine the total energy as function of distance moved by the first nanotube. The spring stiffness in different directions is then obtained by calculating the double derivatives of these plots as discussed below. The LAMMPS input script for the crossed nanotube case is shown in Appendix A.

### 2.3 Results

Figure 2.2a shows the variation of the energy of the parallel nanotube system ( $\gamma=0$ ) with respect to the distance between the nanotubes d. The equilibrium distance  $d_o$  is taken to be the distance at which the system energy is minimum. The value of  $d_o$  obtained from the simulation is 3.09 Å, quite close to the theoretical value of 3.17 Å, which is calculated by plugging in the values of constants A (15.2 eVxÅ<sup>6</sup>) and B (24100 eVxÅ<sup>12</sup>) in equation 2.4.[57] The well depth  $\phi_o = 1.73$  eV is the energy of the system at the equilibrium distance



Figure 2.2: Results of parallel nanotube configuration



Figure 2.3: Results of crossed nanotube configuration



Figure 2.4: Results of perpendicular nanotube configuration

 $d_o$ . Since this is the total energy, the energy per unit length is obtained by dividing by nanotube length, which is 24 Å in our case. Therefore the value of  $\phi_o$  obtained from the simulation is 72.15 meV/Å. The theoretical value of  $\phi_o$  is calculated by plugging in the values of constants  $\nu$  (0.381 Å<sup>-2</sup>),  $r_1$  (4 Å),  $r_2$  (4 Å), A (15.2 eVxÅ<sup>6</sup>) and B (24100 eVxÅ<sup>12</sup>) in equation 2.5.[57], giving  $\phi_o = 75.78 \text{ meV/Å}$ . The two values of  $\phi_o$  are in good agreement. These results validate our method for modeling VDW interactions between two nanotubes. The first derivative of the energy v/s distance curve gives the force between nanotubes, which is zero at the equilibrium separation, as shown in figure 2.2b. The second derivative of the energy curve gives the stiffness of a spring that resists translation in the z direction,  $k_z$ . At the equilibrium separation distance, this value is equal to 116.14 N/m.

Figure 2.3a shows the variation of energy with respect to angle  $\gamma$  between crossed nanotubes. It is clear that the system energy is minimum when  $\gamma$  is zero. This means that the parallel configuration of nanotubes is most stable, since VDW interactions are maximized. As shown in figure 2.3b, the second derivate of this plot at  $\gamma = 0$  gives the value of stiffness of a torsional spring that resists rotation along z direction,  $k_{rotz}$ . From our simulation, this value is equal to 4.59E-21 Nm/rad. Although  $k_{rotz}$  varies with  $\gamma$  as shown in figure 2.3b, in our network deformation simulations we will assume it to be

		C. I.C.	Maximum Force/	
Junction Type	Equilibrium Configuration	Stiffness	Torque	
resists change in spacing $d$	$d_o = 3.09 \text{\AA}$	116.14 N/m	$1.6 * 10^{-9}N$	
nociata change in angle .	- 09	$4.59 * 10^{-21}$	$2.20 \times 10^{-21} M_{}$	
resists change in angle $\gamma$	$\gamma_o = 0^{-1}$	Nm/rad	$3.20 * 10^{-1} Mm$	
resists sliding	perfect overlap of C atoms	$0.108 \ N/m$	$3.84 * 10^{-12}N$	

Table 2.1: Properties of SWCNT junctions governed by VDW interactions

constant.

Figure 2.4a shows the variation of energy during sliding of one nanotube against another in the perpendicular configuration. We will model sliding using springs that resist translation in x and y direction. These springs have equal stiffness constants in either direction. The stiffness constants,  $k_x \& k_y$ , are obtained by calculating the maximum in the second derivative of the energy curve shown in figure 2.4b. This value is equal to 0.108 N/m as obtained from the simulation. The force resulting due to compression or stretching of these springs is obtained by calculating the first derivative of the energy curve shown in figure 2.4b. The maximum value of this force,  $f_{max}$  (3.84E-12 N) is taken to be the fracture strength of the spring, beyond which the spring breaks or slip occurs in physical sense. In our finite element model, we will assume that complete fracture of the node occurs as soon as either spring breaks.

Table 2.1 summarizes the characteristics of the three junction types discussed above. As evident from the values of stiffness obtained from calculating the second derivates of the energy curves, nanotube junctions offer maximum resistance to any deformation that affects the spacing between them. The energy cost associated with the sliding and rotation of nanotubes at junctions is small, therefore such junction deformations have small stiffness. The energy fluctuations associated with the sliding of nanotubes arise from the periodic atomic structure of nanotubes and thus such fluctuations produce tiny forces as compared to the other two deformation types. Therefore, the junctions between nanotubes can easily break via sliding. Thus, the springs that model nanotube sliding at junctions in the finite element model have a fracture strength associated with them.

## 2.4 Conclusion

Molecular Dynamics simulations offer insight into van der Waals interactions of nanotubes at the atomistic scale, providing a measure of resistance offered against sliding, rotation and translation of nanotubes. These interactions will be modeled as linear translational / torsional springs added at each node in the network deformation model. However, the spring model is only valid in the limit of small strain. Therefore, in the event of large deformation, these springs will be allowed to break if strained beyond their maximum length. This length can be calculated based on the spring fracture strength obtained from the MD simulations.
# Chapter 3

# **FEM:** Method



Figure 3.1: Basic algorithm for generating a random network

# 3.1 Network Geometry

The method used to generate a random network of nanotubes described in this section is for a three dimensional network. The method for generating a two dimensional network is slightly different and much simpler. Major differences between the two methods are highlighted. Figure 3.1 outlines the basic algorithm for generating a random network. The major modules of the algorithm are covered in detail in the next few pages.

### 3.1.1 Placing and modifying nanotubes

A nanotube is represented by a line segment of length l. Line segments are placed sequentially in a unit cell of side length L until the desired network line density  $\rho_l$  is reached.  $\rho_l$  has the same meaning as defined in chapter 1 but in a 3D network it is the total nanotube length per unit volume of the unit cell. A line segment in three dimension can be represented by its mid-point and orientation. The orientation is described by azimuthal and polar angle,  $\vartheta_s$  and  $\varphi_s$ . In order to ensure a random distribution of nanotubes, midpoint co-ordinates are selected at random from [0, L]. To pick a random point on the



Figure 3.2: Modifying nanotube and generating junctions

surface of a unit sphere,  $\vartheta_s$  and  $\varphi_s$  are defined as  $2\pi u_r$  and  $\cos^{-1}(2v_r-1)$  respectively. Here  $u_r$  and  $v_r$  are random numbers chosen from [0, 1].[62] Once a line segment is placed its end-points and angles of orientation are stored. If a line segment extends outside the simulation box it is modified as shown in figure 3.2a. This is done in order to make the network periodic. In case of a 2D network, the unit cell is square instead of a cube and only one angle is required to define the orientation. This angle is selected from  $[0, \pi]$ .

#### 3.1.2 Generating junctions between nanotubes

A nanotube crossing in 3D is defined as the shortest distance between two line segments. A line segment is checked for such crossings against all previously deposited line segments. If the separation distance at a crossing is less than a certain cutoff distance, that crossing is assigned as a junction between nanotubes in the finite element model. The coordinates of the end-points of nanotube junctions are also stored. In the 2D network, a crossing is more simply defined as wherever two line segments intersect and all such crossings are assigned as nanotube junctions. The algorithm for finding the shortest distance between two line segments in 3D is described as follows.

Let end points of the two line segments shown in figure 3.2b be  $(x_{1a}, y_{1a}, z_{1a}), (x_{2a}, y_{2a}, z_{2a})$ 

and  $(x_{1b}, y_{1b}, z_{1b}), (x_{2b}, y_{2b}, z_{2b})$  respectively. Also let's assume that the shortest distance between two line segments is defined between points A on the first line segment and B on the second. The coordinates of points A and B then can be written as:

$$(x_A, y_A, z_A) = (x_{1a} + \lambda_1 (x_{2a} - x_{1a}), y_{1a} + \lambda_1 (y_{2a} - y_{1a}), z_{1a} + \lambda_1 (z_{2a} - z_{1a}))$$
(3.1)

$$(x_B, y_B, z_B) = (x_{1b} + \lambda_2(x_{2b} - x_{1b}), y_{1b} + \lambda_2(y_{2b} - y_{1b}), z_{1b} + \lambda_2(z_{2b} - z_{1b}))$$
(3.2)

Here  $\lambda_1$  and  $\lambda_2$  are two unknowns. So the final problem reduces to finding the values of parameters,  $\lambda_1$  and  $\lambda_2$  such that the distance between points A and B is minimum and both  $\lambda_1$  and  $\lambda_2 \in [0,1]$ . This problem is solved using the method of Lagrangian multipliers for inequality constraints as outlined below.

The distance between points A and B can be written as  $h(\lambda_1, \lambda_2)$ :

$$h(\lambda_{1},\lambda_{2}) = ((x_{A} - x_{B})^{2} + (y_{A} - y_{B})^{2} + (z_{A} - z_{B})^{2})^{1/2}$$
(3.3)

Let us define four parameters,  $\mu_1, \mu_2, \mu_3, \mu_4$ , such that the Lagrangian  $L(\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4)$  can be defined as:

$$L(\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4) = -h(\lambda_1, \lambda_2) + \mu_1(1 - \lambda_1) + \mu_2\lambda_1 + \mu_3(1 - \lambda_2) + \mu_4\lambda_2$$
(3.4)

So we need to solve the following 6 equations for 6 unknowns  $\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4$ :

$$\frac{\partial L(\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4)}{\partial \lambda} = 0 \tag{3.5}$$

$$\frac{\partial L(\lambda_1, \lambda_2, \mu_1, \mu_2, \mu_3, \mu_4)}{\partial \lambda_2} = 0 \tag{3.6}$$

$$\mu_1(1 - \lambda_1) = 0 \tag{3.7}$$

$$\mu_2 \lambda_1 = 0 \tag{3.8}$$

$$\mu_3(1 - \lambda_2) = 0 \tag{3.9}$$

$$\mu_4 \lambda_2 = 0 \tag{3.10}$$

Note that there will be 9 sets of solution in total. But only the solution that satisfies the following inequalities will be valid:

$$0 \le \lambda_1 \le 1 \tag{3.11}$$

$$0 \le \lambda_2 \le 1 \tag{3.12}$$

$$\mu_1, \mu_2, \mu_3, \mu_4 \ge 0 \tag{3.13}$$

Once the appropriate values of  $\lambda_1$  and  $\lambda_2$  are known, they can be used to calculate the shortest distance, |AB|, between the two line segments.

#### 3.1.3 Generating beam elements

Nanotube segments between two nanotube junctions, between a nanotube junction and a boundary, or between two boundaries are assigned as beam elements. In order to do this, nanotube junctions need to be sorted according to their order along each nanotube. This can be easily done since junction coordinates are represented as  $p_1 + \lambda(p_2 - p_1)$ . Here  $p_1$  and  $p_2$  are the coordinates of nanotube endpoints and  $\lambda$  dictates the position of a junction in the nanotube reference frame. So nanotube junctions can be sorted by their corresponding value of  $\lambda$ . It is also necessary to check that no two junctions along a nanotube have same coordinates, since it will result in a beam element of length zero. Nanotube segments with an end that doesn't lie on a boundary or nanotube junction (i.e dangling segments) carry no stress during network deformation. In a 2D network, such segments are not assigned as beam elements to save computation cost. However in a 3D network, it is more expensive to search for such elements than to include them in the



Figure 3.3: Example of an unconnected network

finite element model. Therefore, those nanotube segments are made into beam elements as well.

#### **3.1.4** Checking for network connectedness

Before a network is analyzed in the finite element model, it is important to ensure that the network is connected and there are no isolated clusters of nanotubes, because this results in a singular stiffness matrix. A connected network is connected with itself across all boundaries, and a continuous path exists between any two nodes in the network. The connected with itself part is very important. A network might not be a connected one even if there exists a continuous path between two opposite cell boundaries as shown in figure 3.3. [63]

In order to analyze a network for connectedness, it is stored as a graph with each junction stored as a node and each element stored as an edge between two nodes. Graph theory then can be used to check for network connectedness. There are several efficient algorithms that are specifically designed for this purpose. Efficiency can be defined both in terms of memory usage and number of operations performed. In a 3D network however, memory usage becomes more critical especially for dense networks. A 3D network is stored in the form of an adjacency list since its space usage is  $\Theta(n + m)$ , while a 2D network is stored in form of an adjacency matrix since it is much easier to manipulate. Here, n

and m are the number of nodes and edges in a graph respectively. A depth first search (DFS) algorithm is used to find all possible connected clusters in a network: A node v is selected. The function DFS(v) traverses the graph to find all nodes u connected to node v in a depth-first order. v is marked as visited. For each node u connected to v, if not visited previously, the function DFS(u) is called. This is the recursive version of the DFS algorithm that is implemented for a 2D network. For a 3D network, the recursive calls are replaced with a stack.

Once all possible connected clusters in a network are identified, each cluster is checked that it connects with itself across all boundaries. In a 2D network, there can be only one connected cluster that satisfies this criterion, while in a 3D network there can be more than one.[63] Each cluster that satisfies the above criterion is completed with any parts that it is connected to across boundaries but not within the unit cell. The rest of the connected clusters are isolated and are removed from the network. Figures 3.4a and 3.4b show a random network of nanotubes for a 2D and 3D case respectively.

There can be some clusters in a network that are connected to the rest of the network through a single nanotube. These dangling clusters will not participate in deformation, but they occur very rarely and are difficult to find.[63] Therefore, such clusters aren't removed from the network.

# 3.2 Finite Element Model

The finite element method consists of dividing the network into beam and connection elements, writing the stiffness matrix for each element, assembling element stiffness matrices into the global stiffness matrix, specifying load and displacement boundary conditions, and finally solving the system of linear equations.[64] Each of these steps are explained in detail in the following sections.



(a) 2D network

(b) 3D network

Figure 3.4: Random network of nanotubes

## 3.2.1 Elements

#### 3.2.1.1 Beam Elements

Carbon nanotubes can be treated as effective elastic beams for small strain behavior before buckling initiates.[65] In the FEM, a hollow elastic cylinder with a wall thickness  $t_{tube}$  of 0.34 nm represents a SWCNT.[65, 66] As mentioned previously, SWCNTs have a very high aspect ratio.[2, 1] Therefore the effect of shear deformation can be neglected for carbon nanotubes of length l. However, in the actual network of carbon nanotubes, there is a broad distribution of nanotube segment lengths.[16] Some of the segments might be short enough for Euler-Bernoulli theory to become inaccurate. For this reason, Timoshenko beam theory is used for modeling nanotubes in a 3D network, although in 2D networks, Euler-Bernoulli theory is used.

In case of a 3D network, a carbon nanotube is modeled as a two-node elastic beam element with six degrees of freedom at each node as shown in figure 3.5. An element stiffness matrix relates the nodal displacements and rotations to nodal forces and torques as represented by the equation f = Ku. In this equation, f and u are force and displacement vectors given by equation 3.14, and K is the stiffness matrix.

$$\begin{aligned}
f_{x1} & u_{1} \\
f_{y1} & v_{1} \\
f_{z1} & w_{1} \\
\tau_{x1} & \theta_{x1} \\
\tau_{y1} & \theta_{y1} \\
f = \frac{\tau_{z1}}{f_{x2}} & u = \frac{\theta_{z1}}{u_{2}} \\
f_{y2} & v_{2} \\
f_{z2} & w_{2} \\
\tau_{x2} & \theta_{x2} \\
\tau_{y2} & \theta_{y2} \\
\tau_{z2} & \theta_{z2}
\end{aligned}$$
(3.14)

The stiffness matrix  $K_{\text{beam}}$  for a beam element is given by equation 3.15.[64]

where,  $X = \frac{A_{tube}E_{tube}}{l_S}, Y_1 = \frac{12E_{tube}I_{tube}}{(1+\phi_{tube})l_s^3}, Y_2 = \frac{6E_{tube}I_{tube}}{(1+\phi_{tube})l_s^2}, Y_3 = \frac{(4+\phi_{tube})E_{tube}I_{tube}}{(1+\phi_{tube})l_s},$ 



Figure 3.5: Two node elastic beam element (taken from [64])

$$Y_{4} = \frac{(2 - \phi_{tube})E_{tube}I_{tube}}{(1 + \phi_{tube})l_{s}}, \ S = \frac{G_{tube}J_{tube}}{l_{s}}, \ \phi_{tube} = \frac{24E_{tube}I_{tube}}{A_{tube}G_{tube}l_{s}^{2}},$$
  
and  $Z_{1} = Y_{1}, \ Z_{2} = Y_{2}, \ Z_{3} = Y_{3}, \ Z_{4} = Y_{4}$ 

 $K_{\text{beam}}$  is a symmetric matrix characterized by the area of cross-section  $A_{tube}$ , elastic modulus  $E_{\text{tube}}$ , second moment of inertia  $I_{tube}$ , shear modulus  $G_{\text{tube}}$ , polar moment of inertia  $J_{\text{tube}}$  and the segment length between two nanotube junctions,  $l_s$ . Table 3.1 provides a list of values of these parameters used in the finite element model.

The stiffness matrix given by equation 3.15 is for a beam element lying along the global x axis, but in a network of nanotubes, they can realize any orientation. Therefore the stiffness matrix must be transformed from local to global coordinates. This transformation is done according to the equation  $K_{el} = T^t KT$ . Here T is the transformation matrix defined in the equation 3.16.

$$T_{12x12} = \begin{pmatrix} \Lambda & 0 & 0 & 0 \\ 0 & \Lambda & 0 & 0 \\ 0 & 0 & \Lambda & 0 \\ 0 & 0 & 0 & \Lambda \end{pmatrix} \qquad where \qquad \Lambda = \begin{pmatrix} l_1 & m_1 & n_1 \\ l_2 & m_2 & n_2 \\ l_3 & m_3 & n_3 \end{pmatrix}$$
(3.16)

 $l_i$ ,  $m_i$ , and  $n_i$  are the direction cosines of the local  $x_i$  axis with respect to the global x, y and z axes respectively as given by equation 3.17.

For a beam element,

$$\Lambda = \begin{pmatrix} \frac{(x_2 - x_1)}{l_s} & \frac{(y_2 - y_1)}{l_s} & \frac{(z_2 - z_1)}{l_s} \\ \frac{-(y_2 - y_1)}{e_s} & \frac{(x_2 - x_1)}{e_s} & 0 \\ \frac{-(z_2 - z_1)(x_2 - x_1)}{f_s} & \frac{-(z_2 - z_1)(y_2 - y_1)}{f_s} & \frac{(x_2 - x_1)^2 + (y_2 - y_1)^2}{f_s} \end{pmatrix}$$
(3.17)  
where,  $l_s = ((x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2)^{1/2}$ ,  
 $e_s = ((x_2 - x_1)^2 + (y_2 - y_1)^2)^{1/2}$ ,  
 $f_s = (((z_2 - z_1)(x_2 - x_1))^2 + ((z_2 - z_1)(y_2 - y_1))^2 + ((x_2 - x_1)^2 + (y_2 - y_1)^2)^2)^{1/2}$ 

and  $(x_1, y_1, z_1)$ ,  $(x_2, y_2, z_2)$  are the coordinates of the end-points of a nanotubes.

The local x axis for a beam element is defined along its length vector. Since the cross-section of the nanotube is circular, any mutually perpendicular set of axes that are also perpendicular to the local x axis can be taken as the local y and z axes. In this simulation, the local y axis is taken to lie in the global x-y plane for a beam element.

For a 2D network, the beam element is a two node element with three degrees of freedom at each node (translation along x, y and rotation along the z axis), so the stiffness matrix is a 6x6 symmetric matrix. Also the transformation matrix is much simpler in that case.

#### 3.2.1.2 Spring elements

As mentioned earlier, nanotube junctions are represented by spring elements that resist translation and rotation in all directions. The stiffness matrix  $K_{junction}$  of such an element is given by equation 3.18.

Parameter	Value
Diameter $d_{tube}$	1 nm
Wall thickness $t_{tube}$	0.34  nm
Length of nanotube $l$	1 µm
Elastic modulus $E_{tube}$	1.25 TPa
Shear Modulus $G_{tube}$	0.5 TPa
Area of cross-section $A_{tube}$	$\pi \ d_{tube} t_{tube} = 1.07 \ nm^2$
Second moment of Inertia $I_{tube}$	$\pi \left( (d_{tube} + t_{tube})^4 - (d_{tube} + t_{tube})^4 \right) / 64 = 0.15 \ nm^4$
Polar moment of Inertia $J_{tube}$	$2I_{tube} = 0.29 \ nm^4$

Table 3.1: List of values of nanotube parameters

$$K_{junction} = \begin{pmatrix} k_x & 0 & 0 & 0 & 0 & 0 & -k_x & 0 & 0 & 0 & 0 & 0 \\ k_y & 0 & 0 & 0 & 0 & 0 & -k_y & 0 & 0 & 0 & 0 \\ k_z & 0 & 0 & 0 & 0 & 0 & -k_{rotx} & 0 & 0 \\ k_{rotx} & 0 & 0 & 0 & 0 & 0 & -k_{roty} & 0 \\ k_{roty} & 0 & 0 & 0 & 0 & 0 & -k_{roty} & 0 \\ k_{rotz} & 0 & 0 & 0 & 0 & 0 & -k_{rotz} \\ k_x & 0 & 0 & 0 & 0 & 0 \\ k_y & 0 & 0 & 0 & 0 \\ k_{rotx} & 0 & 0 & 0 \\ k_{rotx} & 0 & 0 & 0 \\ k_{roty} & 0 & k_{roty} & 0 \\ k_{rotz} \end{pmatrix}$$
(3.18)

This equation is characterized by translational  $(k_x, k_y, k_z)$  and rotational stiffness  $(k_{rotx}, k_{roty}, k_{rotz})$  along the x, y and z directions. In this simulation, we have assumed  $k_{rotx}$  and  $k_{roty}$  to be equal to  $k_{rotz}$ . The different types of junctions are shown in figure 3.6. To represent the case of VDW interactions at nanotube junctions, values of the stiffness constants are obtained from the MD simulation results mentioned in chapter 2. The local



Figure 3.6: Different junction types in CNT network

stiffness matrix is transformed according to the equation  $K_{global} = T^t K T$ . However, in this case, the local z direction is defined along the length vector of the spring element. The direction cosines for this element are given via equation 3.19.

$$\Lambda = \begin{pmatrix} \frac{(z_2 - z_1)(x_2 - x_1)}{f_s} & \frac{(z_2 - z_1)(y_2 - y_1)}{f_s} & \frac{-(x_2 - x_1)^2 + (y_2 - y_1)^2}{f_s} \\ \frac{-(y_2 - y_1)}{e_s} & \frac{(x_2 - x_1)}{e_s} & 0 \\ \frac{(x_2 - x_1)}{l_s} & \frac{(y_2 - y_1)}{l_s} & \frac{(z_2 - z_1)}{l_s} \end{pmatrix}$$
(3.19)

Here,  $l_s$ ,  $e_s$ ,  $f_s$ ,  $(x_1, y_1, z_1)$  and  $(x_2, y_2, z_2)$  are defined similarly as for the beam elements.

Again for a 2D network, the stiffness matrix is a 6x6 matrix with only  $k_x$ ,  $k_y$ ,  $k_{rotz}$  terms, and it is not transformed.

## 3.2.2 Assembly of element matrices

The element stiffness matrices need to be assembled to form the global stiffness matrix. The global stiffness matrix  $K_{global}$  relates degrees of freedom (DOFs) to forces and torques at each node in the network according to the equation  $f_{global}=K_{global}u_{global}$ . In a 2D network,  $K_{global}$  is stored directly in form of a matrix, since it is easier to modify such matrix for future operations. However the space usage is  $\Theta(n^2)$ . This becomes very costly in case of a 3D network, especially at high density. Since most of the elements in the stiffness matrix are zero (sparse matrix), it makes more sense to store only non-zero elements. Therefore, in a 3D network, the global stiffness matrix is stored in sparse triplet matrix format. In this format, three matrices each with memory usage of  $\Theta(n)$  are used. The first two matrices are used to store the row and column of each non-zero entry in the matrix. The third matrix stores the non-zero entry corresponding to that position. Element stiffness matrices are directly assembled in this format for a 3D case. The only drawback of sparse triplet matrix format is that it is not trivial to manipulate a specific entry in the matrix.

### 3.2.3 Specifying boundary conditions

The end goal of this simulation is to calculate response of the nanotube network subjected to deformation. A deformation may correspond to various modes of strain  $\epsilon = (\epsilon_x, \epsilon_y, \epsilon_x, \gamma_{xy}, \gamma_{xz}, \gamma_{yz})$ . Due to computational limitations, the length scale of the material to be simulated can't be matched with the experiments. The unit cell, which is periodic, can be treated as one of many identical cells making up a larger network in the undeformed state. Therefore strain should be applied in such a way that the cell boundaries should match in the deformed state as well fulfilling the continuity requirement. This is achieved by subjecting the network to a set of cyclic displacement boundary conditions given by equation 3.20.[63]

$$\begin{pmatrix} u^{s2} - u^{s1} \\ v^{s2} - v^{s1} \\ w^{s2} - w^{s1} \\ w^{s2} - w^{s1} \\ \theta^{s2}_{x} - \theta^{s1}_{x} \\ \theta^{s2}_{y} - \theta^{s1}_{y} \\ \theta^{s2}_{z} - \theta^{s1}_{z} \\ u^{s4} - u^{s3} \\ v^{s4} - v^{s3} \\ \theta^{s4}_{x} - \theta^{s3}_{x} \\ \theta^{s4}_{y} - \theta^{s3}_{y} \\ \theta^{s4}_{z} - \theta^{s3}_{z} \\ \theta^{s4}_{z} - \theta^{s3}_{z} \\ u^{s6} - u^{s5} \\ v^{s6} - v^{s5} \\ \theta^{s6}_{y} - \theta^{s5}_{y} \\ \theta^{s6}_{z} - \theta^{s5}_{z} \\ \theta^{s6}_{z} - \theta^{s6}_{z} \\ \theta^{s6}_{z} \\ \theta^{s6}_{z} - \theta^{s6}_{z} \\ \theta^{s6}_{z} \\$$

The DOFs correspond to a particular node on the boundary that is specified by the superscript. Since the unit cell is periodic, for every boundary node there exists a node at exactly the same location on the opposite boundary. Therefore, the DOFs of the two nodes on opposite boundaries can be constrained by a relationship. Boundaries s1 to s6 of the unit cell are shown in figure 3.7. Since, the networks in this simulation are only subjected to compressive strain in the y direction,  $\epsilon_y$ , boundary constraints shown in equation 3.20 reduce to  $v^{s4} - v^{s3} = L\epsilon_y$ . The rest of the DOFs of the opposing boundary nodes are set to be equal to each other. In case of a 2D network, the constraint relations remain the same. Only the number of DOFs reduces to 2 and the cell boundaries reduce from 6 to 4.



Figure 3.7: Boundaries of the unit cell

## 3.2.4 Solving the system of linear equations

The stiffness matrix obtained by assembling the element matrices is singular in nature, since it has not been modified to accommodate the displacement constraints mentioned in equation 3.20. In order to enforce these constraints, the constraint elimination method is used.[67, 68, 69, 70, 71] The displacement of node i at the boundaries x, y, z=L can be written in terms of the displacement of nodes at boundaries x, y, z=0 respectively as shown in equation 3.21. Here g can be calculated from equation 3.20.

$$u_L^i = u_0^i + g, \qquad i = 1, ..., p$$
 (3.21)

The original system of equation is:

$$K_{global}u_{global} = f_{global} \tag{3.22}$$

The displacement vector  $u_{global}$  can be divided into three parts: displacements of internal nodes  $u_I$ , displacements of nodes at boundaries  $x, y, z=L, u_L$  and displacements of nodes at boundaries  $x, y, z=0, u_0$ :

$$u_{global} = \begin{pmatrix} u_I \\ u_L \\ u_0 \end{pmatrix}$$
(3.23)

Accordingly equation 3.22, can be reorganized as:

$$\begin{pmatrix} K_{II} & K_{IL} & K_{I0} \\ K_{LI} & K_{LL} & K_{L0} \\ K_{0I} & K_{0L} & K_{00} \end{pmatrix} \begin{pmatrix} u_I \\ u_L \\ u_0 \end{pmatrix} = \begin{pmatrix} f_I \\ f_L \\ f_0 \end{pmatrix}$$
(3.24)

On inserting equation 3.21 in equation 3.24, results in:

$$\begin{pmatrix} K_{II} & K_{IL} + K_{I0} \\ K_{LI} + K_{0I} & K_{LL} + K_{L0} + K_{0L} + K_{00} \end{pmatrix} \begin{pmatrix} u_I \\ u_0 \end{pmatrix} = \begin{pmatrix} f_I - (K_{IL})g \\ f_L + f_0 - (K_{LL} + K_{0L})g \end{pmatrix}$$
(3.25)

Since no external force is applied at the interior nodes of the network, the  $f_I$  term in equation 3.25 is  $\theta$ . There will be reaction forces at boundaries in the network,  $f_L$  and  $f_{\theta}$ . But since the displacement boundary conditions are periodic in nature, the reaction forces at opposite boundaries will be in opposite direction, equal in magnitude and therefore will cancel each other out. Thus,  $f_L + f_{\theta}$  will also be  $\theta$ . This equation can be re-written as:

$$\begin{pmatrix} K_{II} & K_{IL} + K_{I0} \\ K_{LI} + K_{0I} & K_{LL} + K_{L0} + K_{0L} + K_{00} \end{pmatrix} \begin{pmatrix} u_I \\ u_0 \end{pmatrix} = \begin{pmatrix} -(K_{IL})g \\ -(K_{LL} + K_{0L})g \end{pmatrix}$$
(3.26)

or

$$\widetilde{K_{global}}\widetilde{u_{global}} = \widetilde{f_{global}}$$
(3.27)

By using this method, the number of unknowns in the system can be reduced. Equation 3.27 can be solved for the reduced displacement vector  $\widetilde{u_{global}}$  consisting of  $u_I$  and  $u_0$ . Once values of  $u_0$  are known, equation 3.21 can be used to calculate  $u_L$ .

In addition to prescribing displacement constraints at boundary nodes, the translational displacements of an interior node should also be set to zero to prevent rigid body motion.[63] The global stiffness matrix and force vector need to be modified accordingly. This is done using the penalty method.[72] In this method, the stiffness term corresponding to the DOF to be prescribed is modified by adding a large penalty term to it. For NDOFs, the *i*<sup>th</sup> DOF equation may look like:

$$K_{i1}u_1 + \dots + K_{ii}u_i + \dots + K_{iN}u_N = f_i \tag{3.28}$$

Now, in order to set  $u_i$  to some known value D, equation 3.28 is modified as follow:

$$K_{i1}u_1 + ... + (K_{ii} + p)u_i + ... + K_{iN}u_N = f_i + pD$$
(3.29)

Here, p is a large penalty term. Using the implied equilibrium relation given by equation 3.28, equation 3.29 can be simplified to:

$$pu_i = pD \tag{3.30}$$

In this simulation, the penalty term p is chosen as:

$$p = \max \left( |K_{i,j}| \right) x 10^{12}$$

$$(3.31)$$

The reduced system of linear equations 3.27 can now be solved. In this simulation, a direct sparse linear solver, CSPARSE, developed by Timothy Davis, is used for this purpose.[73] For this solver, the stiffness matrix should be input in compressed column format. A C library known as ST\_TO\_CC is used to convert the stiffness matrix from sparse triplet to compressed column file format.[74]

## 3.2.5 Post-processing

The post-processing operations involve calculating the elastic modulus of the network, contribution of various deformation modes to the total elastic modulus of the network and non-affinity measures. The method underlying each of these calculations is outlined as follows.

#### 3.2.5.1 Elastic modulus of the network

The elastic modulus of the network can be calculated in two ways. The first method involves determining the reaction forces at the boundaries of the network. Once the displacement vector  $u_{global}$  is obtained, it can be multiplied with the global stiffness matrix  $K_{global}$  to obtain the global force vector  $f_{global}$ . As mentioned earlier, reaction forces should



Figure 3.8: Reaction forces at boundaries of the unit cell

be zero at all interior nodes except one node that is constrained to prevent rigid body motion. However, there will be reaction forces at the boundary nodes as shown in figure 3.8. The stresses can be calculated from the reaction forces via equations 3.32-3.37.

$$\sigma_x = \frac{F_x^{s2}}{L^2} = -\frac{F_x^{s1}}{L^2} \tag{3.32}$$

$$\sigma_y = \frac{F_y^{s4}}{L^2} = -\frac{F_y^{s3}}{L^2} \tag{3.33}$$

$$\sigma_z = \frac{F_z^{s6}}{L^2} = -\frac{F_z^{s5}}{L^2} \tag{3.34}$$

$$\tau_{xy} = \frac{F_y^{s2}}{L^2} = -\frac{F_y^{s1}}{L^2} = \frac{F_x^{s4}}{L^2} = -\frac{F_x^{s3}}{L^2}$$
(3.35)

$$\tau_{xz} = \frac{F_z^{s2}}{L^2} = -\frac{F_z^{s1}}{L^2} = \frac{F_x^{s6}}{L^2} = -\frac{F_x^{s5}}{L^2}$$
(3.36)

$$\tau_{yz} = \frac{F_z^{s4}}{L^2} = -\frac{F_z^{s3}}{L^2} = \frac{F_y^{s6}}{L^2} = -\frac{F_y^{s5}}{L^2}$$
(3.37)

The elastic modulus of the network  $Y_{network}$  can be calculated via equation 3.38.

$$Y_{network} = \frac{\sigma_y}{\epsilon_y} \tag{3.38}$$

In a 2D network, stress is calculated as reaction force divided by side length of the unit cell. Therefore, both stress and elastic modulus of the network have units of force per unit length.

Another method of obtaining the elastic modulus of the network involves calculating the total elastic energy of the system. This method is outlined in the next section. Once the elastic energy of the network  $E_{network}$  is known, the elastic modulus of the network can be obtained via equation 3.39.

$$Y_{network} = \frac{2E_{network}}{\epsilon_y^2 L^3} \tag{3.39}$$

In this study, the elastic modulus is calculated via both methods. The two values of elastic modulus should be equal for a correct FEM simulation.

#### 3.2.5.2 Elastic energy of the network

The elastic response of the network can be viewed as a result of deformation of the nanotubes as well as deformation of the junctions. Nanotube deformation can be further divided into torsional deformation, axial stretching/compression and bending of the nanotube. Junction deformation can be due to stretching/compression of springs along the x, y and z directions as well as rotation along these directions. Therefore, it is necessary to calculate the contribution of each of these deformation modes to the overall elastic energy of the network. In order to do this, the elastic energy of each element in the network is added to the relevant deformation mode. The global displacement vector obtained via solving equation 3.27 contains the displacement and rotation values of each node in the deformed network. For any given element, the displacement and rotation of its two nodes can be extracted from this vector. However, since their value is in global coordinates, it

needs to be converted into the local coordinate system of that element. This can be done by multiplying the displacement vector of that element with its transformation matrix given by equation 3.16. Similarly the element local force vector can be obtained as a product of the stiffness matrix and the local displacement vector of that element. After obtaining the local element force and displacement vectors ( $f_{local}$  and  $u_{local}$ ), the elastic energies can be calculated. For a beam element, the equations for calculating elastic energy stored in axial stretching/compression ( $E_{stretch}$ ), torsional ( $E_{torsion}$ ) and bending ( $E_{bend}$ ) modes are given by 3.40-3.42.

$$E_{stretch} = \frac{1}{2} \frac{E_{tube} A_{tube}}{l_s} (u_2^{local} - u_1^{local})^2$$
(3.40)

$$E_{torsion} = \frac{1}{2} \frac{G_{tube} J_{tube}}{l_s} (\theta_{x2}^{local} - \theta_{x1}^{local})^2$$
(3.41)

$$E_{bend} = \frac{1}{2} (f_{y1}^{local} v_1^{local} + f_{z1}^{local} w_1^{local} + \tau_{y1}^{local} \theta_{y1}^{local} + \tau_{z1}^{local} \theta_{z1}^{local} + f_{y2}^{local} v_2^{local} + f_{z2}^{local} w_2^{local} + \tau_{y2}^{local} \theta_{y2}^{local} + \tau_{z2}^{local} \theta_{z2}^{local})$$

$$(3.42)$$

For a junction spring element, the equations for calculating elastic energy stored in translational  $(E_{kx}, E_{ky}, E_{kz})$  and rotational modes  $(E_{krotx}, E_{kroty}, E_{krotz})$  are given by 3.43-3.47.

$$E_{kx} = \frac{1}{2}k_x(u_2^{local} - u_1^{local})^2$$
(3.43)

$$E_{ky} = \frac{1}{2}k_y(v_2^{local} - v_1^{local})^2$$
(3.44)

$$E_{krotx} = \frac{1}{2} k_{rotx} (\theta_{x2}^{local} - \theta_{x1}^{local})^2$$
(3.45)

$$E_{kroty} = \frac{1}{2} k_{roty} (\theta_{y2}^{local} - \theta_{y1}^{local})^2$$
(3.46)

$$E_{krotz} = \frac{1}{2} k_{rotz} (\theta_{z2}^{local} - \theta_{z1}^{local})^2$$
(3.47)

The total elastic energy of the network  $E_{network}$  is then obtained by adding the elastic energy of all deformation modes of all the elements in the network. In the final result, the contribution of individual deformation modes to the overall deformation of the network is represented as a fraction of its corresponding elastic energy to the total elastic energy of the network.

#### 3.2.5.3 Non affinity measure

A network deformation is said to be affine if the local strain field is homogeneous and varies according to far-field applied strain. However, this is never exactly true in a random network of filaments as discussed in chapter 1. In order to quantify the non affinity in the network deformation, two parameters are used in this study. First is the root mean square deviation (RMS) of the vertical displacement of each node in the network from the affine prediction as mentioned in equation 3.48.

$$RMS^2 = <\left(\frac{u_y}{\epsilon_y L} - \frac{y}{L}\right)^2 > \tag{3.48}$$

Second is the standard correlation function (COREL) between the vertical displacement and initial position of each node given by equation 3.49.

$$COREL = < \left(\frac{u_y}{\epsilon_y L} - < \frac{u_y}{\epsilon_y L} >\right) \left(\frac{y}{L} - < \frac{y}{L} >\right) > /(\sigma_{u_y/\epsilon_y L} \sigma_{y/L})$$
(3.49)

For perfect affine deformation, RMS and COREL should be equal to  $\theta$  and 1 respectively. The non-affinity  $(NA_r)$  measure represented by equation 3.50 captures non-affinity in the system at different length scales

$$NA_r = \left(\frac{\Delta r_{sim} - \Delta r_{affine}}{r + \Delta r_{affine}}\right)^2 \tag{3.50}$$

Here  $u_y$  is the displacement of a node in y-direction, y is the y-coordinate of that node, angled brackets  $\langle \rangle$  denote average over all nodes in the network,  $\sigma$  denotes standard deviation, r is the distance between two nodes,  $\Delta r$  is the change in distance between that set of nodes and the subscript represents whether it is a result of simulation or affine theory.



Figure 3.9: Original configuration of an elastic beam of length l with n cross-links at an angle  $\theta$ 

# **3.3** Affine theory

Under the premise of the affine theory, the strain distribution in the network is spatially uniform. Therefore the network can be treated as an elastic continuum. Figure 3.9 shows the original configuration of an elastic beam of length l making an angle  $\theta$  with the x-axis connected by n number of crosslinks to the underlying network. A compressive strain  $\epsilon$ is applied in y direction that according to the affine assumption manifests into vertical displacement at each node of the network as  $\epsilon^* y$ . The elastic energy E of a single beam is averaged over all possible configurations  $\theta$ . The elastic modulus of the network can be extracted as  $Y_{affine} = 2NE/(V\epsilon^2)$ , where N/V is the number of elastic beams per unit volume in the network. Equation 3.51 gives the relationship between the number of beams per unit volume N/V in a network with the number of cross-links per beam n.[26]

$$\frac{N}{V} = \frac{n\pi}{2l^2} \tag{3.51}$$

This theory is developed only for a 2D network. In the case of a 3D network, each cross-link can also assume any random orientation. This effect cannot be captured using this simple model.

# 3.4 Conclusion

In this chapter we have discussed the method for generating a 2D and 3D network of nanotubes. The basic algorithms for generating junctions between the nanotubes and checking for network connectedness are examined. The finite element formulation for studying the linear deformation of these networks is also covered. It involves the stiffness matrices of beam and spring elements, coordinate transformation, assembly of elements, loading boundary conditions, solving the system of linear equations, and post-processing operations. In the next chapter, we will apply these concepts to study the deformation of CNT networks.

# Chapter 4

**FEM: Results** 

## 4.1 Introduction

In this chapter, the results obtained from the finite element simulations of deformation of 2D and 3D networks are discussed. The first section of this chapter talks about the variation in the number of junctions with network density in 2D and 3D networks. In the next section, the FEM results for a 2D network are discussed, including characterizing the network deformation as a function of network structure and junction parameters. These results are also compared with the predictions from affine theory, and the amount of nonaffine deformation in the network is calculated. Similar parameter maps are developed for a 3D network, as well and a set of rules that dictates the network deformation mode. This chapter ends with a brief discussion about the non-linear mechanical behavior of CNT networks in the event of junction fracture.

## 4.2 Network Geometry

The variation in the average number of junctions per nanotube  $n_{c2D}$  with respect to network line density  $\rho_l$  for a 2D network is shown in figure 4.1. In this simulation, network line density is increased while keeping the nanotube length l constant. 10 simulations with different initial seeds are carried out at each density and the results show the average values. The number of junctions per nanotube  $n_{c2D}$  increases linearly with  $\rho_l$  as shown in the plot. This agrees with the theoretical relationship given by equation 4.1 shown by the solid line in the plot.

$$n_{c2D} = \frac{2l\rho_l}{\pi} \tag{4.1}$$

$$n_{c3D} = \frac{\pi d_{tube} l \rho_l}{2} \tag{4.2}$$

The corresponding equation for a 3D network is given by 4.2. The number of junctions per nanotube  $n_{c3D}$  in a 3D network depends on both network line density  $\rho_l$  and nanotube diameter  $d_{tube}$ . It increases linearly with both. As mentioned earlier, in the simulations the nanotube diameter is captured using a cutoff distance (*COD*), and a crossing between



Figure 4.1: Variation of number of junctions per nanotube with 2D network density

two nanotubes is assigned as a junction only if its separation distance is below the *COD*. In the simulations, both network line density and *COD* for junction formation are varied. Figures 4.2a and 4.2b show the variation of junctions per nanotube  $n_{c3D}$  with respect to *COD* and  $\rho_l$  respectively.

A perfect data collapse for figure 4.2a can be obtained by dividing the number of junctions with  $\rho_l$  as shown in figure 4.3a. This means that the number of junctions per nanotube varies linearly with the network density in a 3D network as well. We try to fit this data to get the relationship between the variation in the number of nodes per nanotube with respect to the *COD*. It is evident from figure 4.3a that a linear fit doesn't work, especially at high *COD*. However, a quadratic polynomial fit gives an  $R^2$  value of 1. This conclusion is further supported by figure 4.3b. We get a perfect data collapse of variation of  $n_{c3D}$  with density when each data point is divided by the polynomial function of its *COD* obtained in the previous plot. The linear fit between  $n_{c3D}$  and density in this case further confirms the linear scaling behavior of the number of nodes per nanotube with respect to network line density. It can be extracted from figures 4.3a and 4.3b that the number of junctions per nanotube varies as equation 4.3.

$$n_{c3D} = l\rho_l(8.2d_{tube}^2 + \frac{\pi}{2}d_{tube})$$
(4.3)



Figure 4.2: Variation of number of junctions per nanotube in 3D network



Figure 4.3: Data collapse for number of junctions per nanotube in 3D network

It is interesting to note that this equation does reduce to equation 4.2 with the linear part dominating at very small *COD*. The coefficient of linear term is same in both the equations. In the rest of the simulations , network density will be referred to as the number of nodes per nanotube in the network.

# 4.3 FEM: 2D analysis

### 4.3.1 Variation of elastic modulus with unit cell size

In this study, the material at laboratory-scale is treated as an array of periodic unit cells deforming under cyclic boundary conditions. Therefore, the size of the unit cell can be much smaller than that of the actual material, saving us a lot of computation. However, it is also very important to determine the lower limit on the size of the unit cell such that the choice doesn't affect the results of the simulations. Therefore, we study the variation of the elastic modulus of the network  $Y_{network}$  with respect to system size, i.e, side length L of the unit cell. In these simulations, the network density is fixed and the side length L of the unit cell is varied. Since the nanotube length l used in each simulation is 1  $\mu m$ , the value of L actually represents the ratio of the length of the unit cell to the length of individual nanotubes. 10 simulations are run for each value of L. The average elastic modulus and standard error bar at each point are calculated and plotted against L. Figures 4.4a and 4.4b show such curves for low and high network densities respectively. In both of these curves, the standard error and the variation in elastic modulus are very high for small system size. As L increases, the standard error decreases, and the elastic modulus of the network becomes constant. This behavior is more pronounced in the low density network. Since the goal is to save computational cost without affecting the results, the lowest value of L at which the elastic modulus becomes constant is chosen as the default system size for the rest of the simulations. This value is  $L = 1.4 \ \mu m$ . Since the standard error is very small at this value of L, 10 simulations are carried for each set of parameters.



Figure 4.4: Variation of network elastic modulus with system size L in 2D network



Figure 4.5: Mapping of network elastic modulus with junction stiffness and network density in 2D network



Figure 4.6: Variation of network elastic modulus with junction axial stiffness and network density in 2D network

## 4.3.2 Parameter Map

The parameters to study for a 2D network can be categorized as network, nanotube and junction parameters. In this study, we fix the nanotube parameters to the experimental values mentioned in table 3.1.[7, 6, 9] The network parameter that is varied is network density,  $n_{c2d}$ . Junction parameters include translational stiffness along the x any y directions.  $k_x$ ,  $k_y$  and rotational stiffness along the z direction,  $k_{rotz}$ . In this study,  $k_x$  and  $k_y$  are assumed to be equal in magnitude. Figure 4.5 shows a parameter map of the variation of elastic modulus of the network  $Y_{network}$  with respect to  $k_x/k_y$  and  $k_{rotz}$  at different network densities  $n_{c2d}$ . It is clear from the figure that the elastic modulus only varies with  $k_x/k_y$  and not with  $k_{rotz}$ . In order to study the variation with respect to  $k_x/k_y$ , a cross-section of the graph in figure 4.5 is shown in figure 4.6. As can be seen from the figure, the elastic modulus increases with  $k_x/k_y$  initially and becomes constant after a certain value, which is equal in magnitude to the axial stiffness of the nanotube,  $E_{tube}A_{tube}/l$ (1.3375 N/m). This might suggest that for values of  $k_x/k_y$  lower than the nanotube axial stiffness, network deformation is governed by stretching/compression of cross-links, while axial deformation of nanotubes governs the deformation for stiff cross-links.



Figure 4.7: Variation of network elastic modulus with network density in 2D network

In order to extend this argument further, we look at the variation of elastic modulus of the network with respect to network density at different values of  $k_x/k_y$  as shown in figure 4.7. The results from both the affine prediction and the finite element model are shown. The affine results are fitted with the appropriate curve in order to predict the scaling law. The results obtained from simulation follow the respective scaling laws at high network densities. At low densities, however the behavior is quite different from the affine prediction. This is elaborated upon further in the next section. The scaling behavior changes with respect to  $k_x/k_y$ . At values of  $k_x/k_y$  lower than nanotube axial stiffness (figures 4.7a, 4.7b & 4.7c), the scaling behavior takes the form of a quadratic polynomial. The ratio of the quadratic to the linear term in this polynomial decreases with increasing  $k_x/k_y$ . This indicates that the scaling law becomes more linear as the junction axial stiffness is increased. At values of  $k_x/k_y$  larger than the nanotube axial stiffness (figures 4.7d, 4.7e & 4.7f), the quadratic terms completely disappear and the scaling behavior becomes exactly linear. This transition in the scaling behavior is also indicative of a change in the governing mode of network deformation with varying junction stiffness. These two different scaling responses are also mentioned in the literature of stiff[20, 21, 23, 24] and compliant cross-links.[25, 51, 52, 53]

In order to actually predict the governing mode of network deformation, we look at the fraction of total elastic energy stored in each deformation mode. Figure 4.8 shows the curves for the variation of the fraction of total elastic energy in various deformation modes as a function of network density at different values of  $k_x/k_y$ . The curves from both the affine prediction and the finite element model are shown. As mentioned before, simulation results match affine prediction at high densities. For very low values of  $k_x/k_y$ (figures 4.8a, 4.8b & 4.8c), most of the elastic energy of the network is stored in axial stretching/compression of junction springs. As the value of  $k_x/k_y$  increases beyond the nanotube axial stiffness (figures 4.8d, 4.8e & 4.8f), most of the elastic energy of the network is stored in axial deformation of the nanotubes. Figure 4.8c shows a special case when the value of  $k_x/k_y$  is still lower than the nanotube stiffness but very close to it. In this case, at intermediate densities most of the energy is stored in stretching/compression of crosslinks. But since the difference between nanotube stiffness and  $k_x/k_y$  is considerably smaller ( $\approx 1.22 N/m$ ), the energy stored in axial deformation relative to cross-link deformation



Figure 4.8: Variation of elastic energy distribution with network density in 2D network



Figure 4.9: Variation of root mean square deviation (RMS) and standard correlation function (COREL) with network density in 2D network

starts to increase with density. At very high density, the former should govern the overall deformation of the network (not shown in the graphs).

Based upon the three results presented above, it can be concluded that network deformation is governed by the stretching/compression of cross-links for compliant cross-links, while axial deformation of nanotubes governs the deformation for stiff cross-links. Such a network can be thought of as springs connected in series, where the softer spring will always govern the overall deformation. This analogy has previously been discussed in the literature.[16, 17] This is, however, only true at high network densities in some cases. At low network density, the mechanical behavior of network is no longer affine, and bending of nanotubes also plays an important role as discussed in the next section.

### 4.3.3 Non-affine deformation

As mentioned in the previous section, the scaling behavior of the elastic modulus with respect to network density is quite different from the affine prediction especially at low densities. Also the value of the elastic modulus of the network is much lower than its affine prediction at such small densities. This is because at small network density, the system can reduce is energy via softer deformation modes, that is bending of nanotubes.



Figure 4.10: Variation of non affinity measure (NA) with distance between nodes r in 2D network

This is indeed the case as evident from figure 4.8. Much of the system's elastic energy is stored in the bending mode for small network densities. As the network density increases,  $Y_{network}$  starts to scale as dictated by affine theory. However it never reaches the affine value because even at high density, some fraction of energy is still stored in bending mode as shown in figure 4.8. Out of the six cases shown in figure 4.7, only for the case when the value of  $k_x/k_y$  is very small (figure 4.7a), does the elastic modulus follows the affine prediction closely even at small density. This is because for such small value of  $k_x/k_y$  there is no contribution of nanotube bending as shown in the respective energy curve (figure 4.8a). This non-affinity in the system is captured using two parameters. First is the root mean square deviation (RMS), and second is the standard correlation function (COREL)as discussed in chapter 3. For perfectly affine deformation, RMS and COREL should be equal to 0 and 1 respectively. The variation of both *RMS* and *COREL* with network density for high and low values of  $k_x/k_y$  is shown in figures 4.9a & 4.9b respectively. It can be seen from figure 4.9a that at low network densities, the value of RMS is high and *COREL* is very low indicating non-affine deformation. But with increasing network density, RMS starts decreasing and saturates to a low value. The value of COREL,
on other hand, increases with network density and saturates to a value close to 1 at high network density. This further indicates that the deformation of a random network follows affine prediction at high densities. This is true however only for networks with stiff cross-links. For the case of compliant cross-links, there is no contribution of bending deformation even at very low network density as mentioned above. In this case, values of *RMS* and *COREL* functions remain constant with network density according to figure 4.9b. Even at low network densities, *COREL* is close to 1. This shows that deformation is always affine in a network with compliant cross-links. Non affinity is not only a function of network density, but also of the probing length scale.[16, 21, 22] In figure 4.10, non affinity in the network deformation  $NA_r$  in the system is plotted against different length scales for a high density network ( $n_{c2d}=62$ ). The value of  $NA_r$  is considerably higher at very small length scale but decreases significantly at the length scale comparable to the system size, *L*. This shows that the degree of non-affinity also depends on length scale at which it is measured. In particular, two distinct scaling regimes can be seen in figure 4.10. This kind of scaling behavior is also mentioned in the literature.[22]

Finally we look at the map of elastic energy distribution in a deformed 2D network. Figures 4.11 & 4.12 show the elastic energy distribution map for networks with stiff and compliant crosslinks respectively. As mentioned earlier these networks are subjected to a compressive strain in the y direction. For networks with stiff crosslinks as shown in figure 4.11, most of the elastic energy is stored in nanotube deformation. At high network density, most of the elastic energy is stored in axial compression of nanotubes aligned along the y direction as shown in figure 4.11a. The deformation field is also affine in this case. For a low density network, most of the elastic energy is stored in bending of a few nanotubes aligned along the x direction as shown in figure 4.11b. This also shows that deformation of the network is highly non-affine in this case. On the contrary, most of the elastic energy is stored in deformation of junctions in networks with compliant cross-links as shown in figures 4.12. The elastic energy is stored uniformly in the axial stretching/compression of junctions, irrespective of the network density in this case. This shows that network deformation is affine for very weak junctions even at low densities, further supporting the results mentioned earlier.



(a) junction axial stiffness  $k_x/k_y{=}1e09~N/m,$  network density  $n_{\rm c2D}{=}50$ 



(b) junction axial stiffness  $k_x/k_y{=}1e09~N/m,$  network density  $n_{\rm c2D}{=}11$ 

Figure 4.11: Elastic energy distribution in deformed 2D networks with stiff crosslinks



(a) junction axial stiffness  $k_x/k_y{=}1e{-}\theta 6~N/m,$  network density  $n_{\rm c2D}{=}50$ 



(b) junction axial stiffness  $k_x/k_y=1e-06$  N/m, network density  $n_{c2D}=11$ 

Figure 4.12: Elastic energy distribution in deformed 2D networks with compliant crosslinks



Figure 4.13: Variation of elastic modulus of 3D network with system size L

## 4.4 FEM: 3D analysis

#### 4.4.1 Variation of elastic modulus with unit cell size

The variation of the elastic modulus of the network  $Y_{network}$  with respect to system size, i.e, side length L of the unit cell is studied for a 3D network as well. In these simulations, network density and COD for nanotube junction formation is fixed while side length of the unit cell L is varied. The length of individual nanotubes l in these simulations is  $1 \ \mu m$ . The number of computations increases by a large amount for a 3D network especially with high density and large system size. Therefore, these simulations are run only for a low density network. The results in figure 4.13 show an average value of elastic modulus for 10 simulations at each value of L. Variation in the elastic modulus between each simulation is represented by a standard error bar. As shown in figure 4.13a, the elastic modulus is very high with a large error bar for  $L = 1 \ \mu m$ . On increasing L, the elastic modulus decreases and appears to be constant. In order to study the variation at larger L, the first data point is removed (figure 4.13b). Initially a small variation in elastic modulus with system size can be seen. It becomes constant at large system size. The



Figure 4.14: Variation of elastic modulus of 3D network with network density for different values of junction torsional stiffness  $k_{\text{rotz}}$ 

error bar length also decreases with system size. But since the variation is very small, and considering the fact that the number of computations increases significantly with system size,  $L= 1.2 \ \mu m$  is chosen as the default system size for all subsequent simulations.

#### 4.4.2 Parameter Map

As in the 2D network, we vary only network and junction parameters. However in a 3D network, parameters include network line density  $\rho_l$  and cut of distance *COD* for junction formation as discussed in section 4.1. In these simulations,  $\rho_l$  is fixed and only *COD* is varied to save computation time. The junction parameters include stiffness constants that resist translation and rotation along three local coordinate axes,  $k_x$ ,  $k_y$ ,  $k_z$ ,  $k_{rotx}$ ,  $k_{roty}$  and  $k_{rotz}$  respectively But since  $k_x$  and  $k_y$  are assumed to be equal and so are  $k_{rotx}$ ,  $k_{roty}$  and  $k_{rotz}$ , the independent junction parameters reduce to  $k_x$ ,  $k_z$  and  $k_{rotz}$  only.



(a) Variation of network elastic modulus



(b) Variation of dominant deformation mechanisms

Figure 4.15: Variation of elastic modulus of 3D network and dominant deformation mechanisms with junction axial stiffness  $k_x/k_z$  and network density  $n_{c3D}$  (Ekx, Eky, Ekz: junction deformation in x, y and z directions, Ebend, Estretch: bending and axial deformation of nanotubes)



(a) Variation of the elastic modulus (

(b) Variation of dominant deformation mechanism

Figure 4.16: Variation of network elastic modulus and dominant deformation mechanisms with junction axial stiffness in x direction  $k_x$  and network density  $n_{c3D}$  in 3D network

Figure 4.14 shows the variation of the elastic modulus of the network with density for two extreme values of  $k_{\text{rotz}}$ . There is no difference in the scaling behavior or the values of the elastic modulus for the two values of  $k_{\text{rotz}}$ . This shows that rotation at nanotube junctions doesn't play a significant role in the deformation of the overall network. This makes sense since the network is under compression. This idea is further supported using the elastic energy argument presented later in this section.

In the next set of simulations both  $k_x$  and  $k_z$  are varied while  $k_{rotz}$  is fixed. The elastic modulus of the network  $Y_{network}$  is calculated as a function of network density  $n_{c3D}$  for each case. Figure 4.15a shows a parameter map of the variation of elastic modulus of the network with respect to  $k_x$  and  $k_z$  for different network densities. Figure 4.15b shows a parameter map of the dominant deformation mechanism in the network at different values of  $k_x$ ,  $k_z$  and  $n_{c3D}$ . These plots show that  $Y_{network}$  varies with both  $k_x \& k_z$  and different mechanisms govern the network deformation at different values of  $k_x$ ,  $k_z$  and  $n_{c3D}$ . Therefore, it is better to look at the cross-sections of these maps at different values of  $k_x$  and  $k_z$ .

Figure 4.16 shows the cross-sections of figures 4.15a and 4.15b at  $k_z = 1e \cdot 06 N/m$ . It is evident from figure 4.16a that  $Y_{network}$  initially increases with  $k_x$  and becomes constant



Figure 4.17: Variation of elastic modulus of 3D network with network density

after a certain value of  $k_x$  for different network densities. Thus deformation of the network is controlled by stretching/compression of cross-links along local x and y directions for small  $k_x$ . It can also be observed in figure 4.16b that for low values of  $k_x$  most of the elastic energy of the network is stored in stretching/compression of cross-links along local x and y directions for all network densities. On increasing the value of  $k_x$ , the governing deformation mechanism switches to another kind depending upon the network density. For low network density, stretching/compression of cross-links along the local z direction governs the network deformation when  $k_x$  is increased beyond current value of  $k_z$  (1e-06 N/m). At intermediate network densities, non-affine bending of nanotubes becomes the governing deformation mechanism when  $k_x$  is increased beyond nanotube bending stiffness,  $12E_{tube}I_{tube}/l^3(2.25e-06 N/m)$ . For high network densities, affine axial stretching/compression of nanotubes becomes the governing deformation mechanism when  $k_x$  is increased beyond nanotube axial stiffness,  $E_{tube}A_{tube}/l$  (1.3375 N/m).

Figure 4.17 shows the variation of  $Y_{network}$  with respect to network density at different values of  $k_x$ . The scaling of the elastic modulus with density in the affine regime changes with respect to  $k_x$ . At values of  $k_x$  lower than the nanotube axial stiffness, the scaling behavior takes the form of a quadratic polynomial. The ratio of quadratic to linear term in this polynomial decreases with increasing  $k_x$ . At values of  $k_x$  larger than the nanotube axial stiffness, the quadratic terms completely disappear and the scaling behavior becomes exactly linear. The change in scaling behavior reflects the change in deformation mechanism from stretching/compression of crosslinks to axial deformation of nanotubes at high network density. The elastic behavior of the 3D network for the current value of  $k_z$  (1e-06 N/m), which is always less than  $k_x$  in this case, is similar to the behavior of a 2D network as discussed in the previous section. The only exception is that at very low network density, deformation is governed by the  $k_z$  term instead of  $k_x$ , since the former is smaller in magnitude. Therefore it can be said that in a 3D network of nanotubes, the two mechanisms of nanotube junction deformation (along the local x & y directions and the local z direction) can be thought of as springs connected in series for very low network density. At higher density, the two mechanisms of nanotube junction deformation (along the local x & y directions) and nanotube stretching/compression act as springs connected in series similar to a 2D network.





(b) Variation of dominant deformation mechanism

Figure 4.18: Variation of network elastic modulus and dominant deformation mechanisms with junction axial stiffness in z direction  $k_z$  and network density  $n_{c3D}$  in 3D network

The deformation of the 3D network becomes very different from a 2D network when  $k_z$  is greater than  $k_x$ . In order to study this difference, the value of  $k_x$  is fixed. Figure 4.18 shows the cross-sections of figures 4.15a and 4.15b at  $k_x = 1e \cdot 06 N/m$ . Figure 4.18a shows that  $Y_{network}$  doesn't vary with  $k_z$  at all for low network densities. However, at high network densities,  $Y_{network}$  increases with  $k_z$  and becomes constant at higher  $k_z$ . This can be explained using the elastic energy map shown in figure 4.18b. For low network density, most of the elastic energy is stored in stretching/compression of cross-links along the local x and y directions at all values of  $k_z$ . Therefore,  $Y_{network}$  doesn't vary with  $k_z$ at all for low network densities. It is in accordance with the previous finding that the two mechanisms of nanotube junction deformation act as springs connected in series for low network densities. Therefore, network deformation is governed by the  $k_x$  term in this case since it is always less than  $k_z$ . However, on increasing the value of  $k_z$  at high network densities, the governing deformation mechanism changes to stretching/compression of cross-links along the local z direction. This explains the initial increase in  $Y_{network}$  at high network density when  $k_z$  becomes greater than  $k_x$ . However, the fact that deformation is controlled by the  $k_z$  term despite it being higher in magnitude than  $k_x$  suggests that at high network density, the two mechanisms of nanotube junction deformation act as springs

connected in parallel. It might be true because at high density, the network becomes more connected and nodes in the network can't move independent of each other anymore. The governing deformation mode changes to bending of nanotubes on further increasing  $k_z$  at high network densities. Therefore  $Y_{network}$  becomes constant at large value of  $k_z$  as shown in figure 4.18a. The change in governing mechanism from junction deformation along the z direction to nanotube bending happens when  $k_z$  is increased beyond the nanotube bending stiffness. This suggests that the two mechanisms of nanotube junction deformation (along the local z direction) and nanotube bending can be viewed as springs connected in series for a 3D network.

Figure 4.19 shows the variation of  $Y_{network}$  with respect to network density at different values of  $k_z$ . For low values of  $k_z$  when deformation of the network is governed by nanotube junction deformation either along the x, y or z direction, the scaling of the elastic modulus with respect to network density is quadratic in nature. However, for larger values of  $k_z$ , when nanotube bending starts dominating, scaling behavior changes to a power law fit with a large scaling exponent. The scaling exponent becomes equal to 6.7 for very large values of  $k_z$  (figures 4.19d and 4.19e). A scaling exponent value of 6.67 is reported in the literature for the bending dominated regime in a random network.[24]

The discussion above can be summarized as follows. The elastic modulus of the network under compression doesn't vary with the rotational stiffness of nanotube junctions. The dominant deformation mechanisms are stretching and bending of nanotubes as well as translation of nanotube junctions along the local x, y and z directions. The network structure and the junction parameters determine which mechanism will govern the overall deformation of the network. The two mechanisms of junction deformation (translation along the local x & y directions and local z direction) act as springs connected in series and parallel for low and high network densities respectively. The two mechanisms of nanotube junction translation (along the local x & y directions) and nanotube stretching/compression act as springs connected in series. The two mechanisms of nanotube junction translation (along the local z direction) and nanotube bending can also be viewed as springs connected in series. This set of rules can be applied to explain all the cases shown in the parameter maps in figures 4.15a and 4.15b. Appendix B presents curves for the variation of elastic modulus with network density for each of these individ-



Figure 4.19: Variation of elastic modulus of 3D network with network density

ual cases. The corresponding elastic energy curves for each of these cases are shown in Appendix C.

### 4.5 Deformation modes in pristine SWCNT network

As mentioned earlier, single wall carbon nanotubes (SWNCTs) interact via weak van der Waals (VDW) forces at junctions in a 3D network of nanotubes. In chapter 2, VDW interactions between nanotubes is modeled using MD simulations. The values of junction stiffness obtained from the simulations are  $k_x/k_y = 0.108 \ N/m$ ,  $k_z = 116.14 \ N/m$  and  $k_{rotz}/k_{rotx}/k_{roty} = 4.59E-21 \ Nm/rad$ . These values of junction stiffness are used in the finite element model for the next set of simulations. The variation of elastic modulus of the network with respect to network density or number of junctions per nanotube  $(n_{c3d})$ is studied.

In a 3D network of nanotubes, the nanotube separation distance at junctions can span a wide interval. Figure 2.2b shows that the nanotube junction stiffness  $k_z$  along the local z direction varies with distance between the nanotubes. It is higher in magnitude at the equilibrium distance of 3.17 Å but decreases very steeply as the distance between the nanotubes increases. The value of  $k_z$  almost becomes zero above a distance of 3.529 Å. This arises from the short range nature of VDW interactions. In order to accurately predict the elastic modulus of a 3D nanotube network, the effect of the variation of  $k_z$ with respect to separation distance needs to be taken into account. In the next series of finite element simulations, the value of  $k_z$  is set to 116.14 N/m for all the junctions separations less than or equal to 3.17 Å. The value of  $k_z$  is set to a very small value (1E-06 N/m) instead of zero for separation distances greater than or equal to 3.529 Å. This is done to ensure that the FE solution is determinate. The value of  $k_z$  is set to 0.108 N/m for distances between these two values. Similarly the value of  $k_x$  is set to 0.108 N/m for distances less than or equal to 3.529 Å.

Figure 4.20 shows the variation of elastic modulus of the network with respect to network density. The scaling behavior of the elastic modulus is quadratic in nature. This means that the deformation of the network is governed mainly by the deformation of



Figure 4.20: Elastic modulus of SWCNT network v/s density

Figure 4.21: Elastic energy distribution in SWCNT network v/s density

nanotube junctions. This is confirmed by figure 4.21 that shows the variation of the elastic energy stored in different deformation mechanisms in the network as a function of network density. According to the figure, most of the elastic energy is stored in stretching/compression of cross-links along local x and y directions at all network densities. Since stretching/compression of cross-links is equivalent to nanotube sliding, it can be said that the deformation of a 3D network of SWCNTs is governed by nanotube sliding for any value of network density. The deformation field is mostly affine. Elastic energy is distributed homogeneously among the nanotube junctions as shown in figure 4.22. The value of *RMS* is low and *COREL* is close to one at all network densities as shown in figure 4.23.

In [7], the elastic modulus of pristine SWCNT aerogels under compression is reported to be 0.21 MPa. The number of nodes per nanotube in these aerogels is 83. At this value of  $n_{c3d}$ , the value of the elastic modulus of the network obtained from this simulation is 0.07 MPa. This is lower than the experimental value since we expect that in an actual CNT network, the nanotube entanglements may offer some additional stiffness. In addition, a power law scaling of the elastic modulus with a scaling exponent of 2 is reported in [7] as well which agrees with the simulation result. However, the bending of nanotubes is



Figure 4.22: Distribution of the elastic energy in SWCNT network

Figure 4.23: *RMS* and *COREL* variation with network density

speculated to be the dominant deformation mechanism in [7] as compared to the sliding of nanotubes established by this simulation. This discrepancy might arise from the fact that in the experiments, each nanotube junction is assumed to contribute equally to the total elastic energy of the network.

## 4.6 Non linear mechanics: Junction fracture

In this section, the effect of junction fracture on the mechanical behavior of a 3D network of nanotubes is studied. A junctions is allowed to fracture if the force that resists the sliding along local x or y directions reaches a critical value,  $f_{max}$ , equal to 3.84E-12 N via the MD simulations in chapter 2. It represents the phenomena of slip between nanotubes. In an actual network, there might be several instances of slip occurring before two nanotubes don't interact anymore, but in this simulation, we have assumed that only one instance of slip leads to this effect. In these simulations, the compressive strain  $\epsilon_y$  is applied to the network in small increments. The resulting stress  $\sigma_y$  in the network. The junctions that fracture are removed from the network and the network is checked for connectedness.

In the first set of simulations, the effect of varying junction stiffness,  $k_x$  and  $k_z$ , is



Figure 4.24: Stress-strain curves of 3D network at different junction axial stiffness in x direction,  $k_x$ 

studied. Figure 4.24 shows the stress-strain curves for three networks with junction stiffness,  $k_x = 1E-06 N/m$ , 0.1 N/m and 100 N/m. In all three simulations  $k_z$  is equal to 1E-06 N/m and the density of network is very small. According to this figure, no junction fracture occurs for  $k_x = 1E - 06 N/m$  since the stress-strain curve is linear. On increasing  $k_x$ , the shape of the stress-strain curve changes. For small strains when no junction fracture occurs, the stress-strain curve is linear. The slope of the linear part gives the elastic modulus of the network which is equal for both cases. This is because at low network density, the deformation is controlled by bending of nanotubes. At larger strains, nanotube junctions start to fracture, and a saw-tooth pattern is observed in the stress-strain curves. Figure 4.25 shows the percentage of fractured junctions in the network as a function of applied strain. On increasing the strain, the network density decreases as the number of fractured junctions increases. It leads to the change in the elastic energy distribution in the network as shown in figure 4.26. According to the figure, at low strain most of the elastic energy is stored in nanotube bending. The contribution of bending decreases and deformation of junctions along the local z direction increases on increasing the strain. This is in agreement with the variation of the elastic energy distribution as a function of network density, shown in figure 4.16b.



Figure 4.25: % fractured junctions in 3D network v/s applied strain

Figure 4.26: Elastic energy distribution in 3D network v/s applied strain



Figure 4.27: Stress-Strain curves at different junction stiffness in z direction,  $k_z$ 

Figure 4.28: % fractured junctions in 3D network v/s applied strain



Figure 4.29: Stress-strain curves of 3D network at different network densities



Figure 4.30: % fractured junctions in 3D network v/s applied strain

Figure 4.31: Number of junctions in 3D network v/s applied strain

Figure 4.27 shows the stress-strain curves for two different values of  $k_z$  (1E-06 N/m and 0.1 N/m). The value of  $k_x$  is fixed in both cases at 0.1 N/m. The linear part of the curves is the same in both cases since deformation is initially controlled by bending of nanotubes. But in the case of the network with higher  $k_z$ , the non-linearity in the stress-stain curve occurs at smaller strain, and stress decays to a lower value at higher strains. This is because at higher  $k_z$  the number of fractured junctions in the network is higher as shown in figure 4.28.

In the second set of simulations, the effect of varying network density is studied. Here,  $k_x$  and  $k_z$  are fixed at 100 N/m and 0.1 N/m respectively. Figure 4.29 shows the stress-strain curves at three different network densities. At low network density, the stress-strain curve has an initial linear part followed by a saw-tooth pattern at higher strains. At intermediate and high network densities, the stress strain curve is characterized by a sharp decrease in the value of stress at very small strain. This is because a large percentage of junctions fracture after the first load step as shown in figure 4.30. On increasing the strain further, the number of junctions remaining in the network decreases very rapidly for both network densities as shown in figure 4.31. After the first few load steps, the high and intermediate density networks reduce to a low density network. Therefore, their stress-strain curves becomes similar to a low density network as shown in the insets to figures 4.29b and 4.29c.

## 4.7 Conclusion

We have established that the deformation of a random nanotube network (both 2D and 3D) can be governed either by the deformation of the nanotubes (bending, axial compression) or deformation of the junctions (translation of junctions in x, y or z directions). The junction stiffness parameters in combination with network density determine the network deformation mode. The results of this study are also applicable to any stiff fiber network. This validates our first hypothesis. This study provides us with a set of rules that can be applied to predict the elastic behavior of a random network of nanotubes with any set of junction stiffness and network density. The rotation of nanotubes at junctions doesn't play a significant role in the overall deformation of the network under compression. The two mechanisms of junction deformation (translation along the local x & y directions and local z direction) act as springs connected in series and parallel for low and high network densities respectively. The two mechanisms of nanotube stretching/compression act as springs connected in series. The two mechanisms of nanotube stretching/compression act as springs connected in series.

We have also shown that the network deformation is significantly different from the predictions of affine theory, especially at smaller network densities, when deformation is governed by nanotube bending. The network elastic modulus never reaches the affine prediction even at large network densities. Although the affine theory is not developed for a 3D network, we expect the same general behavior for a 3D network as well, since whenever the network deformation is governed by nanotube bending, the deformation is non-affine.

# Chapter 5

## **Meso-scale Simulations**

### 5.1 Introduction

As discussed in the previous chapter, the elastic modulus of a CNT network as predicted via the finite element model is significantly less than the experimentally obtained value. This difference in stiffness is attributed to the entanglements between nanotubes present in a CNT aerogel that are not captured by a continuum model. This model is also not ideal for studying large deformations in a CNT aerogel, since the linear spring model of VDW interactions between nanotubes at junctions is valid only in the limit of small strain. Furthermore, the failure of nanotube junctions through multiple slip events is hard to capture in the finite element model as well. Therefore, in this chapter we will employ meso-scale simulations that have been shown to accurately model filamentous networks mediated via VDW interactions. [75, 76, 77, 78] In essence, they are a coarse graining approach in which atoms are grouped together in order to reduce the total number of degrees of freedom (DOFs) of the system. In this section, the formulation and implementation of the three most common meso-scale modeling approaches, namely the bead-spring model (BSM), the meso-scopic force field (MFF) model, and distinct element method (DEM), are discussed. The three techniques are then contrasted in terms of their accuracy and suitability to investigate the self-assembly and large deformation behavior of SWCNT networks.

#### 5.1.1 Bead Spring Model

In a bead-spring model, a carbon nanotube is represented by a group of beads as shown in figure 5.1. They are connected via harmonic springs that resist change in distance and angle between them. The beads also interact via 12-6 Lennard-Jones (LJ) potential. Therefore, the total potential energy of the system can be written as equation 5.1.[79]

$$E_{system} = E_{axial} + E_{bend} + E_{LJ} \tag{5.1}$$

where,

$$E_{axial} = \frac{1}{2} k_{axial} (r - r_o)^2$$
(5.2)



Figure 5.1: Bead-Spring representation of a CNT (taken from [79])

$$E_{bend} = \frac{1}{2} k_{bend} (\theta - \theta_o)^2 \tag{5.3}$$

$$E_{LJ} = 4\epsilon \left[ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^6 \right]$$
(5.4)

The first two terms in equation 5.1 represent the contribution of the axial and bending deformation of the nanotube and are given by equations 5.2 and 5.3 respectively. The value of the equilibrium distance  $r_o$  between beads dictates the degree of coarsening of the model. The equilibrium angle  $\theta_o$  between beads is 180 degrees. The third term is the LJ potential term given by equation 5.4 that captures the VDW interaction between nanotubes. The value of axial and bending stiffness parameters ( $k_{axial}$  and  $k_{bend}$  respectively) and distance  $\sigma$  and energy  $\epsilon$  parameters are determined from full atomistic simulations. The translational degrees of freedom and corresponding velocities of the beads are evolved with time according to Newton's second law of motion.

The LJ potential term given by equation 5.4 is symmetric in nature as it only depends on distance r between two beads. This introduces a corrugation in the total VDW potential energy  $V_{DEM}$  between the two nanotubes when plotted as a function of mismatch z between the two as shown in figure 5.2.[80] This corrugation, however, is non-physical and an artifact of this model because of replacing the cylindrical nanotubes by spherical



Figure 5.2: The total VDW potential energy  $V_{DEM}$  between two nanotubes plotted as a function of distance L and mismatch z between the two (taken from [80])

beads. Thus, it might provide additional stiffness to nanotubes against sliding and give incorrect simulation results. Reducing the degree of coarsening in the model can minimize this effect.

#### 5.1.2 Distinct Element Model

In a distinct element model, a nanotube is represented by a group of spherical particles that have translational as well as rotational degrees of freedom as shown in figure 5.3. Parallel bonds that resist shear and twist in addition to axial and bending deformation connect these particles as shown in figure 5.4. These particles also interact via LJ potential. The total potential energy of the system is given by equation 5.5.[80]

$$E_{system} = E_{axial} + E_{bend} + E_{shear} + E_{twist} + E_{LJ}$$

$$(5.5)$$

where,

$$E_{axial} = \frac{1}{2}k_n A \Delta x^2 \tag{5.6}$$

$$E_{bend} = \frac{1}{2} k_n I \Delta \theta^2 \tag{5.7}$$



Figure 5.3: Distinct element representation of a CNT (taken from [80])



Figure 5.4: Parallel bond model for elasticity of a CNT (taken from [80])



Figure 5.5: The total VDW potential energy  $V_{DEM}^5$  between two nanotubes plotted as a function of distance L and mismatch z between the two (taken from [80])

$$E_{shear} = \frac{1}{2}k_s A\Delta y^2 \tag{5.8}$$

$$E_{twist} = \frac{1}{2} k_s J \Delta \varphi^2 \tag{5.9}$$

$$E_{LJ} = V(R,\theta)\Gamma(R,\gamma) \tag{5.10}$$

$$A = 2\pi r_{CNT}h, I = \pi r_{CNT}h(r_{CNT}^2 + 0.25h^2), J = 2I$$
(5.11)

The first four terms in equation 5.5 represent the contribution of axial, bending, shear and twist deformation of a nanotube and are given by equations 5.6, 5.7, 5.8 and 5.9 respectively. The fifth term is the LJ potential term given by equation 5.10 that captures the VDW interaction between nanotubes. The value of stiffness  $(k_n, k_s)$  and LJ parameters are determined from full atomistic simulations. The translational and rotational degrees of freedom and corresponding velocities of the particles are evolved with time according to Newton's second law of motion that accounts for both forces and moments acting on them.



Figure 5.6: MFF representation of a CNT (taken from [81])

The LJ potential in this model is a function of two terms.  $V(R, \theta)$  accounts for VDW interaction between two nanotubes in parallel configuration. It depends on both distance R and angle  $\theta$  between the two particles. This introduces anisotropy in the LJ term and removes the artificial corrugation in the total VDW potential energy  $V_{DEM}^5$  between the two nanotubes when plotted as a function of mismatch z between the two as shown in figure 5.5. The second term in the LJ potential  $\Gamma(R, \gamma)$  accounts for VDW interaction if the nanotubes are in a crossed configuration and results in an aligning moment on the particles.[80]

#### 5.1.3 Meso-scopic Force Field Model

In the meso-scopic force field (MFF) model, a CNT is represented by a group of "breathing flexible" cylinders as shown in figure 5.6. The interaction of these cylinders is defined by a force field that accounts for stretching, bending, torsional and radial breathing deformation modes of a CNT. These cylinders also interact via pairwise LJ potential. The degrees of freedom in this model are the positions  $r_i$  of the end-nodes of each cylinder, the radii  $R_i$  of the cylinder at the cross-sections through the nodes, and the torsional angles  $\theta_i$  at the nodes. The Lagrangian L for this system is given by equation 5.12. [81]

$$L = \frac{1}{2} \sum_{i} m_{i} \left(\frac{\partial r_{i}}{\partial t}\right)^{2} + \frac{1}{2} \sum_{i} M_{i} \left(\frac{\partial R_{i}}{\partial t}\right)^{2} + \frac{1}{2} \sum_{i} M_{i}^{\theta} \left(\frac{\partial \theta_{i}}{\partial t}\right)^{2} - \sum_{i} U_{i}^{stretch} - \sum_{i} U_{i}^{bend} - \sum_{i} U_{i}^{breathing} - \sum_{i} U_{i}^{torsion} - \sum_{ij} U_{ij}^{LJ}$$
(5.12)

Here,  $m_i$  is the mass of a part of nanotube, represented by node *i*.  $M_i$  and  $M_i^{\theta}$  are the inertia terms for breathing and twisting modes of nanotube at node *i*. The Lagrangian is defined by the sum of kinetic energy corresponding to each degree of freedom minus the potential energy for each mode of deformation and pair-wise LJ interaction. The LJ potential term  $U^{LJ}$  in this equation captures the VDW interaction between two cylinders in a crossed configuration. The degrees of freedom of the system are evolved with time according to equation 5.13.[81]

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0 \tag{5.13}$$

where,

$$\{q_i\} = \{r_i, R_i, \theta_i\}$$
(5.14)

## 5.1.4 Comparison between bead-spring, distinct element and meso-scopic force field model

The most important thing to consider while evaluating any simulation technique is that it should be able to generate the microscopic features or the experiment findings for the problem at hand. The simulation of the complex self-assembly process present in CNT networks can serve as a challenging test for contrasting the validity and accuracy of the meso-scale modeling approaches discussed above. In studies[77, 80, 82], these techniques are employed to simulate the self-assembly of CNT films and mats. These macrostructures of nanotubes, unlike aerogels, contain large close-packed bundles of nanotubes which are entangled as shown in a TEM micrograph in figure 5.7.[80] Figures 5.8a,5.8b and 5.8c show three different self-assembled network of nanotubes generated via, BSM, DEM and MFF models respectively. The self-assembled networks obtained from the DEM and



Figure 5.7: TEM image of semi-conducting CNT film (image width and length= $0.5 \ \mu m$ ) (taken from [80])

MFF models clearly show the formation of large entangled bundles of CNTs which the BSM fails in generating. This is attributed to the large corrugation artifact in the VDW potential energy of the BSM that prevents nanotube sliding.[77] Since the CNT aerogels are mostly networks of isolated nanotubes, as concluded in experimental studies[7], the BSM might simulate correct self assembled network structure in this case, but it can still generate non-physical results during the mechanical testing of those self-assembled networks. Therefore, implementation of the correct form of LJ potential is an important factor in any meso-scale simulation technique.

The DEM and MFF techniques are capable of modeling the shear and torsional deformation of nanotubes as well.[80, 81] While the shear deformation might be important in smaller aspect ratio nanotubes[30], and torsional deformation might play an important role in mechanics of macrostructures such as CNT ropes[83], they can be neglected in modeling the mechanics of aerogels containing large aspect ratio nanotubes.

One last thing to consider is the ease with which these techniques can be implemented and the computational cost involved. The BSM can be implemented in LAMMPS which is an open-source package with parallel scaling capability and built-in implementation of several MD ensembles.[59, 60] DEM is implemented in PFC3D, which is a commercial package with no built-in implementation of MD ensembles.[80] MFF is implemented



(a) Bead Spring Model (taken from [77])



Figure 5.8: Self-assembled networks of nanotubes generated from different simulation techniques

in "Tubular" Molecular Dynamics package developed by the Computational Mechanics Group at University of Alabama. It has the inbuilt implementation of some MD ensembles, however, it is not open source.[81] Therefore, the BSM offers a major advantage in terms of its implementation.

In this study, we model the self-assembly and large deformation behavior of CNT aerogels using the bead spring model. We make modifications to the LJ term in equation 5.1 such that it is physical in modeling VDW interactions between nanotubes. We implement this model in LAMMPS by making some modifications to its source code. The modified BSM and its implementation are discussed in the next section.

## 5.2 Method

In the bead spring model (BSM) introduced in the previous section, a group of atoms of a carbon nanotube are represented by a single bead. The total potential energy of this system is given by equation 5.1. The axial and bending deformations of nanotubes are modeled as harmonic springs in this model, while the VDW interactions between nanotubes are modeled by a 12-6 LJ potential. In this section, we will discuss each of these three modes of deformation in more detail and fit the model using results from atomistic MD simulations for individual nanotubes. All the atomistic simulations reported in this section are carried out for (6,6) SWCNTs since they have the same diameter (0.8 nm) as the ones used in experiments.[7] The AIREBO potential is used to model the interactions between C atoms of nanotubes in the traditional MD simulations.[61] At the end of this section, the implementation of the corrected BSM in LAMMPS is also discussed.

#### 5.2.1 Axial Deformation of Nanotubes

Figure 5.9 shows the stress-strain curves corresponding to tensile and compression testing of individual nanotubes obtained from atomistic simulation. In the limit of small strain (< 1%), stress varies linearly with strain for both cases. The elastic modulus in this regime is obtained to be 1 TPa. A sharp decrease in the value of stress is observed in the compression case (blue curves) at a critical strain  $\epsilon_c$ . This feature in the stress-strain



Figure 5.9: Stress-Strain curves of axial deformation of a single CNT (inset shows small strain regime)

curve indicates the onset of elastic bucking of the carbon nanotube under compressive loading. The critical strain  $\epsilon_c$  depends upon the length of the nanotube. Figure 5.9 shows compression stress-strain curves for nanotubes of lengths 1.2 nm (solid curve) and 9.9 nm (dashed curve). The value of  $\epsilon_c$  for the two cases is 7.3% and 3.3% respectively indicating that the critical strain for compressive buckling decreases with increasing nanotube length. However, owing to a large axial stiffness and a small bending stiffness, the axial deformation of a nanotube by a strain of 3.3% is highly unlikely even during large deformation of a CNT network in which the mean free length of a nanotube between two junctions is 12 nm.[7] Therefore, in this study, the compressive buckling of carbon nanotubes is ignored. The axial deformation is modeled as a harmonic spring with deformation energy given by equation 5.2. The value of the axial stiffness parameter  $k_{axial}$  in this equation can be calculated as  $E_{tube}A_{tube}/r_o$ , where  $E_{tube}$  is the elastic modulus of the carbon nanotube in the elastic regime,  $A_{tube}$  is its area of cross-section, and  $r_o$  is the equilibrium distance between the beads. In this study, the value of  $r_o$  is chosen as 1 nm. The value of  $E_{tube}$  as obtained from the atomistic simulation is 1 TPa and  $A_{tube}$  is calculated using the diame-



Figure 5.10: Nanotube bending with arrows showing direction of applied displacement



Figure 5.11: Results of atomistic simulations of bending of nanotubes of different lengths

ter of (6,6) SWCNT. Therefore, the value of the  $k_{axial}$  parameter is calculated to be 54.2 eV/Å<sup>2</sup>. Figure 5.9 also shows the compressive stress-strain curve for a single nanotube obtained from the BSM (black curve). It matches the atomistic simulation curves in the limit of small strain as shown in the inset to the figure.

#### 5.2.2 Bending Deformation of Nanotubes

In order to study the bending deformation behavior of a nanotube, a rigid body translation is applied to nanotube ends as shown in figure 5.10. Figures 5.11a and 5.11b show the variation of bending energy  $E_{bend}$  (blue curve) and its derivative  $dE_{bend}/d\Phi$ 



Figure 5.12: Buckling kink formed along the nanotube surface under compression

(green curve) as a function of bending angle  $\Phi$  for nanotubes of length 10 nm and 20 nm respectively. In both cases, initially,  $E_{bend}$  scales with  $\Phi^2$  and  $dE_{bend}/d\Phi$  scales linearly with  $\Phi$ , indicating a harmonic bending regime in which Hooke's law is obeyed. However, past a critical bending angle  $\Phi_c$ , scaling of  $E_{bend}$  with respect to  $\Phi$  becomes linear and  $dE_{bend}/d\Phi$  becomes constant. This sharp decrease in the derivative of  $E_{bend}$  indicates the formation of buckling kinks along the nanotube surface, which is under compression, as shown in figure 5.12.

The critical bending angle  $\Phi_c$  at which the buckling occurs depends upon the length of the nanotube.  $\Phi_c$  for 20 nm long tubes is almost double that for 10 nm long tubes. This leads to the definition of critical bending buckling curvature,  $\kappa_c = 2\Phi_C/L$ , where Lis the nanotube length.  $\kappa_c$  for (6-6) SWCNT as obtained from the atomistic simulation is 0.0523 rad/nm. The critical bending buckling strain can thus be defined as  $\epsilon_c = \kappa_c d/2$ , where d is the tube diameter.  $\epsilon_c$  for (6-6) SWCNT of any length is thus obtained to be 2.1%. Since bending of nanotubes is a much softer mode of deformation, this critical bending buckling strain can be achieved during large deformation of a CNT network. In [84], bending buckling of nanotubes is shown to be an important factor in the formation of a stable CNT network via spontaneous self assembly. Therefore, in this study the bending buckling of carbon nanotubes is included in the bead spring model. The bending energy and force are then given by equations 5.15 and 5.16 respectively.

$$E_{bend} = \begin{cases} \frac{1}{2} k_{bend} (\theta - \theta_o)^2 & \theta > \theta_c \\ k_{bend} (\theta_c - \theta_o) (\theta - \theta_o) - \frac{1}{2} k_{bend} (\theta_c - \theta_o)^2 & \theta < \theta_c \end{cases}$$
(5.15)



Figure 5.13: Results of bead spring model of bending of nanotube of length 10 nm

$$F_{bend} = \begin{cases} -k_{bend}(\theta - \theta_o) & \theta > \theta_c \\ -k_{bend}(\theta_c - \theta_o) & \theta < \theta_c \end{cases}$$
(5.16)

Here,  $\theta$  is the angle between the beads as shown in figure 5.1 and is related to the bending angle  $\Phi$  as  $\theta = \pi - 2\Phi$ . The value of  $\theta_c$  (3.07 radian) is chosen such that bending buckling of a nanotube in the BSM occurs at a slightly smaller bending angle than in atomistic simulations. This is done to ease the formation of a stable CNT network during self-assembly. The bending stiffness parameter  $k_{bend}$  in the above equations can be calculated as  $E_{tube}I_{tube}/r_o$ , where  $I_{tube}$  is the second moment of inertia of a nanotube. The value of  $k_{bend}$  is obtained to be 526.2 eV. Figure 5.13 shows the results of a BSM simulation of bending of a nanotube of length 10 nm. The two distinct, harmonic bending and post buckling regimes can be seen in both curves.



Figure 5.14: VDW interaction energy between two nanotubes as a function of distance d



(a) Parallel configuration of nanotubes

(b) Sliding of nanotubes

Figure 5.15: Configurations of two carbon nanotubes system


(a) VDW energy as a function of sliding distance z (b) Sliding force as a function of sliding distance z

Figure 5.16: Results of sliding simulation of two nanotubes

## 5.2.3 van der Waals Interactions between Nanotubes

Figure 5.14 shows VDW interaction energy between two nanotubes as a function of distance d between them as obtained from atomistic simulations (red curve) and the BSM (blue curve). In the BSM, the beads are allowed to interact via 12-6 LJ potential given by equation 5.4. The nanotubes in these simulations are in a parallel configuration as shown in figure 5.15a. The parameters  $\epsilon$  and  $\sigma$  in this equation are chosen such that the well-depth  $\epsilon_{eq}$  and equilibrium distance between nanotubes  $r_{eq}$  obtained from atomistic and BSM simulation are equal. However, this potential can't reproduce the shape of the VDW interaction energy curve, obtained from atomistic simulations. This is because the analytical form of tube-tube potential given by equation 5.17 is different from the 12-6 potential.[56]

$$U_{tube-tube} = \left(\frac{A}{d^{9.5}} - \frac{B}{d^{3.5}}\right)$$
(5.17)

Here, A and B are the respective constants of repulsive and attractive terms. Apart from different scaling of the attractive and repulsive terms in the two potentials, the beads are assumed to be point particles in the BSM. This assumption incorrectly replaces the



Figure 5.17: Results of sliding simulation of two nanotubes

distance d between nanotubes in equation 5.17 with distance r between beads present in equation 5.4. This can however easily be corrected as discussed later in this section. The main problem in the BSM lies in the assumption of replacing cylindrical tubes with spherical beads. Figures 5.16a and 5.16b show the variation of VDW interaction energy and sliding force between two nanotubes as a function of sliding distance z for both atomistic and BSM simulation. The nanotubes in these simulations are in a parallel configuration and have equilibrium spacing  $r_{eq}$  between them as shown in figure 5.15b. The curves obtained from the BSM show large corrugation while the atomistic simulation curves are smooth. This corrugation is an artifact of the BSM and can prevent free sliding of nanotubes against each other. Therefore, the LJ term in BSM should be modified to remove any corrugation artifact.

Since corrugation is a consequence of coarse graining, a small degree of coarse graining will result in a smaller corrugation. But on reducing the degree of coarse-graining, the number of degrees of freedom in the system also increases. This in turn increases the computation time drastically. The value of  $r_o$  dictates the degree of coarse-graining of the model. Instead of decreasing the value of  $r_o$ , one of the simplest way of removing



Figure 5.18: VDW interaction energy between two nanotubes as a function of distance d

corrugation artifact without affecting computation time is to increase the value of  $\sigma$  in equation 5.4.[76] Figures 5.17a and 5.17b show the results of BSM simulations carried out with  $\sigma = 2.3r_o$ , for the variation of VDW interaction energy and sliding force between two nanotubes as a function of sliding distance z. The energy curve doesn't show any corrugation while the corrugation in the force curve is drastically reduced and is of the same order as the small corrugation observed in the atomistic curve due to the atomic structure of nanotubes. However, this simulation still cannot reproduce the exact shape of the VDW interaction energy curve for two parallel nanotubes as a function distance d between them as obtained from atomistic simulation and shown in figure 5.18. Note that the curve obtained from the BSM simulation as shown in this figure is shifted along the x-axis for comparing it with the atomistic simulation curve. The actual value of the atomistic value. Changing the functional form of LJ potential, such that it can reproduce the atomistic simulation curve, results in corrugation.

In order to implement a LJ potential similar to equation 5.17 in the BSM without any corrugation artifact, the degree of coarse graining needs to be decreased. But, as men-



Figure 5.19: Calculation of 10-4 LJ interaction between particles i and j



Figure 5.20: Results of sliding simulation of two nanotubes



Figure 5.21: VDW interaction energy between two nanotubes as a function of distance d

tioned earlier, this increases the computation time drastically, because for an N particle system, 6N first order ODEs need to be solved, which requires calculating LJ interactions between approximately  $N(N_{cut})/2$  pairs. Here,  $N_{cut}$  is the number of particles within cut-off distance of a single particle. In order to reduce the degree of coarsening without affecting the total number of particles, the LJ potential is calculated in a different manner. A line segment formed by two neighboring particles of a nanotube is divided into n ghost particles as shown in figure 5.19. The LJ interaction between a pair of actual particles, i and j, is then calculated as a summation of LJ interactions between pairs of ghost particles (shown in red). Since each particle has (n-1)/2 ghost particles on either side and one ghost particle at its center, each pair interaction between the actual particles is the summation of  $n^2$  ghost pair interactions. The contribution of these ghost particles is only towards calculating LJ interactions in order to reduce the degree of coarse graining. They don't sum to the total number of degrees of freedom of the system. Therefore, the total number of equations of motion to be solved still remain the same, although the total number of LJ interactions that need to be calculated increase by a factor of  $n^2$ . In order to reproduce the exact shape of the VDW interaction energy curve as obtained from

LJ Potentials	$\epsilon(eV)$	$\sigma(\AA)$	$r_{cut}$ (Å)
12-6	0.42	10.32	25
12-6 (large $\sigma$ )	0.39	23.1	40
10-4	0.0072	2.9	28.34

Table 5.1: LJ potentials parameters

atomistic simulation, a LJ potential similar to equation 5.17 is implemented. The exact form of the LJ potential  $U_{LJ}^{ij}$  between two particles i and j is given by equation 5.18.

$$U_{LJ}^{ij} = \sum_{k=1}^{n} \sum_{l=1}^{n} 4\epsilon \left[ \left( \frac{\sigma}{r_{kl} - 2r_{cnt}} \right)^{10} - \left( \frac{\sigma}{r_{kl} - 2r_{cnt}} \right)^4 \right]$$
(5.18)

Here,  $r_{kl}$  is the distance between ghost particles k and l belonging to actual particles i and j respectively.  $r_{cnt}$  is the radius of (6,6) SWCNT. In this study, each nanotube segment between two neighboring particles is divided into 9 ghost particles (n=9). This is sufficient to remove any corrugation artifacts in the energy and force curves for sliding as shown in figures 5.20a and 5.20b respectively. This also reproduces the exact shape of the VDW interaction energy curve between two nanotubes as a function of distance d between them as shown in figure 5.21.

Table 5.1 lists values of  $\sigma$  and  $\epsilon$  parameters and cut-off distance  $r_{cut}$  for the three different LJ potentials discussed above. It should be noted that both first and second order corrections are applied to all three potentials so that both the energy and its first derivative are continuous across the cut-off. Figure 5.22 shows the contrast between the shapes of VDW interaction energy curves for parallel nanotubes for the three LJ potentials. Figures 5.23a and 5.23b show the contrast in the amount of corrugation in the energy and force curves respectively for the sliding case for these three different potentials. Note that the 10-4 potential performs best in all cases.

In order to test these potentials, the simulation of self-assembly of a three carbon nanotube system is carried out and the results are compared with the atomistic simulation results. In the original configuration as shown in figure 5.24, three nanotubes of length 400 nm each form a equilateral triangle with a spacing of  $r_{eq}$  and an angle of  $60^{\circ}$  between them



Figure 5.22: VDW energy between two nanotubes as a function of distance d



Figure 5.23: Results of sliding simulation of two nanotubes



Figure 5.24: Configuration of three carbon nanotubes system

at the three junctions. The nanotubes are allowed to equilibrate in a NVE ensemble for 2500 ps with an initial temperature of 0 K. Figure 5.25 shows the snapshots of nanotube configurations at various time-steps as obtained from atomistic simulation. The VDW interactions are strongest at junctions, and they tend to rotate the nanotubes in order to align them. But since the triangular configuration constrains the rotation of an entire nanotube at once, nanotubes start to bend and a small curvature appears at their centers as visible in the snapshot at 1000 ps. However, since the decrease in VDW energy is much higher than the increase in bending energy of the nanotubes, the total potential energy of the system decreases with the decrease in angle between the two nanotubes as shown in figure 5.26. The rate of decrease of potential energy with angle is initially small but as the angle between the nanotubes becomes smaller than a critical angle, the VDW interactions at junctions become so strong that potential energy starts decreasing at a faster rate. This configuration of nanotubes, which occurs at a critical angle of  $25^{\circ}$ , is shown in the snapshot at 1900 ps. Once the nanotubes at junctions are completely aligned, the alignment of the rest of the nanotubes happens via a zipping action which can clearly be observed in the snapshots of later time-steps. In the final equilibrium configuration, corresponding to



Figure 5.25: Snapshots of 3 CNTs configurations as obtained from atomistic simulation



Figure 5.26: Potential energy of 3 CNTs system as a function of angle

the snapshot at 2500 ps, the entire length of the nanotubes is aligned except for a small section at the center where the large bending curvature prevents nanotubes from aligning further.

The three LJ potentials discussed above produced the same configurations as the atomistic simulation for the three carbon nanotube system. The snapshots of the nanotube configurations at various time-steps as obtained from the BSM simulation are shown in figure 5.27. However in the simulations carried out using 12-6 type LJ potentials, the nanotube configuration corresponding to the critical angle happens at an earlier-time step, although the value of the critical angle remains the same. This can also be observed in the variation of potential energy with time curves shown in figure 5.28. In each curve, three different regimes can be observed. In the first regime, the rate of decrease of potential energy is small. This regime is also shown in detail in the inset to this figure. In the second regime, the rate of decrease of potential energy suddenly increases. This transition happens at the nanotube configuration corresponding to the critical angle. In the third regime, the potential energy eventually becomes constant with time when the equilibrium configuration of the nanotubes is achieved. In figure 5.28, the curves obtained from 12-6



Figure 5.27: Snapshots of 3 CNTs configurations as obtained from bead spring simulations



Figure 5.28: Potential energy of 3 CNTs system as a function of time (inset shows energy variation in the first regime)

type LJ potentials are shifted along the time axis. This is because in these simulations the transition from the first regime to the second happens early. Only the 10-4 LJ type potential reproduces the same curve as obtained from the atomistic simulation. However, the implementation of 10-4 LJ type potential is computationally expensive relative to 12-6 type potentials. The time taken to run this simulation for 2500 ps on a single processor is ten times greater for 10-4 LJ type potential as compared to 12-6 type potential. This factor might increase to a higher value as particles move closer to each other and the number of particles  $N_{cut}$  within cut-off distance of a single particle increases.

# 5.2.4 Implementation in LAMMPS

LAMMPS is an open-source package with parallel scaling capability and built-in implementation of several MD ensembles.[59, 60] Traditionally, implementation of the beadspring model in LAMMPS is used for modeling polymer systems. In this study, we implemented the BSM for carbon nanotubes systems in LAMMPS. Since some parts of this model are different from traditional bead spring models, certain changes are made in the LAMMPS source code in order to implement this model. In this section, these changes are discussed in more detail.

The axial deformation of nanotubes is modeled as a simple harmonic spring in our BSM. In LAMMPS, several types of bond interactions can be specified between a pair of atoms. In our case, we specify harmonic bond potential between pairs of atoms belonging to the same nanotube to model axial deformation of nanotubes. Similarly, for modeling bending deformation of nanotubes, a harmonic angle potential is specified between triplets of atoms belonging to the same nanotube. However bending deformation of nanotubes also incorporates bending buckling of nanotubes past a critical strain. Therefore, in order to implement this in LAMMPS, a function named *AngleHarmonic::compute* in the *angle\_harmonic.cpp* file in the LAMMPS source code is modified accordingly. Basically we first calculate the angle  $\theta$  between the atoms in a triplet and if  $\theta < \theta_c$ , then the force on each atom and the bending energy of this triplet is calculated using equations for bending buckling as given by equations 5.15 and 5.16. The modified c++ code is presented in Appendix D.

The VDW interactions can be implemented in LAMMPS by specifying pairwise interactions between non-bonded atoms belonging to different nanotubes. The 12-6 type LJ potentials are simply implemented by specifying pairwise interactions to be lj/smooth/*linear* which models a 12-6 LJ potential with first and second order corrections. However, for implementing the 10-4 type LJ potential discussed above, a function named *PairLJ*-*Cut::compute* in the source file *pair\_lj\_cut.cpp* is modified. In order to calculate the potential between atoms i and j, the orientation of each atom with respect to the neighboring atoms is required. It can be extracted by accessing the coordinates of neighboring atoms. But since LAMMPS is written to be implemented on a parallel machine, the simulation volume is first divided amongst different processors and the atoms on each processor have a local id which is different from their global id. Therefore, we first get the global id of atom i by calling the tag/i function. The global ids of neighboring atoms will then be tag[i]+1 and tag[i]-1. Then we convert these global ids back to local ids, m and n, by calling  $atom \rightarrow map(tag[i]+1)$  and  $atom \rightarrow map(tag[i]-1)$  functions. But since atoms are divided amongst different processors in LAMMPS, multiple copies of an atom on different processors might exist. We need copies of neighboring atoms that are closest to atom *i*. Therefore, images of atoms m and n which are closest to atom i are obtained by calling domain->closest\_image(i,m) and domain->closest\_image(i,n) functions. Similarly local ids of neighboring atoms of atom j are obtained. Using the coordinates of neighboring atoms, the orientation of atoms i and j is calculated. This information is used to calculate the positions of ghost atoms surrounding atoms i and j. Then the LJ interactions (both forces and energies) are calculated between pairs of these ghost atoms according to equation 5.18. These interactions are then summed up to calculate LJ interaction between atoms i and j. The modified C++ code in presented in Appendix E.

Besides the implementation for different force-fields as discussed above, LAMMPS already has inbuilt implementation of different ensembles which are used during equilibration and deformation simulations. In deformation simulations, the strain is applied on a system by deforming the simulation box at a specified strain rate using *deform fix* command in LAMMPS.

# 5.3 Results

## 5.3.1 Self-assembly of a CNT Network

It is very important to ensure that the structure of a CNT network, whose mechanical behavior is to be studied, is physical and energetically stable at room temperature. Such a network can be generated via self-assembly. During the self-assembly process, a random network of nanotubes, which is represented by a group of beads, is allowed to evolve under a force field defined in the bead-spring model discussed above. The system moves towards either a local or global energy minimum with time. If the equilibration process is carried out at room temperature, then the system may take a long time to reach an energy minimum. This process can be expedited by subjecting the system to higher temperature or pressure. In this section, the results of some self-assembly simulations, carried out using the BSM with the three different LJ potentials, are discussed. A quantitative as well as qualitative comparison of the self-assembled networks with experimentally observed structures is also presented.



Figure 5.29: Random network of 1  $\mu m$  nanotubes within a simulation box of length 1.05  $\mu m$ 

#### 5.3.1.1 12-6 LJ potential

A random network of nanotubes as shown in figure 5.29 is generated via an algorithm similar to the one used for the finite element model discussed in chapter 3. The length of each nanotube in this network is 1  $\mu m$ . Each nanotube is represented by 1001 beads with a spacing of 10 Å between two beads. Each bead has a mass of 1170.96 amu. There are a total of 4900 nanotubes and therefore, 4904900 beads in the network. The length of the simulation box is 1.05  $\mu m$ . The initial density of the network is 8.24 mg/mL. This random network is first subjected to quasi-static energy minimization. In this step, the energy of the system is minimized by iteratively adjusting the atom coordinates. This step is important because in a random network of nanotubes, if the spacing between two nanotubes is less than the equilibrium spacing, then the forces acting on them will be very high. The simulation might blow up if such a network is directly subjected to a dynamic run. There are several energy minimizers implemented in LAMMPS. cg and quickmin styles of minimizers are used in the study.[59, 60] The network obtained from the quasi-static energy minimization is then subjected to a sequence of dynamic equilibration



Figure 5.30: Potential energy of CNT network as a function of time (inset shows energy fluctuations in the last 10 ns)

steps. The network is initially equilibrated for 10 ns in an NVE ensemble with an applied Langevin thermostat at 300 K.[85] The network is then heated to 600 K over 10 ns and held there for 20 ns. The network is finally cooled down to 300 K over 10 ns and held there for 20 ns. This whole annealing-quench process is carried out in an NPT ensemble so that a pressure of 0 Pa can be maintained at the boundaries of the simulation box.

The potential energy of the system decreases with time as shown in figure 5.30. The initial rate of decrease of potential energy is small when the equilibration is carried out at room temperature. After 10 ns, energy starts decreasing at a faster rate owing to applied pressure and temperature during the equilibration process. The rate of change of potential energy then again decreases, and the potential energy of the system eventually becomes constant indicating that the system has achieved a local energy minimum. The configuration of the system at the end of 70 ns is taken as the final equilibrium configuration, since by then the energy fluctuations have reduced to less than 1% as shown in inset to figure 5.30. The network structure evolves via bundling and bending of nanotubes. This is reflected in figure 5.31 showing the variation of the system increases with time but this is compensated by a much larger decrease in the VDW energy of the system due



Figure 5.31: Variation of components of potential energy of CNT network during selfassembly

to some degree of bundling among nanotubes. There is not much variation in the axial deformation energy of the system.

Figure 5.32 shows the final self-assembled network of nanotubes. The density of this network is 10.48 mg/mL which is in the same range as densities of aerogels (7.3-18.5 mg/mL) used in the experiments.[7, 6, 86] A closer look at the structure of this self-assembled network of nanotubes shows both entanglements and some degree of bundling among the nanotubes as shown in figure 5.33. Both of these features can also be observed in SEM and TEM micrographs of actual CNT aerogels shown in figures 5.34a and 5.34b respectively. It can therefore be said that the self-assembled network of nanotubes closely resembles an actual CNT aerogel network. In order to do a more quantitative analysis, we look at a histogram showing the distribution of VDW energy among nanotubes shown in figure 5.35. A total of four peaks can clearly be seen in this curve. These peaks correspond to the energies of nanotubes with zero, one, two and three nearest neighbors. Using this data and the surface areas of the individual configurations, the total specific surface area of this self-assembled network is calculated to be 1103 m<sup>2</sup>/gm. This value is close



Figure 5.32: Final configuration of self assembled nanotube network (nanotubes are color coded according to VDW energy, scale shown on right)



Figure 5.33: Bundling and entanglements present in CNT network



Figure 5.34: Micrographs of CNT aerogels (taken from [7])



Figure 5.35: Distribution of VDW energy among nanotubes in CNT network (Number of nearest neighbors corresponding to each peak is indicated in roman numerals at the top)



Figure 5.36: Potential energy of CNT network as a function of time (inset shows energy fluctuations in last 10 ns)

to the experimentally measured value of  $1190 \text{ m}^2/\text{gm}$  for CNT aerogels.[86] This analysis further validates the self-assembled network generated using 12-6 potential. Although, this potential generates large corrugation artifacts that prevent the free sliding of nanotubes, it is able to simulate the correct network structure of the CNT aerogel. This is because during the fabrication of SWCNT aerogels, nanotubes are coated with surfactants that prevent nanotubes from bundling together by acting as barriers to sliding of nanotubes.[7] These surfactants are later removed once a stable network forms.

## 5.3.1.2 12-6 LJ potential (large $\sigma$ )

The same random network of nanotubes shown in figure 5.29 is used in this simulation as well. This network is first subjected to a quasi static energy minimization and then a series of dynamic equilibration steps, which include equilibration in an NVE ensemble with an applied Langevin thermostat for 5 ns followed by equilibration in an NPT ensemble with 0 Pa pressure at simulation box boundaries for 35 ns. The equilibration is carried out at 300 K.

The potential energy of the system decreases with time and eventually becomes constant as shown in figure 5.36. The curve is qualitatively similar to the one shown in figure



Figure 5.37: Variation of components of potential energy of CNT network during selfassembly

5.30 except the potential energy saturates earlier in this case. The final value of potential energy in this case is also almost three times that of the earlier case. The configuration of the system at the end of 40 ns is taken as the final equilibrium configuration, since by then energy fluctuations have reduced to less than 1% as shown in inset to figure 5.36. The variation of components of potential energy of the system as shown in figure 5.37 is also similar to the earlier case. The final value of the VDW energy is 2.5 times that of the previous case due to large degree of bundling amongst nanotubes.

Figure 5.38 shows the final self-assembled network of nanotubes. The density of this network is 10.64 mg/mL. The network consists of large bundles of carbon nanotubes entangled at junctions between them as shown in figure 5.39. These large bundles of nanotubes differentiate this self-assembled network of nanotubes from actual CNT aerogels. A total of seven peaks are present in the distribution of VDW energy as shown in figure 5.40. These peaks correspond to the energies of nanotubes with zero, one, two, three, four, five and six nearest neighbors. Based on this data, the total specific surface area of this self-assembled network is calculated to be  $618 \text{ m}^2/\text{gm}$ . This value is much lower than



Figure 5.38: Final configuration of self assembled nanotube network (nanotubes are color coded according to VDW energy, scale shown on right)



Figure 5.39: Bundling and entanglements present in CNT network



Figure 5.40: Distribution of VDW energy among nanotubes in CNT network (Number of nearest neighbors corresponding to each peak is indicated in roman numerals at the top)

the experimentally measured value. This puts this self-assembled network in a different category from the CNT aerogel. In studies [84, 82, 80] which employ different modeling techniques such as DEM and MFF, similar self-assembled structures of nanotubes containing large bundles are obtained.

The effects of changing the initial network structure or equilibration process are also studied to ascertain whether such changes can produce structures similar to CNT aerogels. In one such simulation, the network shown in figure 5.32, which is obtained from the previous model, is used as an initial structure. In other simulation, an ordered mesh of nanotubes with  $r_{eq}$  as spacing between nanotubes at junctions. as shown in figure 5.41, is used as an initial structure. Both of these simulations resulted in similar nanotube networks with large bundles as shown in figure 5.38. In another simulation, instead of expediting the equilibration process by applying some pressure or temperature, the network is allowed to equilibrate slowly at room temperature. Figure 5.42 shows the variation of potential energy of the system with time. The potential energy has not become constant even after 1.5  $\mu s$  which is approximately 150 million time-steps. The



Figure 5.41: Ordered mesh network of carbon nanotubes



Figure 5.42: Potential energy of CNT network as a function of time



Figure 5.43: Configuration of self assembled carbon nanotube network at 1.5  $\mu s$  (nanotubes are color coded according to VDW energy, scale shown on right)

degree of bundling present in the network at 1.5  $\mu s$  as shown in figure 5.43 is already higher than the network obtained from the previous model and thus this network also doesn't resemble the structure of a CNT aerogel. It can be expected that given enough equilibration time, this network will ultimately evolve into the network shown in figure 5.38. Thus, neither changing the initial structure or the equilibration process, resulted in a different network for this potential.

#### 5.3.1.3 10-4 LJ potential

Since the 10-4 LJ potential requires more computation time, a smaller system size is chosen. In this case, a random network of 943 nanotubes is generated within a simulation box of length 420 nm as shown in figure 5.44. The length of each nanotube is 400 nm and is represented by 400 beads. The initial density of the network is 9.9 mg/mL. This network is first subjected to a quasi static energy minimization and then a series of dynamic equilibration steps, which include equilibration in an NVE ensemble with an applied Langevin thermostat for 5 ns followed by equilibration in an NPT ensemble with



Figure 5.44: Random network of 400 nm nanotubes within a simulation box of length 420 nm

0 Pa pressure at simulation box boundaries for 15 ns. The equilibration is carried out at 300 K.

Figure 5.45 shows the variation of the potential energy of the system with time. The potential energy decreases with time and so does the rate of decrease of potential energy. With time, as the bundling among the nanotubes increases, the number of particles within the pairwise interaction cutoff distance of a single particle also increases. This increases the computation time drastically. Therefore, the simulation is discontinued after 15 ns despite the fact that the system has not achieved an equilibrium configuration by then. However, some information can still be derived from simulation results generated until 15 ns. The equilibration proceeds by an increase in bending energy compensated by a decrease in VDW energy of the system as shown in figure 5.46. This behavior is similar to the previous cases.

Figure 5.47 shows the self-assembled network at the end of 15 ns. The final density of network is 11.9 mg/mL. The network structure is similar to the one obtained from 12-6 LJ potential with large  $\sigma$  as shown in figure 5.38. The large bundles of nanotubes with



Figure 5.45: Potential energy of CNT network as a function of time



Figure 5.46: Variation of components of potential energy of CNT network with time (inset shows the axial deformation component)



Figure 5.47: Final configuration of self assembled nanotube network (nanotubes are color coded according to VDW energy, scale shown on right)



Figure 5.48: Bundling and entanglements present in CNT network



Figure 5.49: Distribution of VDW energy among nanotubes in CNT network (Number of nearest neighbors corresponding to each peak is indicated in roman numerals at the top)

entanglements between them at junctions is present in this network as well as shown in figure 5.48. Figure 5.49 shows the VDW energy distribution amongst the nanotubes. It also shows seven distinct peaks. The specific surface area in this case is larger than the previous case and is calculated to be 699 m<sup>2</sup>/gm. But since this is not the equilibrium configuration of the network, we can expect it to eventually reach the same value when this network finally evolves into the network shown in figure 5.38. As discussed before, the 10-4 LJ potential is of a similar form to the actual tube-tube potential given by equation 5.17. This potential was also able to reproduce the shapes of VDW energy curves obtained from atomistic simulations for 2 and 3 CNTs systems. Therefore, it can be argued that the self-assembled network of nanotubes, containing large bundles and entanglements amongst them as obtained from both a 10-4 LJ potential and a 12-6 LJ potential with large  $\sigma$ , is a physical and energetically stable configuration of nanotubes at room temperature when sliding between them is not hindered.



Figure 5.50: Stress-strain curve of CNT network obtained from 12-6 LJ potential

## 5.3.2 Mechanical Behavior of a CNT network

The CNT networks obtained via self-assembly are subjected to compressive loading in order to study their mechanical behavior. Only the networks obtained from simulations employing 12-6 LJ potentials are considered in this study since they are well equilibrated. The simulation box is deformed along the z axis at a constant loading rate of 10 m/sec. A pressure of 0 Pa is maintained at boundaries perpendicular to the x and y axes. The simulation is carried out at a temperature of 300 K.

Figure 5.50 shows the stress-strain curve obtained from the compressive testing simulation of the network shown in figure 5.32, carried out using a 12-6 LJ potential that prevents sliding of nanotubes. The curve shows three different regimes: a linear elastic regime (for strains e < 2%), a plateau regime (for strains 2% < e < 50%) and a densification regime (for strains e > 50%). These three regimes are also present in the stress-strain curve of CNT aerogels obtained from experiments as shown in figure 5.51.[7] However, the linear elastic regime in the experimental curve continues upto 9% strain. The value of the plateau stress in the simulation curve is also very high resulting in an elastic modulus



Figure 5.51: Stress-strain curve of a CNT aerogel (taken from [7])

of 10 MPa, which is two orders of magnitude higher than the experimentally obtained value of 0.21 MPa. Such a high value of stiffness can be attributed to the large corrugation artifacts for this potential. This is also reflected in the curve shown in figure 5.52 exhibiting the variation of different components of stress with strain. Since nanotubes cannot slide freely, the axial deformation component of the stress is highest. It dictates the shape of the overall stress-strain curve. This emphasizes the role of nanotube sliding in the deformation of CNT networks. This is why the 12-6 LJ potential fails at capturing the correct deformation behavior of CNT networks although it generated the correct structure during self-assembly.

Figure 5.53 shows the stress-strain curve obtained from the compressive testing simulation of the network shown in figure 5.38, carried out using the 12-6 LJ potential with large  $\sigma$  that corrects the corrugation artifact and thus allows free sliding of nanotubes. This curve also shows three different regimes: a linear elastic regime (e < 10%), a plateau regime (10% < e < 50%) and a densification regime (e > 50%). The strain limits of these regimes, especially the elastic regime, are very similar to those of the experimental CNT aerogel. The value of the plateau stress in this case is 0.027 MPa, which results in an elastic modulus of 0.27 MPa. The actual density of this network is 10.64 mg/mL. The experimental elastic modulus of a CNT aerogel corresponding to a density of 9.9 mg/mL is



Figure 5.52: Variation of stress components of CNT network with strain



Figure 5.53: Stress-strain curve of CNT network obtained from 12-6 LJ potential with large  $\sigma$ 



Figure 5.54: Variation of potential energy of CNT network with strain (inset shows energy variation in linear regime)

0.22 MPa.[7] In [7], it is also noted that the elastic modulus scales with density as power law with an exponent of 2. Therefore, the elastic modulus of a network with density 10.64 mg/mL should be 0.25 MPa. This is close to the value of 0.27 MPa obtained from simulation (8% error). However, the simulation fails to reproduce the strain hardening observed in the experimental stress-strain curve during the plateau regime.

Figures 5.54 and 5.55 show the variation of potential energy of the system and its normalized components respectively as a function of strain. In the elastic regime, the potential energy of the system increases as shown in the inset to figure 5.54. After that it becomes constant and eventually decreases, indicating the onset of the plastic deformation regime. In the linear regime, the bending component of the potential energy is highest and therefore, nanotube bending governs the elastic response of the CNT network. At the beginning of the plateau regime, the bending energy component starts to decrease while the VDW energy component increases. The VDW energy component eventually becomes larger than bending energy component and therefore, VDW interactions between



Figure 5.55: Variation of potential energy components of CNT network with strain

nanotubes govern the plastic response of the CNT network. The contribution of axial deformation to the potential energy of the system is insignificant.

Figure 5.56 shows the variation in the components of stress as a function of strain. In a state of zero strain, there are compressive stresses in the network due to VDW interactions which are balanced by tensile axial stresses in the nanotubes. The magnitudes of both VDW and axial stresses initially decrease with strain. But however, since the stresses act in opposite directions and VDW stresses decrease at a much faster rate, the total stress in the system is tensile and increases linearly with strain, giving a small stiffness to the network against compressive loading in the initial elastic regime. Eventually both VDW and axial stresses flatten out resulting in a plateau regime. On increasing the strain further, the axial stresses in the system are initially small but become comparable to VDW and axial stresses at large strain. The thermal component of stress also becomes significant at large strain since the volume of the system decreases by a factor of 3.7 at 76% strain. The variation of VDW, axial and bending components of stress is studied in



Figure 5.56: Variation of stress components of CNT network with strain

detail so that the deformation mechanism of the CNT network can be deduced.

The orientation of nanotube bundles in the initial network configuration is isotropic as shown in figure 5.57a which exhibits the distribution of the orientation angle of nanotubes with respect to the z-axis. The number of nanotubes scales with angle as a sine function which confirms the isotropic distribution of nanotube bundles. It should be noted that the deviation near 90 degrees is due to finite system size effects. However, as the strain on the network is increased, the distribution of orientation angles becomes skewed and no longer follows a sine curve as shown in figure 5.57b. With further increase in strain, more and more nanotube bundles are orienting themselves in directions perpendicular to the z-axis along which the strain is applied.

Figure 5.58a shows the distribution of average axial stress in nanotubes with respect to their binned orientation angle. The average axial stress corresponding to an angle bin is calculated by dividing the sum of the axial stresses of all the nanotubes in that bin by the number of nanotubes. The average axial stress decreases with an increase in the orientation angle. However, since the number of nanotubes increases with orientation angle, the sum of axial stresses of nanotubes in a particular angle bin varies non monotonically with the orientation angle as shown in figure 5.58b. At 5% strain, the nanotubes with



Figure 5.57: Distribution of orientation angle of nanotubes with respect to the compressive axis in CNT network as function of strain



Figure 5.58: Distribution of average and total axial stress over orientation angle in CNT network as function of strain (bin size = 10 degrees)


Figure 5.59: Distribution of average and total VDW stress over orientation angle in CNT network as function of strain (bin size = 10 degrees)

an orientation angle of  $30^{\circ}$  contribute the most to the total axial stress which can be calculated by summing over all bins. As the strain is increased to 30%, the contribution of nanotubes with smaller orientation angles decreases since the fraction of such nanotubes also decreases with strain as shown in figure 5.57b. However, the contribution of nanotubes with larger orientation angles still remains the same despite the relative increase in their number. This is because the average stress on nanotubes with large orientation angles doesn't increase significantly at 30% strain as shown in figure 5.58a. Therefore, the total axial stress at 30% strain is lower than the stress at 5% strain. Basically at smaller strains, nanotubes bundles release stress by reorienting themselves in directions perpendicular to the direction of applied strain via bending of nanotubes. But at larger strain (>50\%), the average axial stress on nanotubes even with large orientation angles increases as shown in figure 5.58a. Therefore, their contribution to the total axial stress also increases as shown in figure 5.58b. Thus, the total axial stresses in the nanotubes start increasing resulting in the densification regime.

Figure 5.59a shows the distribution of average VDW stress in nanotubes with respect to their binned orientation angle. The average VDW stress is calculated in a similar



Figure 5.60: Deformation of a single nanotube at finite strains in CNT network (Color bar and orientation tripod for each figure is same)



(a) Top view



(b) Front view

Figure 5.61: Top and front views of deformed CNT network at 76% strain (Color bar for each figure is same)

manner as the average axial stress. The VDW stress is negative, which means it is compressive in nature. Its magnitude decreases with an increase in the orientation angle. At angles greater than  $80^{\circ}$ , it takes a small positive value, suggesting that the nanotubes which lie in the x-y plane, have small, tensile VDW stresses acting on them. This curve becomes steeper with strain. However, since the number of nanotubes increases with orientation angle, the sum of VDW stresses of nanotubes in a particular angle bin also varies non monotonically with the orientation angle as shown in figure 5.59b. As the strain is increased from 5% to 30%, the total VDW stress, which can be calculated by summing over all bins, decreases. This can be explained in a similar way as the decrease in total axial stress. However, since VDW stresses are compressive in nature, a net small tensile stress develops in the system when nanotube bundles reorient themselves in directions perpendicular to the direction of applied compressive strain via bending. At 60% strain, the average VDW stress on nanotubes with large orientation angles becomes large enough to compensate for the decrease in total VDW stress carried by nanotubes with smaller orientation angles. Therefore, the total VDW stress increases with strain. At very large strain (>70%), the fraction of nanotubes with orientation angle greater than  $80^{\circ}$  becomes so high that their large positive contribution results in the decrease of total VDW stresses in the system. Therefore, at large strain, the increasing number of nanotubes oriented perpendicular to the applied strain, which have VDW stresses of tensile nature, leads to an additional increase in overall stress in the densification regime.

The bending stresses in the system are negative and increase with an increase in strain as shown in figure 5.56. In order to explain the compressive nature of these bending stresses, we look at the deformation of a single nanotube in a bundle. Figure 5.60 shows snapshots of deformation of a single nanotube at several strains. A single nanotube deforms by forming bending loops along the x, y and z axes. The bending loops formed along the x and y axes have bending stresses of compressive nature (shown in blue) while the bending loops formed along the z axis have bending stresses of tensile nature (shown in red). Figure 5.61 shows the top and front views of the deformed nanotube network at a strain of 76%. In these figures, there can be seen more bending loops oriented along the x and y axes than along the z axis. This observation is supported by the distribution of bending stresses among the nanotube atoms shown in figure 5.62. This distribution



Figure 5.62: Distribution of bending stresses among nanotubes in CNT network

is skewed towards the left. This explains why the total bending stress in the system is negative. Its magnitude increases with strain as the amount of bending deformation increases.

#### 5.3.3 Revisiting hypothesis

Table 5.2 draws a comparison between experiments and simulation by listing several parameters characterizing the large deformation behavior of a CNT network. The meso-scale simulation successfully captures the three deformation regimes associated with the large deformation behavior of a CNT network. The stress and strain values associated with the linear elastic regime are in good agreement with the experiments. The error of 8% in the value of elastic modulus obtained from the meso-scale simulation is much lower than the error of 67% obtained from the finite element model. This is because the structure of a CNT network obtained from self-assembly simulation is further from a simple random network of nanotubes assumed in the finite element simulation. The VDW junctions are formed by entanglements between multiple nanotubes instead of just

	Experiment[7]		Simulation	
Network density (mg/mL)	9.9		10.64	
Elastic Modulus (MPa)	0.21		0.27	
Deformation Regimes	Strain	Stress (MPa)	Strain	Stress (MPa)
Linear regime	0.09	0.02	0.1	0.027
Plateau regime	0.6	0.08	0.5	0.03
Densification regime	0.8	0.19	0.8	0.08

Table 5.2: Parameters characterizing the large deformation behavior of a CNT network

two. Likewise a single nanotube can be a part of multiple bundles of nanotubes. The strain limits of the plateau and the densification regimes are also accurately captured in the meso-scale simulation. However, the BSM fails to model the strain hardening observed in experiments during the plateau regime. This is reflected in the different values of stresses in the plateau and the densification regimes. This can be attributed to the degree of bundling in the network as explained below. Apart from this discrepancy, the meso-scale simulation accurately reproduces the stress-strain curve of a CNT network, thereby offering insights into the governing deformation mechanisms and also capturing the key features of the linear elastic regime. This validates our hypothesis that a mesoscale simulation can model the large deformation behavior of a CNT network accurately including the three deformation regimes.

In chapter 1 of this study, plausible governing mechanisms for the large deformation behavior of CNT aerogels are mentioned and summarized in figure 1.2. In light of the insights offered by meso-scale simulation, the governing deformation mechanisms are revised and are summarized in figure 5.63. A distinction is made between the contribution of these deformation mechanisms to the system's potential energy, which dictates the elastic/plastic response of the network, and their respective manifestations into stresses, which drives the stress-strain response of the network. Under small compressive strains, the bundles of nanotubes start realigning themselves perpendicular to the direction of applied strain via bending at junctions. The bending energy of the system increases more than the decrease in the VDW energy of the system. As a result, the response of the



Figure 5.63: Governing mechanisms for the large deformation behavior of a CNT network

network under small compressive strains is elastic governed by the bending of nanotubes. But since, bending is a softer mode of deformation, the bending stresses in the network are small. The realignment of nanotubes decreases the magnitudes of compressive VDW stresses and to a lesser extent tensile axial stresses in the network, generating a net tensile stress in the network against the applied small compressive strain. Therefore, in the linear elastic regime, the stress response of the material is driven by the VDW interactions between nanotubes. As the compressive strain is increased, the decrease in the system's VDW energy becomes greater than the increase in the bending energy. Therefore, past a critical strain, the response of the material becomes plastic governed by the VDW interactions between nanotubes. As the strain induced alignment of nanotubes continues, the decrease in the number of nanotubes with small orientation angles carrying large stresses is balanced by an increase in the average stresses on nanotubes with large orientation angles. Therefore, the net VDW and axial stresses in the system become constant causing the overall stress response of the network to plateau. At large strain, the average axial stresses on nanotubes with large orientation angles become large enough, causing the total axial stresses in the network to increase, leading to a densification regime. The steep rise in the value of stress in the densification regime is also caused by the tensile VDW stresses acting on the nanotubes with orientation angle greater than 80 degrees at large strain.

It can be expected that in a network with a smaller degree of bundling and more isolated nanotubes, there will be an additional volume constraint for realignment of nanotubes. Thus the average tensile stresses on nanotubes with large orientation angles will increase at a faster rate, leading to a strain hardening in the plateau regime. Since, the degree of bundling is also dictated by the form of the LJ potential to some extent, employment of a different form of the LJ potential in the meso-scale model could reproduce the strain hardening.

#### 5.4 Conclusion

In this chapter, a bead spring model is developed for simulating a network of carbon nanotubes. This model is capable of capturing the axial and bending deformation of carbon nanotubes as well as VDW interactions between them. Several type of LJ potentials are employed to model the VDW interactions. It is found that LJ potentials that restrict nanotube sliding generate a network structure similar to CNT aerogels whereas LJ potentials that allow free sliding of nanotubes generate a bundled network of nanotubes during the self-assembly process. However, the stiffness of the network, whose mechanical behavior is modeled using a LJ potential of the first kind, is found to be two orders of magnitude higher than that of actual CNT aerogels. The elastic regime also shrinks by a considerable amount in this case. Therefore, it is essential to allow the free sliding of nanotubes during deformation of a CNT network.

The stress-strain curve of the CNT network, generated using the LJ potential that allows free sliding of nanotubes, is similar to the stress-strain curve of a CNT aerogel both in terms of the presence of the three deformation regimes as well as the stress and strain values associated with the linear elastic regime. The error in the value of the elastic modulus calculated from this simulation is 8% when compared with experiments. This validates the accuracy of such meso-scale simulations in modeling the large deformation behavior of pristine CNT networks. A deformation mechanism is presented based on the insights offered by this simulation. It is found that the elastic-plastic response of CNT networks is dictated by an interplay between the bending energy of nanotubes and the VDW interaction energy of nanotubes. The elasticity of CNT networks is governed by the bending deformation of nanotubes while the plastic deformation of CNT networks is governed by the VDW interactions between nanotubes. The stress response of the material in the elastic regime is dictated by the VDW stresses on nanotubes while in the plateau and densification regimes, both the VDW and axial deformation stresses in the material drive the overall stress response.

#### Chapter 6

## Summary and Future Work

#### 6.1 Summary

The two major goals of this study are to quantify the elastic behavior of a random network of carbon nanotubes (CNTs) as a function of its density and junction properties and to study the large deformation behavior of a pristine carbon nanotube network and identify the underlying deformation mechanisms. In order to realize these goals, a multiscale modeling approach is employed.

Molecular dynamics (MD) simulations are used at the atomic scale to model the mechanical properties of individual nanotubes and to study the van der Waals (VDW) interactions between a group of nanotubes. The self-assembly and large deformation behavior of a pristine carbon nanotube network are studied using meso-scale simulations. In these simulations, a carbon nanotube is modeled as group of beads connected via harmonic springs that resist axial and bending deformations. The beads are allowed to interact via a Lennard Jones (LJ) potential to model van der Waals (VDW) interactions between them. The parameters of the bead spring model are calculated using results from the atomistic simulations of individual nanotubes. As a result of coarse graining, this model can capture the complex physics associated with carbon nanotubes networks while avoiding the large computation cost of traditional MD simulations. A continuum scale finite element model is used to carry out high throughput simulations characterizing the elastic behavior of a random network of nanotubes over a large parameter space. In this model, a nanotube is modeled as an elastic beam element while the junctions between nanotubes are modeled as spring elements that resist translation and rotation along the x, y and z axes. The parameters that are considered in this study are the network density, and the translational and the torsional stiffnesses of the junctions.

The first major finding of the finite element study is that that the deformation of a random nanotube network (both 2D and 3D) can be governed either by the deformation of the nanotubes (bending, axial compression) or deformation of the junctions (translation of junctions in the x, y or z directions). The junction stiffness parameters in combination with the network density determine the network deformation mode. The results of this study are also applicable to any stiff fiber network. The parameter maps obtained from this study provide us with a set of rules that can be applied to predict the elastic

deformation behavior of a random fibrous network with any set of junction stiffnesses and network density. The rotation of nanotubes at junctions doesn't play a significant role in the overall deformation of the network under compression. The two mechanisms of junction deformation (translation along the local x & y directions and local z direction) act as springs connected in series and parallel for low and high network densities respectively. The two mechanisms of nanotube junction translation (along the local x & y directions) and nanotube stretching/compression act as springs connected in series. The two mechanisms of nanotube junction translation (along the local x and nanotube bending can also be viewed as springs connected in series.

The second major finding of finite element study is that the network deformation is significantly different from the predictions of affine theory, especially at smaller network densities, when deformation is governed by nanotube bending. The network elastic modulus never reaches the affine prediction even at large network densities. In the finite element model, the elastic behavior of a pristine carbon nanotube network is also studied by modeling the junctions mediated by VDW interactions between nanotubes as harmonic springs. The stiffnesses of these springs are calculated from MD simulations. The finite element model predicts the same scaling behavior of the network elastic modulus with density as obtained in experiments. However, it produces an error of 67% in the actual value of elastic modulus when compared with experiments. This discrepancy is attributed to the possibility that the structure of a CNT aerogel can be different from a simple random fibrous network. This is later confirmed by the results of self-assembly simulations carried out using a meso-scale model.

One of the main findings of the meso-scale simulations is the network structure obtained from the self-assembly simulations, which is found to be very different from a simple random network of nanotubes. The self-assembled network contains large bundles of nanotubes entangled at junctions between them. It is significant that this structure is different from the structures of CNT aerogels reported in the experimental studies. The CNT aerogels reported in these studies contained mostly isolated nanotubes which gave them a large specific surface area. Nanotube networks with similarly large specific surface areas are generated from self-assembly simulations employing a LJ potential that hinders free sliding of nanotubes. However, the stiffness of such networks, whose mechanical behavior is also modeled using the same kind of LJ potential, is found to be two orders of magnitude higher than that of actual CNT aerogels. The elastic regime also shrinks by a considerable amount in this case. Therefore, it is essential to allow the free sliding of nanotubes during deformation of a CNT network.

The stress-strain curve of the CNT network, generated using the LJ potential that allows free sliding of nanotubes, is similar to the experimental stress-strain curve of a CNT aerogel both in terms of the presence of the three deformation regimes as well as the stress and strain values associated with the linear elastic regime. The error in the value of the elastic modulus calculated from this simulation is 8% when compared with experiments. This validates the accuracy of such meso-scale simulations in modeling the large deformation behavior of pristine CNT networks. The other major finding of this study is the deformation mechanism of the CNT networks which is deduced from the insights offered from these simulations. It is found that the elastic-plastic response of CNT networks is dictated by an interplay between the bending energy of nanotubes and the VDW interaction energy of nanotubes. The elasticity of CNT networks is governed by the bending deformation of nanotubes while the plastic deformation of CNT networks is governed by the VDW interactions between nanotubes. The stress response of the material in the elastic regime is dictated by the VDW stresses on nanotubes while in the plateau and densification regimes, both the VDW and axial deformation stresses in the material drive the overall stress response.

#### 6.2 Future Work

The next major extension of this work can be towards modeling the self-assembly and mechanical behavior of CNT networks with modified junction properties such as graphene coated and covalent cross-linked. Experimental studies have shown significant improvement in the mechanical properties of CNT aerogels with modified junction properties. It will be interesting to study how these changes to the junctions affect the underlying deformation mechanisms and thereby the mechanical properties of such materials. Such studies can be conducted via meso-scale simulations. They will involve building additional coarse-graining models for graphene and polymer chains. The process of equilibrating the initial structure during the self-assembly simulations will also play an important role in deciding the mechanical properties of the final structure. Since such networks are also shown to have superelastic properties, the effect of cyclic loading and mechanisms behind energy dissipation during the loading-unloading cycle can form an interesting study in itself.

The second extension of this work can be towards improving the current bead spring model developed for studying the pristine network of nanotubes. A better model will result in less error in predicting the elastic modulus of the network and will also be able to capture the strain hardening associated with the plateau regime. One improvement can be employing a 10-4 LJ potential to model the large deformation behavior of the CNT network, since this potential has the same functional form as an actual tube-tube potential and doesn't result in any corrugation artifacts. The major challenge in this case will be reducing the large computation time associated with this potential. It can be done by developing a parallel implementation of this potential.

The third and probably the most challenging extension of this study can be towards developing a continuum model informed by the results from the meso-scale simulations. Meso-scale simulations, although faster than traditional MD simulations, are still orders of magnitude slower than the continuum simulations. It is impossible to perform high throughput studies using the meso-scale simulations alone. However, superelements that represent CNT networks more accurately, in terms of the underlying constitutive equation, than traditional beam or spring elements can be created using insights from meso-scale simulations. Such continuum scale simulations will provide a huge advantage in terms of both cost and accuracy.

#### Bibliography

- Stephane Evoy Massimiliano Ventra and James R. Heflin. Introduction to Nanoscale Science and Technology. Springer US, New York, 1 edition, 2004.
- [2] Boris I. Yakobson and Phaedon Avouris. Mechanical Properties of Carbon Nanotubes, pages 287–327. Springer Berlin Heidelberg, Berlin, Heidelberg, 2001.
- [3] Hao Chen, Ajit Roy, Jong-Beom Baek, et al. Controlled growth and modification of vertically-aligned carbon nanotubes for multifunctional applications. *Materials Science and Engineering: R: Reports*, 70(36):63 – 91, 2010. 3rd {IEEE} International NanoElectronics Conference (INEC).
- [4] Mei Zhang, Ken R. Atkinson, and Ray H. Baughman. Multifunctional Carbon Nanotube Yarns by Downsizing an Ancient Technology. *Science*, 306(5700):1358–1361, 2004.
- Rajib K. Das, Bo Liu, John R. Reynolds, and Andrew G. Rinzler. Engineered Macroporosity in Single-Wall Carbon Nanotube Films. *Nano Letters*, 9(2):677–683, 2009.
   PMID: 19170555.
- [6] M.B. Bryning, D.E. Milkie, M.F. Islam, et al. Carbon Nanotube Aerogels. Advanced Materials, 19(5):661–664, 2007.
- [7] Kyu Hun Kim, Youngseok Oh, and Mohammad F. Islam. Mechanical and Thermal Management Characteristics of Ultrahigh Surface Area Single-Walled Carbon Nanotube Aerogels. Advanced Functional Materials, 23(3):377–383, 2013.

- [8] Xuchun Gui, Jinquan Wei, Kunlin Wang, et al. Carbon Nanotube Sponges. Advanced Materials, 22(5):617–621, 2010.
- [9] Oh Youngseok Kim Kyu Hun and F. IslamM. Graphene coating makes carbon nanotube aerogels superelastic and resistant to fatigue. *Nat Nano*, 7(9):562–566, Sep 2012.
- [10] L. J. Gibson and M. F. Ashby. Cellular Solids, Strucutre and properties. Pergamon, New York, 1997.
- [11] Anyuan Cao, Pamela L. Dickrell, W. Gregory Sawyer, Mehrdad N. Ghasemi-Nejhad, and Pulickel M. Ajayan. Super-Compressible Foamlike Carbon Nanotube Films. *Science*, 310(5752):1307–1310, 2005.
- [12] E. K. Hobbie, D. O. Simien, J. A. Fagan, et al. Wrinkling and Strain Softening in Single-Wall Carbon Nanotube Membranes. *Phys. Rev. Lett.*, 104:125505, Mar 2010.
- [13] Richard A. Vaia and H. Daniel Wagner. Framework for nanocomposites. Materials Today, 7(11):32 – 37, 2004.
- [14] Yunpu Zhai, Yuqian Dou, Dongyuan Zhao, et al. Carbon Materials for Chemical Capacitive Energy Storage. Advanced Materials, 23(42):4828–4850, 2011.
- [15] Ali E. Aliev, Jiyoung Oh, Mikhail E. Kozlov, et al. Giant-Stroke, Superelastic Carbon Nanotube Aerogel Muscles. *Science*, 323(5921):1575–1578, 2009.
- [16] R. C. Picu. Mechanics of random fiber networks-a review. Soft Matter, 7:6768–6785, 2011.
- [17] Claus Heussinger and Erwin Frey. Role of architecture in the elastic response of semiflexible polymer and fiber networks. *Phys. Rev. E*, 75:011917, Jan 2007.
- [18] Jan A. Åström, P. B. Sunil Kumar, Ilpo Vattulainen, and Mikko Karttunen. Strain hardening, avalanches, and strain softening in dense cross-linked actin networks. *Phys. Rev. E*, 77:051913, May 2008.

- [19] P. R. Onck, T. Koeman, T. van Dillen, and E. van der Giessen. Alternative Explanation of Stiffening in Cross-Linked Semiflexible Networks. *Phys. Rev. Lett.*, 95:178102, Oct 2005.
- [20] David A. Head, Alex J. Levine, and F. C. MacKintosh. Deformation of Cross-Linked Semiflexible Polymer Networks. *Phys. Rev. Lett.*, 91:108102, Sep 2003.
- [21] D. A. Head, A. J. Levine, and F. C. MacKintosh. Distinct regimes of elastic response and deformation modes of cross-linked cytoskeletal and semiflexible polymer networks. *Phys. Rev. E*, 68:061907, Dec 2003.
- [22] H. Hatami-Marbini and R. C. Picu. Scaling of nonaffine deformation in random semiflexible fiber networks. *Phys. Rev. E*, 77:062103, Jun 2008.
- [23] Gavin A. Buxton and Nigel Clarke. "Bending to Stretching" Transition in Disordered Networks. *Phys. Rev. Lett.*, 98:238103, Jun 2007.
- [24] Jan Wilhelm and Erwin Frey. Elasticity of Stiff Polymer Networks. Phys. Rev. Lett., 91:108103, Sep 2003.
- [25] Knut M Heidemann, Abhinav Sharma, Florian Rehfeldt, Christoph F Schmidt, and Max Wardetzky. Elasticity of 3D networks with rigid filaments and compliant crosslinks. *Soft matter*, 11(2):343–354, 2015.
- [26] H Corte and OJ Kallmes. Statistical geometry of a fibrous network. In *The Formation and Structure of Paper*, pages 13–46. 1962.
- [27] C Fellers, H Andersson, and H Hollmark. The definition and measurement of thickness and density. *Paper Structure and Properties. New York: Marcel Dekker*, 151, 1986.
- [28] RC Picu and H Hatami-Marbini. Long-range correlations of elastic fields in semiflexible fiber networks. *Computational Mechanics*, 46(4):635–640, 2010.
- [29] M Latva-Kokko and J Timonen. Rigidity of random networks of stiff fibers in the low-density limit. *Physical Review E*, 64(6):066117, 2001.

- [30] CW Wang, L Berhan, and AM Sastry. Structure, Mechanics and Failure of Stochastic Fibrous Networks: Part1?Microscale Consideration. ASME J. Eng. Mater. Technol./CrossRef/, 122:450-459, 2000.
- [31] M. Rubinstein and R. Colby. *Polymer Physics*. Oxford Univ. Press, Oxford, 2003.
- [32] ED Siggia, C Bustamante, JF Marko, and SB Smith. Entropic elasticity of λ-phage DNA. Science, 265:1599–1600, 1994.
- [33] Ingo Schwaiger, Michael Schleicher, Angelika A Noegel, and Matthias Rief. The folding pathway of a fast-folding immunoglobulin domain revealed by single-molecule mechanical experiments. *EMBO reports*, 6(1):46–51, 2005.
- [34] PD Wu and E Van der Giessen. On improved 3-D non-Gaussian network models for rubber elasticity. *Mechanics research communications*, 19(5):427–433, 1992.
- [35] PD Wu and E Van der Giessen. On improved network models for rubber elasticity and their applications to orientation hardening in glassy polymers. *Journal of the Mechanics and Physics of Solids*, 41(3):427–456, 1993.
- [36] Mary C Boyce and Ellen M Arruda. Constitutive models of rubber elasticity: a review. Rubber chemistry and technology, 73(3):504–523, 2000.
- [37] AJ Staverman. Properties of phantom networks and real networks. In *Polymer Networks*, pages 73–101. Springer, 1982.
- [38] B. Erman and J. E. Mark. Structures and Properties of Rubber-like Networks. Oxford Univ. Press, Oxford, 1997.
- [39] JA Åström, JP Mäkinen, Mikko J Alava, and J Timonen. Elasticity of Poissonian fiber networks. *Physical Review E*, 61(5):5550, 2000.
- [40] Klaus Kroy and Erwin Frey. Force-extension relation and plateau modulus for wormlike chains. *Physical review letters*, 77(2):306, 1996.

- [41] Erwin Frey, Klaus Kroy, Jan Wilhelm, and Erich Sackmann. Statistical mechanics of semiflexible polymers: theory and experiment. In *Dynamical Networks in Physics* and Biology, pages 103–119. Springer, 1998.
- [42] F Gittes and FC MacKintosh. Dynamic shear modulus of a semiflexible polymer network. *Physical Review E*, 58(2):R1241, 1998.
- [43] H Isambert and AC Maggs. Dynamics and rheology of actin solutions. Macromolecules, 29(3):1036–1040, 1996.
- [44] Bernard Hinner, Markus Tempel, Erich Sackmann, Klaus Kroy, and E Frey. Entanglement, elasticity, and viscous relaxation of actin solutions. *Physical Review Letters*, 81(12):2614, 1998.
- [45] David C Morse. Viscoelasticity of tightly entangled solutions of semiflexible polymers. *Physical Review E*, 58(2):R1237, 1998.
- [46] Matteo Pasquali, V Shankar, and David C Morse. Viscoelasticity of dilute solutions of semiflexible polymers. *Physical Review E*, 64(2):020802, 2001.
- [47] FC MacKintosh, J Käs, and PA Janmey. Elasticity of semiflexible biopolymer networks. *Physical review letters*, 75(24):4425, 1995.
- [48] Yi-Chia Lin, Norman Y Yao, Chase P Broedersz, et al. Origins of elasticity in intermediate filament networks. *Physical review letters*, 104(5):058101, 2010.
- [49] Claus Heussinger and Erwin Frey. Floppy modes and nonaffine deformations in random fiber networks. *Physical review letters*, 97(10):105501, 2006.
- [50] BA DiDonna and TC Lubensky. Nonaffine correlations in random elastic media. *Physical Review E*, 72(6):066619, 2005.
- [51] CP Broedersz, C Storm, and FC MacKintosh. Effective-medium approach for stiff polymer networks with flexible cross-links. *Physical Review E*, 79(6):061914, 2009.

- [52] A Sharma, M Sheinman, KM Heidemann, and FC MacKintosh. Elastic response of filamentous networks with compliant crosslinks. *Physical Review E*, 88(5):052705, 2013.
- [53] CP Broedersz, C Storm, and FC MacKintosh. Nonlinear elasticity of composite networks of stiff biopolymers with flexible linkers. *Physical review letters*, 101(11):118103, 2008.
- [54] Dominique Poquillon, Bernard Viguier, and Eric Andrieu. Experimental data about mechanical behaviour during compression tests for various matted fibres. *Journal of materials science*, 40(22):5963–5970, 2005.
- [55] JP Masse, L Salvo, D Rodney, Y Bréchet, and O Bouaziz. Influence of relative density on the architecture and mechanical behaviour of a steel metallic wool. *Scripta materialia*, 54(7):1379–1383, 2006.
- [56] LA Girifalco, Miroslav Hodak, and Roland S Lee. Carbon nanotubes, buckyballs, ropes, and a universal graphitic potential. *Physical Review B*, 62(19):13104, 2000.
- [57] Evgeny G Pogorelov, Alexander I Zhbanov, Yia-Chung Chang, and Sung Yang. Universal curves for the van der Waals interaction between single-walled carbon nanotubes. *Langmuir*, 28(2):1276–1282, 2011.
- [58] Alexander I Zhbanov, Evgeny G Pogorelov, and Yia-Chung Chang. Van der Waals interaction between two crossed carbon nanotubes. Acs Nano, 4(10):5937–5945, 2010.
- [59] Steve Plimpton. Fast parallel algorithms for short-range molecular dynamics. Journal of computational physics, 117(1):1–19, 1995.
- [60] LAMMPS Molecular Dynamics Simulator, http://lammps.sandia.gov.
- [61] Steven J Stuart, Alan B Tutein, and Judith A Harrison. A reactive potential for hydrocarbons with intermolecular interactions. *The Journal of chemical physics*, 112(14):6472–6486, 2000.

- [62] Weisstein, Eric W. "Sphere Point Picking." From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/SpherePointPicking.html.
- [63] Susanne Heyden. Network Modelling for Evaluation of Mechanical Properties of Cellulose Fibre Fluff, volume 1011. Lund University, 2000.
- [64] Robert D Cook et al. Concepts and applications of finite element analysis. John Wiley & Sons, 2007.
- [65] P Papanikos, DD Nikolopoulos, and KI Tserpes. Equivalent beams for carbon nanotubes. Computational Materials Science, 43(2):345–352, 2008.
- [66] KI Tserpes and P Papanikos. Finite element modeling of single-walled carbon nanotubes. Composites Part B: Engineering, 36(5):468–477, 2005.
- [67] V-D Nguyen, Eric Béchet, Christophe Geuzaine, and Ludovic Noels. Imposing periodic boundary condition on arbitrary meshes by polynomial interpolation. *Computational Materials Science*, 55:390–406, 2012.
- [68] Weidong Wu, Joseph Owino, Ahmed Al-Ostaz, and Liguang Cai. Applying Periodic Boundary Conditions in Finite Element Analysis. In SIMULIA community conference, providence, RI, pages 707–719, 2014.
- [69] Harald Berger, Sreedhar Kari, Ulrich Gabbert, et al. An analytical and numerical approach for calculating effective material coefficients of piezoelectric fiber composites. International Journal of Solids and Structures, 42(21):5692–5714, 2005.
- [70] Zihui Xia, Chuwei Zhou, Qiaoling Yong, and Xinwei Wang. On selection of repeated unit cell model and application of unified periodic boundary conditions in micromechanical analysis of composites. *International Journal of Solids and Structures*, 43(2):266–278, 2006.
- [71] JM Tyrus, M Gosz, and E DeSantiago. A local finite element implementation for imposing periodic boundary conditions on composite micromechanical models. *International journal of solids and structures*, 44(9):2972–2989, 2007.

- [72] Qing-Hua Qin and Hui Wang. Matlab and C programming for Trefftz finite element methods. CRC Press, 2008.
- [73] A Concise Sparse Matrix Package in C, Timothy A. Davis, http://people.sc.fsu.edu/jburkardt/c<sub>s</sub>rc/csparse/csparse.html.
- [74] Sparse Triplet to Compressed Column Conversion, http://people.sc.fsu.edu/ jburkardt/ $c_s rc/st_t o_c c/st_t o_c c.html$ .
- [75] Baptiste Depalle, Zhao Qin, Sandra J Shefelbine, and Markus J Buehler. Influence of cross-link structure, density and mechanical properties in the mesoscale deformation mechanisms of collagen fibrils. *Journal of the mechanical behavior of biomedical materials*, 52:1–13, 2015.
- [76] Navid Kazem, Carmel Majidi, and Craig E Maloney. Gelation and mechanical response of patchy rods. Soft matter, 11(40):7877–7887, 2015.
- [77] Steven W Cranford and Markus J Buehler. In silico assembly and nanomechanical characterization of carbon nanotube buckypaper. *Nanotechnology*, 21(26):265706, 2010.
- [78] Bo Xie, Yilun Liu, Yiting Ding, Quanshui Zheng, and Zhiping Xu. Mechanics of carbon nanotube networks: microstructural evolution and optimal design. Soft Matter, 7(21):10039–10047, 2011.
- [79] Markus J Buehler. Mesoscale modeling of mechanics of carbon nanotubes: self-assembly, self-folding, and fracture. *Journal of materials research*, 21(11):2855–2869, 2006.
- [80] Igor Ostanin, Roberto Ballarini, David Potyondy, and Traian Dumitric. A distinct element method for large scale simulations of carbon nanotube assemblies. Journal of the Mechanics and Physics of Solids, 61(3):762–782, 2013.
- [81] Leonid V Zhigilei, Chenyu Wei, and Deepak Srivastava. Mesoscopic model for dynamic simulations of carbon nanotubes. *Physical review B*, 71(16):165417, 2005.

- [82] Alexey N Volkov, KR Simov, and Leonid V Zhigilei. Mesoscopic simulation of selfassembly of carbon nanotubes into a network of bundles. In *Proceedings of the 47th* AIAA Aerospace Sciences Meeting, AIAA paper, volume 1544, page 2009, 2009.
- [83] Yuezhou Wang, Igor Ostanin, Cristian Gaida?u, and Traian Dumitric? Twisting carbon nanotube ropes with the mesoscopic distinct element method: geometry, packing, and nanomechanics. *Langmuir*, 31(45):12323–12327, 2015.
- [84] Alexey N Volkov and Leonid V Zhigilei. Structural stability of carbon nanotube films: the role of bending buckling. ACS nano, 4(10):6187–6195, 2010.
- [85] T Schneider and E Stoll. Molecular-dynamics study of a three-dimensional one-component model for distortive phase transitions. *Physical Review B*, 17(3):1302, 1978.
- [86] Scott N Schiffres, Kyu Hun Kim, Lin Hu, et al. Gas Diffusion, Energy Transport, and Thermal Accommodation in Single-Walled Carbon Nanotube Aerogels. Advanced Functional Materials, 22(24):5251–5258, 2012.

## Appendix A

## List of important variables

$l_c$	Mean filament segment length
$ ho_l$	Network filament density
$ ho_n$	Network junction density
l	Length of filament
D	Diameter of filament
$l_b$	Square root of the ratio of bending to axial stiffness of
	the filament
d	Distance between two nanotubes
$\gamma$	Angle between two nanotubes
$\phi$	Potential energy between two nanotubes
$d_o$	Equilibrium distance between two nanotubes
$\phi_o$	Potential energy between two nanotubes at the
	equilibrium distance
la / la / la	Stiffness of springs that resist translation along $x, y$
$n_x/n_y/n_z$	and $z$ directions of nanotube junctions
$k_{m,k_{m}}/k_{m,k_{m}}/k_{m,k_{m}}$	Stiffness of springs that resist rotation along $x$ , $y$ and $z$
"rotx/"roty/"rotz	directions of nanotube junctions
fman	Maximum force that results in fracture of springs along
Jmax	x and $y$ directions
L	Side length of simulation unit cell
$n_{c2D}/n_{c3D}$	Number of junctions per nanotube in a 2D and 3D $$
	network
COD	Cut-off distance for junction formation between two
	nanotubes
$Y_{network}$	Elastic modulus of the carbon nanotube network
$\epsilon_y$	Compressive strain on carbon nanotube network in
	finite element simulation
$\sigma_y$	Compressive stress on carbon nanotube network in
	finite element simulation

F .	Elastic energy stored in axial deformation of nanotubes
$E_{stretch}$	during small deformation of network
<i>F</i>	Elastic energy stored in bending deformation of
Dend	nanotubes during small deformation of network
$E_{kx}/E_{ky}/E_{kz}$ $E_{krotx}/E_{kroty}/E_{krotz}$ RMS	Elastic energy stored in translation of junctions during
	small deformation of network
	Elastic energy stored in rotation of junctions during
	small deformation of network
	Root mean square deviation of displacement of each
	node in network from the affine prediction
COREL	Standard correlation function between the vertical
	displacement and initial position of each node
NA	Non-affinity measure of the network deformation
V	Elastic modulus of the carbon nanotube network as
Yaffine	predicted from affine theory
$E_{system}$	Total potential of the carbon nanotube network
$E_{axial}$	Axial deformation energy of nanotubes
$E_{LJ}$	Van der Waals interaction energy of nanotubes
k	Stiffness of spring that resists axial deformation in
$\kappa_{axial}$	bead spring model
k <sub>bend</sub>	Stiffness of spring that resists bending deformation in
	bead spring model
σ	Distance parameter in Lennard Jones potential in bead
	spring model
$\epsilon$	Energy parameter in Lennard Jones potential in bead
	spring model
r,  heta	Distance and angle between beads in bead spring
	model
$r_o,  heta_o$	Equilibrium distance and angle between beads in bead
	spring model

$\Phi$	Bending angle of nanotube
$\Phi_c$	Critical bending angle of nanotube at which buckling
	occurs
z	Sliding distance between two nanotubes
8	Engineering stress on carbon nanotube network in
	bead-spring simulation
e	Engineering strain on carbon nanotube network in
	bead-spring simulation

## Appendix B

## LAMMPS Input Script

atom_style	atomic
units	metal
boundary	f f f
read_data	dataangle.cnt
group	tubel id $\leq 240$
group	tube2 id $> 241$ 480
group	idwy id 3 219 249 465
newton	on
pair_style	hybrid airebo $4.0\ 1\ 0\ lj/cut\ 10.2$
pair_coeff	* * airebo ./CH.airebo C C
pair_coeff	1 2 lj/cut 0.00284 3.40
neighbor	2.0 bin
neigh_modify	delay 5
dump	1 all atom 200 dump.cntvel
minimize	$1.0 \mathrm{e}{-6}$ $1.0 \mathrm{e}{-6}$ $1000$ $1000$
fix	$1 \ {\rm tube1} \ {\rm move} \ {\rm rotate} \ 0 \ 0 \ 0 \ 1 \ 0 \ 100$
fix	$2 \ {\rm tube}2 \ {\rm move}$ rotate $0 \ 2.916 \ 0 \ 0 \ -1 \ 0 \ 100$
fix_modify	1 energy yes
fix_modify	2 energy yes
thermo	200
dump	$2 \ {\rm idwy} \ {\rm custom} \ 200 \ {\rm dump1.cntvel} \ {\rm id} \ {\rm x} \ {\rm y} \ {\rm z}$
run	50000

Command	Explanation
atom_style	Defines attributes associated with atoms to be that of atomic style.
units	Sets units to be used in simulation as defined by metal style.
boundary	Sets boundary of global simulation box to be non periodic and fixed.
read_data	Reads ids, coordinates, mass and type of atoms from an input file.
group	Distributes atoms into two groups each representing an individual
	nanotube. Third group stores atoms coordinates of which are used to
	calculate angle between nanotubes at different time steps.
newton	Turns on Netwon's third law of motion for bonded and pairwise
	interactions. It means that if atoms involved in interactions are on two
	different processors than interaction is calculated by only one and
	communicated to other.
pair_style	Specifies a hybrid pair style that involves two kinds of potentials AIREBO
	and LJ/cut. For AIREBO style scale factor for cut-off distance is 4 and
	contribution from LJ term is turned on but from torsional term is turned
	off. For LJ/cut style cutoff distance is 10.2 Å.
pair_coeff	Defines AIREBO potential to be used for interactions among similar type
	of atoms 1-1 & 2-2 and LJ potential for different types 1-2.
neighbor	Defines the style and skin distance used for building neighbors list
neigh_modify	Specifies not to update neighbors list until at least 5 time steps
dump	Dumps atom ids and coordinates to a data file after certain time steps.
minimize	Minimizes energy of the system iteratively by adjusting atom coordinates
	until either force or energy between them is below a certain threshold.
fix move	Rotates all atoms in the group specified by an angle defined by the time
	period of rotation and direction specified by rotation axis and its point of
	origin.
fix_modify	Includes the contribution of potential energy in thermodynamic output.
thermo	Calculates and prints thermodynamic data after 200 time-steps
run	Runs the simulation for 50000 time-steps

Table B.1: Commands used in input script for LAMMPS

## Appendix C

#### **Elastic Modulus Curves**



## Appendix D

## Elastic Energy Curves



Appendix E

# Modified source code: Bending buckling

```
void AngleHarmonic::compute(int eflag, int vflag)
{
    int i1, i2, i3, n, type;
    double delx1, dely1, delz1, delx2, dely2, delz2;
    double eangle, f1[3], f3[3];
    double dtheta, tk, thetacr=(MY_PI-MY_PI/45), theta, dthetacr;
    double rsq1, rsq2, r1, r2, c, s, a, a11, a12, a22;
    eangle = 0.0;
    if (eflag || vflag) ev_setup(eflag, vflag);
    else evflag = 0;
    double **x = \text{atom} - x;
    double **f = atom \rightarrow f;
    int **anglelist = neighbor->anglelist;
    int nanglelist = neighbor->nanglelist;
    int nlocal = atom \rightarrow nlocal;
    int newton_bond = force->newton_bond;
    for (n = 0; n < nanglelist; n++) {
        i1 = anglelist[n][0];
        i2 = anglelist[n][1];
        i3 = anglelist[n][2];
        type = anglelist [n][3];
        // 1st bond
        delx1 = x[i1][0] - x[i2][0];
        dely1 = x[i1][1] - x[i2][1];
        delz1 = x[i1][2] - x[i2][2];
```
```
rsq1 = delx1*delx1 + dely1*dely1 + delz1*delz1;
r1 = sqrt(rsq1);
```

```
// 2nd bond
```

```
\begin{aligned} delx2 &= x[i3][0] - x[i2][0]; \\ dely2 &= x[i3][1] - x[i2][1]; \\ delz2 &= x[i3][2] - x[i2][2]; \end{aligned}
```

```
rsq2 = delx2*delx2 + dely2*dely2 + delz2*delz2;
r2 = sqrt(rsq2);
```

```
// angle (cos and sin)
```

```
if (c > 1.0) c = 1.0;
if (c < -1.0) c = -1.0;
```

s = sqrt(1.0 - c\*c);if (s < SMALL) s = SMALL; s = 1.0/s;

```
// force & energy
theta=acos(c);
dtheta = theta - theta0[type];
dthetacr = thetacr - theta0[type];
if(theta>=thetacr){
```

```
tk = -2.0 * k [type] * dtheta;
}
if (theta<thetacr) {
    tk = -2.0 * k[type] * dthetacr;
}
if (eflag){
    if (theta>=thetacr){
        eangle = -(tk*dtheta)/2.0;
    }
    if (theta<thetacr) {
        eangle = -tk * dtheta + (tk * dthetacr) / 2.0;
    }
}
a = tk * s;
a11 = a*c / rsq1;
a12 = -a / (r1 * r2);
a22 = a*c / rsq2;
f1[0] = a11*delx1 + a12*delx2;
f1[1] = a11 * dely1 + a12 * dely2;
f1[2] = a11*delz1 + a12*delz2;
f3[0] = a22*delx2 + a12*delx1;
f3[1] = a22*dely2 + a12*dely1;
f3[2] = a22*delz2 + a12*delz1;
// apply force to each of 3 atoms
if (newton_bond || i1 < nlocal) {
```

```
f[i1][0] += f1[0];
    f[i1][1] += f1[1];
    f[i1][2] += f1[2];
}
if (newton_bond || i2 < nlocal) {
    f[i2][0] -= f1[0] + f3[0];
    f[i2][1] = f1[1] + f3[1];
    f[i2][2] -= f1[2] + f3[2];
}
if (newton_bond || i3 < nlocal) {
    f[i3][0] += f3[0];
    f[i3][1] += f3[1];
    f[i3][2] += f3[2];
}
if (evflag) ev_tally(i1, i2, i3, nlocal, newton_bond, eangle, f1, f3,
                       delx1 , dely1 , delz1 , delx2 , dely2 , delz2 );
```

}

171

Appendix F

## Modified source code: LJ potential

```
void PairLJCut::compute(int eflag, int vflag)
{
    int i,j,ii,jj,inum,jnum,itype,jtype,dummy;
    double delxo, delyo, delzo, rsq, evdwl, fpair;
    double r2, r4, r10, forcelj, factor_lj;
    int *ilist ,*jlist ,*numneigh ,** firstneigh ;
    int dCG=10, atom1, atom2, atom3, atom4, indexi, indexj;
    double rcnt = 4.068000345, rcut = 10.2, rcutalp = pow(rcut, 10),
       rcutbet=pow(rcut,4), req=10, rmod, rred, energy, xo, yo, zo, xf,
       yf, zf;
    tagint firstatom = 1, lastatom = 20;
    double xtmpi[9], ytmpi[9], ztmpi[9], xtmpj[9], ytmpj[9], ztmpj
       [9], delx, dely, delz, xtmp, ytmp, ztmp;
    evdwl = 0.0;
    if (eflag || vflag) ev_setup(eflag, vflag);
    else evflag = vflag_fdotr = 0;
    double **x = \text{atom} - x;
```

```
double **f = atom->f;
int *type = atom->type;
int nlocal = atom->nlocal;
double *special_lj = force->special_lj;
int newton_pair = force->newton_pair;
tagint *id=atom->tag;
```

```
inum = list ->inum;
ilist = list ->ilist;
numneigh = list ->numneigh;
```

```
firstneigh = list \rightarrow firstneigh;
```

// loop over neighbors of my atoms

```
for (ii = 0; ii < inum; ii++) {
    i = ilist[ii];
    itype = type[i];
    jlist = firstneigh[i];
    jnum = numneigh[i];
    xtmp = x [i] [0];
    ytmp = x [i] [1];
    ztmp = x [i] [2];
    if(id[i] = firstatom) 
        atom1 = -1000;
    }
    if(id[i]!=firstatom){
        atom1=atom->map(id[i]-1);
        atom1=domain->closest_image(i,atom1);
        if (type [atom1]!=type [i]) {
             atom1 = -1000;
        }
    }
    if (id [i]==lastatom) {
        atom2 = -1000;
    }
    if (id [i]!=lastatom) {
        atom2=atom->map(id[i]+1);
        atom2=domain->closest_image(i,atom2);
        if (type [atom2]!=type [i]) {
             atom 2 = -1000;
```

```
}
}
for (indexi=0;indexi<9;indexi++){</pre>
                if(atom1==-1000){
                                 if(indexi < 4)
                                                 x_0 = x [i] [0] - (x [atom2] [0] - x [i] [0]) /2; y_0 = x [i]
                                                            [1] - (x [atom2] [1] - x [i] [1]) /2; zo=x [i] [2] - (
                                                            x [atom2][2] - x [i][2]) / 2;
                                                 xf = x [i] [0]; yf = x [i] [1]; zf = x [i] [2];
                                                 xtmpi[indexi]=xo+indexi*(xf-xo)/4;ytmpi[
                                                             indexi]=yo+indexi*(yf-yo)/4;ztmpi[indexi
                                                            =zo+indexi*(zf-zo)/4;
                                 }
                                 if(indexi==4){
                                                 xtmpi[indexi] = x[i][0]; ytmpi[indexi] = x[i][1];
                                                             ztmpi[indexi] = x[i][2];
                                 }
                                 if(indexi>4){
                                                 x_0 = x [i] [0]; y_0 = x [i] [1]; z_0 = x [i] [2];
                                                 xf = x [i] [0] + (x [atom2] [0] - x [i] [0]) /2; yf = x [i]
                                                             [1] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [1] /2; z
                                                            x [atom2][2] - x [i][2]) / 2;
                                                 xtmpi [indexi] = xo + (indexi - 4) * (xf - xo) / 4; ytmpi [
                                                             indexi]=yo+(indexi-4)*(yf-yo)/4;ztmpi[
                                                             indexi]=zo+(indexi-4)*(zf-zo)/4;
                                 }
                }
                 else if (atom2==-1000){
                                 if(indexi < 4){
```

```
xo=x [i][0] - (x [i][0] - x [atom1][0]) /2; yo=x [i
                                       [1] - (x[i][1] - x[atom1][1]) / 2; zo=x[i][2] - (
                                       x[i][2] - x[atom1][2]) / 2;
                              xf = x[i][0]; yf = x[i][1]; zf = x[i][2];
                             xtmpi[indexi]=xo+indexi*(xf-xo)/4;ytmpi[
                                        indexi]=vo+indexi*(vf-vo)/4;ztmpi[indexi
                                       =z_0+index_i * (z_f-z_0)/4;
               }
               if(indexi==4){
                             xtmpi[indexi] = x[i][0]; ytmpi[indexi] = x[i][1];
                                        ztmpi[indexi] = x[i][2];
               }
               if(indexi>4){
                             x_0 = x [i] [0]; y_0 = x [i] [1]; z_0 = x [i] [2];
                             x = x [i] = 0 + (x [i] = 0) - x = 10 
                                        [1] + (x[i][1] - x[atom1][1]) /2; zf = x[i][2] + (x[i][2] + (x[i][2]) + (x[i
                                        x[i][2] - x[atom1][2]) / 2;
                             xtmpi[indexi] = xo + (indexi - 4) * (xf - xo) / 4; ytmpi[
                                        indexi]=yo+(indexi-4)*(yf-yo)/4;ztmpi[
                                        indexi]=zo+(indexi-4)*(zf-zo)/4;
               }
}
else{
               if(indexi < 4)
                             x_0 = x [i] [0] - (x [i] [0] - x [atom1] [0]) /2; y_0 = x [i]
                                        [1] - (x[i][1] - x[atom1][1]) /2; zo=x[i][2] - (
                                        x[i][2] - x[atom1][2]) / 2;
                              xf = x[i][0]; yf = x[i][1]; zf = x[i][2];
                             xtmpi[indexi]=xo+indexi*(xf-xo)/4; ytmpi[
```

```
indexi]=yo+indexi*(yf-yo)/4;ztmpi[indexi
```

```
=zo+indexi*(zf-zo)/4;
                                  }
                                   if(indexi==4){
                                                   xtmpi[indexi]=x[i][0]; ytmpi[indexi]=x[i][1];
                                                               ztmpi[indexi] = x[i][2];
                                  }
                                   if(indexi>4)
                                                   x_0 = x [i] [0]; y_0 = x [i] [1]; z_0 = x [i] [2];
                                                   xf = x [i] [0] + (x [atom2] [0] - x [i] [0]) /2; yf = x [i]
                                                               [1] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [2] + (x [atom2] [1] - x [i] [1]) /2; zf = x [i] [1] 
                                                               x [atom2][2] - x [i][2]) / 2;
                                                   xtmpi[indexi]=xo+(indexi-4)*(xf-xo)/4;ytmpi[
                                                               indexi]=yo+(indexi-4)*(yf-yo)/4;ztmpi[
                                                               indexi]=zo+(indexi-4)*(zf-zo)/4;
                                  }
                 }
}
for (jj = 0; jj < jnum; jj++) {
                 j = jlist[jj];
                 factor_lj = special_lj[sbmask(j)];
                 j \&= NEIGHMASK;
                 jtype = type[j];
                 if (id [j]==firstatom) {
                                  atom3 = -1000;
                 }
                 if (id [j]!=firstatom){
                                  atom3=atom->map(id[j]-1);
                                  atom3=domain->closest_image(j,atom3);
                                   if (type [atom3]!=type [j]) {
                                                   atom3 = -1000;
```

```
}
}
if(id[j]==lastatom){
    atom4 = -1000;
}
if (id [j]!=lastatom) {
    atom4=atom->map(id[j]+1);
    atom4=domain->closest_image(j,atom4);
    if (type [atom4]!=type [j]) {
         atom4 = -1000;
    }
}
energy = 0;
delxo = xtmp - x[j][0];
delyo = ytmp - x[j][1];
delzo = ztmp - x[j][2];
rsq = delxo*delxo + delyo*delyo + delzo*delzo;
if (rsq < cutsq[itype][jtype]){</pre>
    for (index_j=0;index_j<9;index_j++){
         if(atom3 = -1000){
             if (index_j < 4)
                  x_0 = x[j][0] - (x[atom4][0] - x[j][0]) /2;
                     vo=x [j][1] - (x[atom4][1] - x[j][1])
                     /2; zo=x[j][2] - (x[atom4][2] - x[j])
                     [2])/2;
                  xf = x[j] [0]; yf = x[j] [1]; zf = x[j] [2];
                  xtmpj[indexj]=xo+indexj*(xf-xo)/4;
                     ytmpj[indexj]=yo+indexj*(yf-yo)
```

```
/4; ztmpj [indexj]=zo+indexj*(zf-zo
            )/4;
    }
    if(index_{j}==4){
         xtmpj[indexj] = x[j][0]; ytmpj[indexj] =
            x[j][1]; ztmpj[indexj]=x[j][2];
    }
    if (indexj >4){
         x_0 = x [j] [0]; y_0 = x [j] [1]; z_0 = x [j] [2];
         xf = x[j][0] + (x[atom4][0] - x[j][0]) /2;
            yf = x[j][1] + (x[atom4][1] - x[j][1])
            /2; zf = x[j][2] + (x[atom4][2] - x[j])
            ][2])/2;
         xtmpi[indexi] = xo + (indexi - 4) * (xf - xo)
            /4; ytmpj [indexj]=yo+(indexj-4)*(
            yf-yo)/4;ztmpj[indexj]=zo+(indexj
            -4 * (zf - zo) / 4;
    }
else if (atom4==-1000){
    if(index_{j} < 4){
         x_0 = x[j][0] - (x[j][0] - x[atom3][0]) /2;
            y_0 = x[j][1] - (x[j][1] - x[atom3][1])
            /2; zo=x [j][2] - (x [j][2] - x [atom3
            ][2])/2;
         xf = x[j] [0]; yf = x[j] [1]; zf = x[j] [2];
         xtmpj[indexj]=xo+indexj*(xf-xo)/4;
            ytmpj[indexj]=yo+indexj*(yf-yo)
            /4; ztmpj [indexj]=zo+indexj*(zf-zo
            )/4;
```

```
}
     if(index_{j}==4){
         xtmpj[indexj] = x[j][0]; ytmpj[indexj] =
             x[j][1]; ztmpj[indexj]=x[j][2];
     }
     if(indexj > 4){
         x_0 = x [j] [0]; y_0 = x [j] [1]; z_0 = x [j] [2];
         xf = x[j][0] + (x[j][0] - x[atom3][0])/2;
             yf = x[j][1] + (x[j][1] - x[atom3][1])
             /2; zf = x [j] [2] + (x [j] [2] - x [atom3])
             ][2])/2;
         xtmpj[indexj]=xo+(indexj-4)*(xf-xo)
             /4; ytmpj [indexj]=yo+(indexj-4)*(
             yf-yo)/4;ztmpj[indexj]=zo+(indexj
             -4 * (zf - zo) / 4;
    }
}
else{
     if(index_j < 4){
         x_0 = x[j][0] - (x[j][0] - x[atom3][0]) /2;
             yo=x [j][1] - (x [j][1] - x [atom3][1])
             /2; zo=x [j][2] - (x [j][2] - x [atom3
             ][2])/2;
         xf = x[j] [0]; yf = x[j] [1]; zf = x[j] [2];
         xtmpj[indexj]=xo+indexj*(xf-xo)/4;
             ytmpj[indexj]=yo+indexj*(yf-yo)
             /4; ztmpj[indexj]=zo+indexj*(zf-zo
             )/4;
     }
     if(index_{j}==4){
```

```
xtmpj[indexj]=x[j][0];ytmpj[indexj]=
                 x[j][1]; ztmpj[indexj]=x[j][2];
         }
         if(index_j > 4){
             x_0 = x [j] [0]; y_0 = x [j] [1]; z_0 = x [j] [2];
             xf = x[j][0] + (x[atom4][0] - x[j][0]) / 2;
                 yf = x[j][1] + (x[atom4][1] - x[j][1])
                /2; zf = x[j][2] + (x[atom4][2] - x[j])
                 ][2])/2;
             xtmpj[indexj]=xo+(indexj-4)*(xf-xo)
                 /4; ytmpj [indexj]=yo+(indexj-4)*(
                 yf-yo)/4; ztmpj[indexj]=zo+(indexj
                -4 * (zf - zo) / 4;
        }
    }
for (indexi=0;indexi<9;indexi++){</pre>
    for (indexj=0;indexj<9;indexj++){</pre>
         delx=xtmpi[indexi]-xtmpj[indexj]; dely=
            ytmpi[indexi]-ytmpj[indexj]; delz=
            ztmpi[indexi]-ztmpj[indexj];
         rmod=sqrt(delx*delx + dely*dely + delz*
            delz);
         rred=rmod-2*rcnt;
         if (rred<rcut) {
             r2 = rred * rred;
             r4 = r2 * r2;
             r10 = r4 * r4 * r2;
             forcelj = (lj1[itype]]/(r10*
                 rred) - lj2[itype][jtype]/(r4*
```

```
rred))-(lj1[itype][jtype]/(
                rcutalp*rcut) - lj2[itype][jtype
                ]/(rcutbet*rcut));
             fpair = factor_lj*forcelj/(rmod);
             f[i][0] += delx * fpair;
             f[i][1] += dely * fpair;
             f[i][2] += delz * fpair;
             if (newton_pair || j < nlocal) {
                 f[j][0] \rightarrow = delx * fpair;
                 f[j][1] \longrightarrow dely*fpair;
                 f[j][2] \longrightarrow delz * fpair;
             }
             energy=energy + (lj3[itype][jtype]/
                r10-lj4[itype]/r4) - (lj3[
                itype][jtype]/rcutalp-lj4[itype][
                jtype]/rcutbet) + (lj1[itype][
                jtype]/(rcutalp*rcut) - lj2[itype
                ][jtype]/(rcutbet*rcut))*(rred-
                rcut);
        }
    }
if (eflag) {
    evdwl =energy;
    evdwl *= factor_lj;
if (evflag) ev_tally(i, j, nlocal, newton_pair,
                       evdwl,0.0, fpair, delx, dely,
```

}

