Theses and Dissertations   1. Thesis and Dissertation Collection, all items

Molecular dynamic modeling and simulation for polymers

Harrell, Anthony F.

Monterey, California. Naval Postgraduate School

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MOLECULAR DYNAMIC MODELING AND SIMULATION FOR POLYMERS

by

Anthony F. Harrell

September 2003

Thesis Advisor: Young W. Kwon

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2003	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE: Molecular Dynamic Modeling and Simulation for Polymers			**5. FUNDING NUMBERS**
6. AUTHOR(S) Anthony F. Harrell			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**
13. ABSTRACT *(maximum 200 words)* Polymers have been widely used in various engineering applications. For more than a quarter century, the materials have been utilized intensively for the binding materials for composites. The material properties of the binding materials called matrix materials play an important role for the composite material behaviors. As a result, the objective of this study was to understand the mechanical properties of polymers. In particular, the goal was to develop insights as to how a molecular level structure is connected to the bulk properties of materials assuming homogeneity. To this end, molecular dynamics was used to model and simulate the polymeric behaviors. Polymeric chains were modeled using the bead and spring model along with interacting potentials. The study examined the effects of different sizes, densities, and numbers of molecules per chain on the shear moduli of the polymers. Furthermore, some preliminary study was also conducted for metallic particle reinforced polymer composites.			
14. SUBJECT TERMS Molecular Dynamics, Shear Modulus, Polymers, Metallic Potential, Embedded Atom Method, Lennard Jones Potential			**15. NUMBER OF PAGES** 67
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

i

THIS PAGE INTENTIONALLY LEFT BLANK

MOLECULAR DYNAMIC MODELING AND SIMULATION FOR POLYMERS

Anthony F. Harrell
Lieutenant, United States Navy
B.S., US Naval Academy, 1996

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 2003

Author:

        Anthony F. Harrell

Approved by:

        Young W. Kwon
        Thesis Advisor

        Anthony J. Healey
        Chairman, Department of Mechanical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Polymers have been widely used in various engineering applications. For more than a quarter century, the materials have been utilized intensively for the binding materials for composites. The material properties of the binding materials called matrix materials play an important role for the composite material behaviors. As a result, the objective of this study was to understand the mechanical properties of polymers. In particular, the goal was to develop insights as to how a molecular level structure is connected to the bulk properties of materials assuming homogeneity. To this end, molecular dynamics was used to model and simulate the polymeric behaviors. Polymeric chains were modeled using the bead and spring model along with interacting potentials. The study examined the effects of different sizes, densities, and numbers of molecules per chain on the shear moduli of the polymers. Furthermore, some preliminary study was also conducted for metallic particle reinforced polymer composites.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would first and foremost like to thank God for being my provider and giving me the strength, mind, and knowledge to complete the research for this thesis.

Professor Young W. Kwon is the person who has guided me and been the backbone for molecular dynamics studies here at the Naval Postgraduate School; and to him I owe a great deal of gratitude for his relentless efforts, dedication to job accomplishment and technical insight.

Lastly, I would like to thank my family for their support. Their presence, understanding, and spiritual guidance were truly instrumental to the completion of my education at the Naval Postgraduate School.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A.   MECHANICAL PROPERTIES

Mechanical properties are the totality of components determining the bodies' response to external and mechanical influences. All of this is manifested in the ability for the bodies to develop reversible and irreversible deformations and to resist failure. Determining material performance requires defining a force constant or Young's modulus. One of two key parameters for this determination is stress, the amount of applied force. The second parameter is strain, the displacement. With the use of molecular dynamics simulations, polymer properties as well as behaviors can be investigated on a nano-scale.

B.   PURPOSE OF MOLECULAR DYNAMICS

Due to the wide applications in technology for polymer usage, the characteristic behaviors of polymer materials have become of theoretical and experimental interest. The methodology employed includes molecular dynamics simulation that uses models representing the molecules at various levels of detail. The behaviors of molecules close to interfaces, in thin films and in confining geometries, have attracted growing interest from both the theoretical and applications points of view [1].

Shear modulus is one of important properties of which molecular dynamics is used to observe the behavior. Polymer simulation embraces a broad range of factors to pinpoint the behavior. There is a state of constant movement by the polymers in which the presence of solid surfaces strongly influences the matrix properties [2]. A picture of a molecular dynamic model is shown in Figure 1.   The goal is to develop new insights as to how a molecular structure, and dynamics, is connected to the bulk properties of materials. The force fields for the polymer chains are clearly seen in Figure 2.

Molecular Dynamics computer simulations are also used to observe the molecular motions between particles and composites.   The study of how nanoparticles influence the structure and dynamics of polymers can be done in great detail through the use of
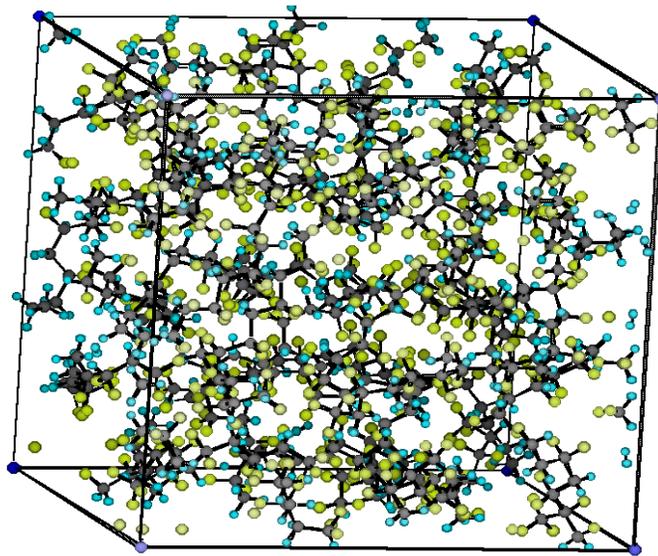
Figure 1.   Molecular Dynamic Simulation

molecular simulations. Computer simulation approaches provide a detailed molecular description of polymer chains' behavior that lead to new insights into materials properties.

The Force Field
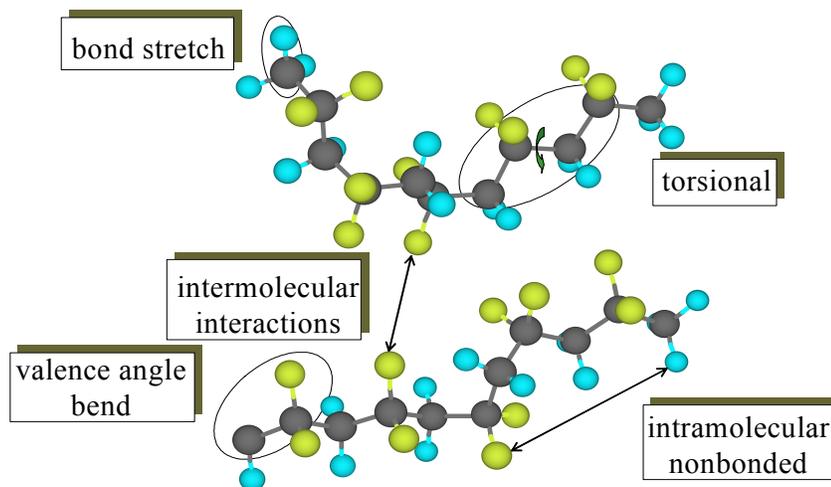
Figure 2.   Force Field for MD Simulations

C. BACKGROUND

1. Polymers

Long chain molecules called polymers are the foundation for major products of various industries today. Polymers are used to make a range of different types of products as summarized in the table below. These materials have so many varied characteristics and applications that their usefulness can only be measured by the imagination.

A polymer is a large molecule formed by joining many small 'monomer' units. Many common classes of polymers are composed of hydrocarbons. Carbon makes up the backbone of the molecule, and hydrogen atoms are bonded along this backbone.

Polymer Type	Example of use
Fibers	Nylon
Plastics	Films, moldings
Elastomers	Rubbers
Coatings	Kitchen worktops, food-can linings
Others	Biodegradable materials

Table 1   Polymer Uses

2. Molecular Dynamics

In many cases, molecular dynamics simulations are based on the Lennard-Jones potential of inter-atomic interactions. For a pair of atoms i and j located at r_i and r_j, the potential energy is given by [3]:

$$U_{LJ}(r_{ij}) = 4\zeta\left[\left(\frac{\tau}{r_{ij}}\right)^{12} - \left(\frac{\tau}{r_{ij}}\right)^{6}\right],$$

where $\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$, $r = |\vec{r}_{ij}|$, ζ and τ are phenomenological parameters specific to each material system. The Lennard-Jones potential features a short-range repulsive potential,

3

together with a long-range attractive potential. The $1/r^6$ attractive potential arises from fluctuating dipole moments on neutral atoms, while the $1/r^{12}$ term models the short-range repulsion that arises from the Pauli exclusion effects of overlapping electron clouds. The Lennard-Jones potential defines the repulsion, attraction and finally the cutoff range where the potential no longer is applicable. The first simplification in molecular dynamics is to ignore the attractive tail. This modifies the above equation to:

$$U_{LJ}(r_{ij}) = 4\zeta\left[\left(\frac{\tau}{r_{ij}}\right)^{12} - \left(\frac{\tau}{r_{ij}}\right)^{6}\right] + \zeta, \quad r_{ij} \leq r_c = 2^{1/6}\tau,$$

with the cut-off radius, r_c, chosen such that $U(r_c) = 0$. The desired result is for the force of the interaction that is given by $\vec{f} = -\nabla U(r)$ to result in:

$$\vec{f}_{ij} = \frac{48\zeta}{\tau^2}\left[\left(\frac{\tau}{r_{ij}}\right)^{14} - \frac{1}{2}\left(\frac{\tau}{r_{ij}}\right)^{8}\right]\vec{r}_{ij}.$$

As r increases to r_c, the force drops to zero, and this prevents any discontinuities.

3. Crosslinking Density

Molecular dynamics simulations are heavily demanded for the study of randomly cross-linked polymers. Barsky and Plischke previously reported on simulations that involved short chains and small system sizes [4]. These simulations "demonstrated the existence of a universal function $P(\xi)$ that describes the distribution of localization lengths for a wide range of crosslink density" [4]. The relationship between the shear modulus and density were never taking into account. During later studies, Barsky and Plischke extended their simulations to longer polymer chains and systems [4]. Their focused was on the shear modulus $E(n)$ as a function of the density of crosslinks n.

The purpose of the crosslinks is to tie all the polymer molecules together. This prevents molecules from flowing past or around each other once the temperature increases, and also increases their resistance to melting. The advantage of being tied together is that they are not easily broken apart from each other. The difference between single uncrosslinked polymer chains and a crosslinked network is described in Figure 3.

However, in this experiment the temperature was held constant as well as the cross-linking density.

When polymers become crosslinked, this becomes this
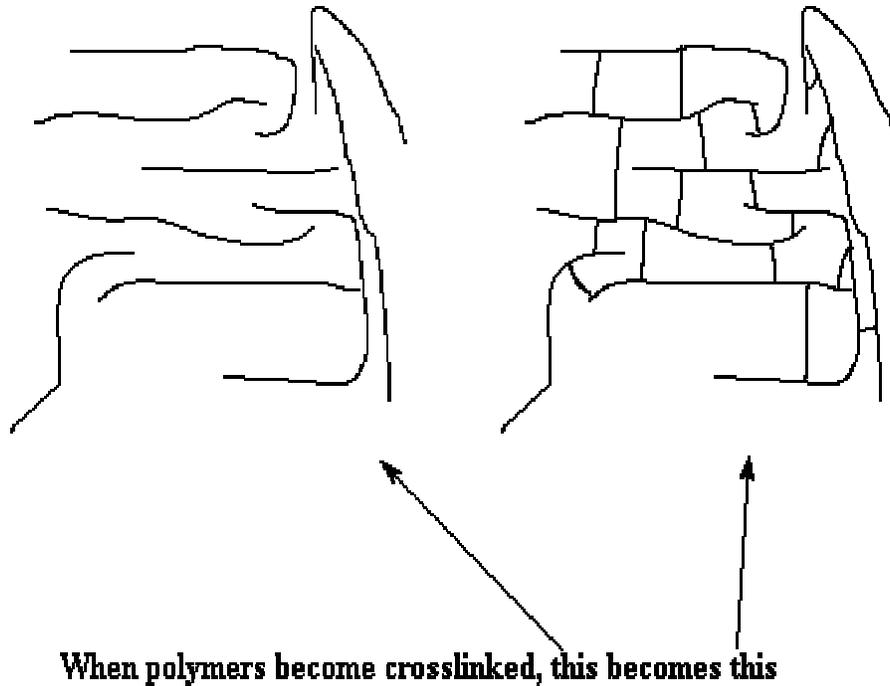
Figure 3.　　How polymers appear once they are crosslinked.

4.　　Embedded Atom Method (EAM)

The motions of individual molecules are best computed through molecular dynamics simulations. The energy and pressure of an equilibrium system are best seen through the use of these simulations. Temperature and density are the terms used to express energy and pressures. The temperature and density are instrumental in obtaining the thermodynamics of a system. The energy of a system is obtained by reversing the process described above. This scheme is called the embedded atom method (EAM) [5]. As pointed out by Daw in another context [6], the electron density uniquely specifies the energy of a system of atoms. With the uses of numerical methods, the forces can be determined from the system's energy as well as the thermodynamic properties of the material. The energy of a system can be represented by the EAM as such [6]:

$$E_{tot} = E_{kin} + E_{pot} \text{ or } E_{tot} = \sum_{i} E_{ee}(\rho_{h,i}) + \frac{1}{2}\sum_{i,j}\phi_{ij}(r_{ij}),$$

5

where $\rho_{h,i}$ is the total electron density seen by atom i due to the rest of the atoms in the system, E_{ee} is the embedding energy for placing an atom into that electron density and ϕ_{ij} is the short range pair interaction representing the core-core repulsion of atoms i and j separated by distance r_{ij}.

D.   OBJECTIVES

In the last decade, a tremendous amount of attention has been dedicated to the development of polymeric blends. The production of materials with tailor made properties may be constructed with the use of these blends. This in turn may offer the development of completely new polymeric materials [7]. Analyzing and predicting polymer properties using atomistic simulations are crucial in the production of materials. Mechanical behavior, cohesion, permeability, and miscibility are all critical to the success of polymeric materials in technological applications. Polymer scientists seek to establish the connection between such properties, and the microscopic and mesoscopic structure of the polymers. The method used by scientist is atomistic simulation where a sample containing a few thousand atoms is a characteristic of the bulk system, and relaxation times for the property of interest are under a nanosecond.

Studying polymers to enhance the material properties of structures requires a model to predict the behavior of molecules. In particular, this thesis examines how the modulus of rigidity changes with density, and with a crystallic structure embedded in polymeric chains. This study uses microscopic principals to examine the molecular dynamics simulations. Scientists that have approached this study have all agreed that the polymer properties substantially deviate from the behavior in the bulk when closer to the solid interface [8-10]. Specifically, this thesis evaluates how many molecules per given chain are necessary for homogeneous behavior to occur.

II.   PRE-PROCESSOR

A.   INTRODUCTION

The contents of this chapter describe the construction of a molecular model. The goal is to randomly generate polymer chains within a cube and determine how many molecules per given chain are necessary to produce homogenous behavior. Computer simulations were used to focus on randomly dispersed particles in 3-D space. These simulations contained different volumes as well as polymers with different lengths, N = 5…45 with unit multiples of five molecules in a chain. The 3-D models of polymers were used to create structures that properly represent real molecules, thus, providing various construction and analysis strategies. Figure 4. shows a simulation with twelve polymer chains extending both in the x and y directions and two chains in the z direction.

With the use of crosslinking potential systems, molecular dynamics simulations were conducted in the three-dimensional polymeric matrix state. The simulations emphasize the relation between the correlation volume of a space, and the concentration of the random particles.
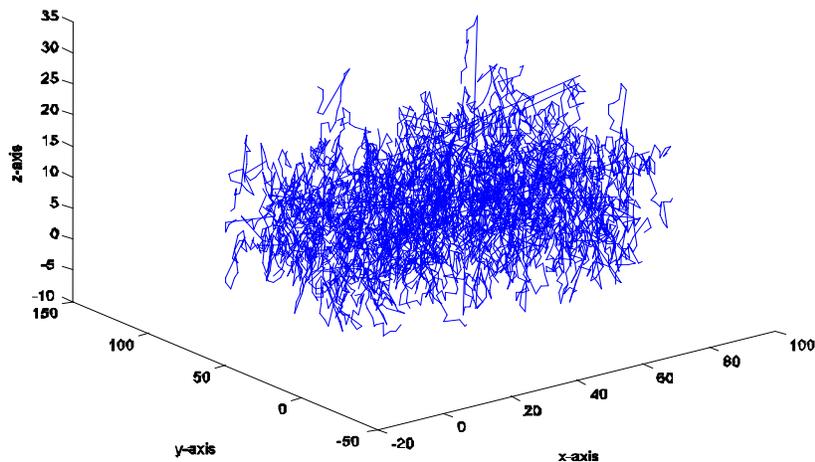
Figure 4.   Composite material with particles randomly dispersed in a space volume of 90X90X15.

B. MOLECULAR MODEL

The interest lies in using the simplest molecular model possible that allows determination of the underlying physical phenomena of homogeneous behavior between the polymers. Polymer chains are typically long in nature. Theoretical models are based on interacting sites of the molecule connected by rods. A program was written to represent them in a similar manner to common theoretical models and address the details of the interactions between chain atoms as well as the nature of the solvent.

1. Chain Structure

As stated by Rapaport, the simplest model attempts to represent the individual monomers of the polymer and the bonds that link them into chains [11]. Monomers can be simple atoms modeled using a soft-sphere potential and the attractive interaction between chain neighbors can produce the limited length variation [11]. The chains are totally flexible with regards to the limits set by the repulsive potential. To control the interaction regulating the separation of the next-nearest neighbors, a degree of stiffness has to be included.

The model produced in this study was based on Rapaport's description of flexible molecules [11]. All pairs of atoms maintain an attractive interaction between each pair of adjacent bonded atoms. The force on particle i due to particle j, $\mathbf{F}_{ij}$, is of the form:

$$F_{ij} = -\frac{\partial U_{LJ}(r_{ij})}{\partial \vec{r}_i} = -\frac{\partial U_{LJ}(r_{ij})}{\partial r_{ij}} \frac{\partial r_{ij}}{\partial \vec{r}_i}$$

and acts along the vector joining the two particles.

2. Interaction

Forces between the non-bonded atoms belonging to the same chain must be taken into consideration. To compensate, the atoms belonging to the same chain must have identical atom identifications, the distance between molecules within the polymer equally spaced, and a tolerance set, for the allowable minimum distance between any two atoms.

3. Initial State

The selection of a monomer was at random and all other particles within a given distance of this particle were identified. The next random particle was connected to the first by the same potential used for the chain construction. The atoms of each chain must be positioned so that the bond lengths are all within their permitted ranges, and there are no overlapping of atoms either of the same or different chains. The added crystallic structure is organized as a face centered cubic (FCC) lattice, which the already existing chains positioned, to allow for wraparound.

THIS PAGE INTENTIONALLY LEFT BLANK

III. MOLECULAR DYNAMICS PROGRAMS

A. INTRODUCTION

Molecular dynamics is a technique that uses computers to solve the classical equations of motion for a system of atoms or molecules. A system of N particles interacts through a potential to determine the system's evolution through time. From the microscopic state of the system, i.e. particle positions and momenta, macroscopic properties such as pressure and energy can be evaluated.

In this chapter, descriptions of the MD program components for an atomic system at equilibrium are discussed.

B. SPACING

Molecules formed using this MATLAB program which is contained in Appendix A, were first confined to a cube with set dimensions in the x, y, and z directions. For the construction of the polymer chain, an allowable distance between molecules within the polymer was first established. All measurements during the construction were dimensionless. To ensure atoms were not too close or too far apart, an allowable tolerance input was set. After creating the foundation, the actual construction of the polymer chain began. For optimum results that were not too expensive, the number of polymer chains in the x and y directions were the same and the z direction was one third of the x direction's value. The number of molecules contained in each chain was chosen to be constant.

To determine the spacing in each direction, the lengths of the cube were divided by the number of polymer chains in the same direction. The total number of atoms contained in the cube was the product of the number of molecules in a chain and the number of polymer chains in the x, y, and z directions.

As far as the atomic structure is concerned, the bead-spring model was used. In essence, the bead-spring model simulates the hydrodynamic properties of a chain macromolecule. These chains consist of a sequence of beads, each of which offers hydrodynamic resistance to the flow of the surrounding medium. The beads are

connected to each other by a spring which does not contribute to the frictional interaction. However, the spring is responsible for the elastic and deformational properties of the chain. The mutual orientation of the springs is random.

For this project, each bead, or mapping point, represented a specific monomer. This required a fixed distance of the beads along the backbone, bond angles around the backbone, and torsion states around the bead-bead connection to be held constant.

A loop was generated to account for the individual atoms and their physical location. To initiate the chain creation, the first atom's position was generated. The remainder of the atoms along the chain was randomly generated. During the random generation, the atoms' positions were checked to verify if they were within the tolerance. Figures (5) and (6) show the results of the generation. The first 25 atoms of the generated polymers are contained in Tables (2) and (3). The program was run an additional two consecutive times with no changes.

Polymer Chains within a Cube
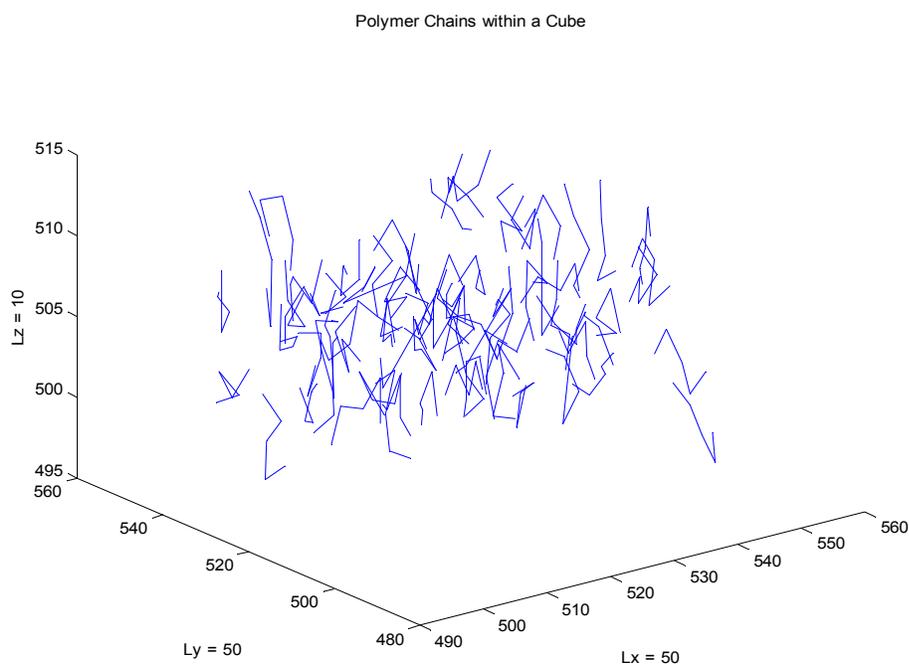
Figure 5.   First iteration of random generated polymers with center (500, 500, 500)

Atom	x	y	z	Atom ID
1.0	504.97609	504.48560	502.45171	101.0
2.0	502.59076	503.47293	503.96324	101.0
3.0	501.99942	502.42032	506.70957	101.0
4.0	499.67814	501.97644	504.86170	101.0
5.0	497.54537	500.01099	505.62873	101.0
6.0	512.37460	503.06179	502.99996	102.0
7.0	514.23613	505.30829	503.69847	102.0
8.0	515.55628	507.07712	505.73033	102.0
9.0	515.60933	504.13545	506.31661	102.0
10.0	516.83033	504.54918	503.84920	102.0
11.0	520.65245	503.22435	502.09824	103.0
12.0	518.28503	505.05995	502.25920	103.0
13.0	519.70948	504.65179	504.86771	103.0
14.0	518.02407	503.33075	506.96872	103.0
15.0	522.95581	505.88808	504.01570	103.0
16.0	528.90952	502.40789	503.49247	104.0
17.0	531.73406	502.57428	502.49534	104.0
18.0	532.67934	500.17630	504.03032	104.0
19.0	533.97179	502.88360	504.03933	104.0
20.0	533.87432	500.11420	505.18864	104.0
21.0	537.16469	505.11076	503.50474	105.0
22.0	535.65462	506.06503	505.91494	105.0
23.0	532.89417	505.52865	506.96004	105.0
24.0	531.48535	502.96331	507.61903	105.0
25.0	529.87864	504.46564	505.57906	105.0

Table 2   First 25 Atoms of Figure (5)

13
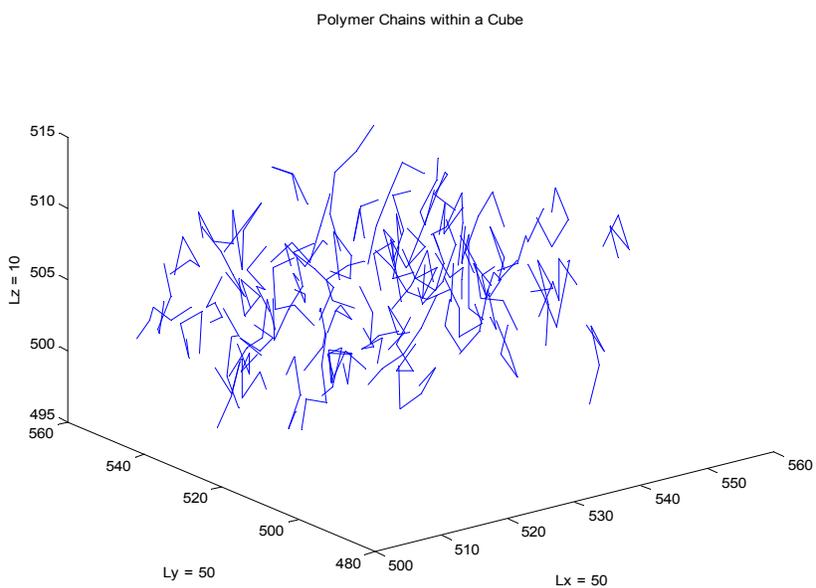
Figure 6.   Second iteration of the same random generated polymers with center (500, 500, 500)

Atom	x	y	z	Atom ID
1.0	505.92154	504.03955	502.51789	101.0
2.0	505.84193	501.20613	503.50041	101.0
3.0	507.36395	501.26903	506.08488	101.0
4.0	506.81471	504.13124	505.37349	101.0
5.0	508.49709	505.65925	503.41522	101.0
6.0	513.94974	505.92256	502.12355	102.0
7.0	516.62952	507.08035	502.81516	102.0
8.0	518.82241	505.27689	503.78403	102.0
9.0	521.22632	504.63156	505.45879	102.0
10.0	522.71982	503.34969	507.72291	102.0
11.0	520.31054	503.88540	501.52354	103.0
12.0	523.12281	504.38051	502.44330	103.0
13.0	522.22704	506.48774	500.50496	103.0
14.0	519.43758	507.16691	499.63453	103.0
15.0	519.57860	507.70756	502.58203	103.0
16.0	527.12844	505.03416	502.87971	104.0
17.0	529.22768	503.53851	504.41470	104.0
18.0	530.24602	503.15056	507.20978	104.0
19.0	531.05067	502.23159	504.46970	104.0
20.0	531.71551	505.11960	504.93591	104.0
21.0	535.71797	504.87248	503.58545	105.0
22.0	533.79039	505.06657	505.87603	105.0
23.0	531.21955	506.59679	506.09787	105.0
24.0	531.74211	508.66777	508.20452	105.0
25.0	529.78274	509.45735	510.33464	105.0

Table 3   First 25 Atoms of Figure (6)

14

C.   POTENTIALS

The interactions between the polymers and atoms lead to prediction of the large-scale bulk properties of material. This molecular dynamics program uses the Lennard Jones (LJ) potential method to evaluate molecule interactions between the polymers and atomic atoms. The LJ potential is an effective potential that describes the interaction between two uncharged molecules or atoms.

The Lennard-Jones potential is mildly attractive as two uncharged molecules or atoms approach one another from a distance, but strongly repulsive when they approach too close.   At equilibrium, the pair of atoms or molecules tends to go toward a separation corresponding to the minimum of the LJ potential. The strong *close-in* repulsion between atoms or molecules is understandable, resulting from mutual deformation of their structures (one atom cannot diffuse through another).   The mild attraction at larger distances is due to the induced dipole-dipole moment interaction of the particles.

1. Polymers

As stated previously, the LJ potential was used to ensure self avoidance of polymers. The general form is:

$$U_{LJ}(r_{ij}) = \begin{cases} 4\varepsilon\left[\left(\dfrac{\sigma}{r_{ij}}\right)^{12} - \left(\dfrac{\sigma}{r_{ij}}\right)^{6} + \dfrac{1}{4}\right] & , \quad r_{ij} < 2^{1/6}\sigma \\ 0 & , \quad r_{ij} \geq 2^{1/6}\sigma \end{cases}$$

where ε and σ, are defined as the energy at the minimum in potential and the distance to zero in potentail, respectively and r_{ij} is the inter-atomic distance.

The following formula was used to compensate for the added attractive potential between neighboring atoms:

$$U_{nn}(r_{ij}) = \begin{cases} -\dfrac{1}{2}kR_o^2 \ln\left[1 - \left(\dfrac{r_{ij}}{R_o}\right)^2\right] & , \quad r_{ij} < R_o \\ \infty & , \quad r_{ij} \geq R_o \end{cases}$$

with $R_o = 1.5\sigma$ and $k=30\varepsilon/\sigma^2$. By incorporating this method, polymers are prevented from passing through each other [4].

2. Metallic Atoms

The embedded atomic potential as described in section I-C.4.a was used for the metallic atoms. The foundation of the embedded atom method (EAM), based on the quasi-atom concept, is the realization that the cohesive energy of a metallic system can be expressed in terms of embedding energies. Daw *et al.* [12] have resorted to empiricism by fitting the embedding function and pair interaction to basic bulk properties, e.g., lattice constant, cohesive energy, elastic constants, and vacancy formation energy. Figure 7 shows the creation of a polymer with an aluminum core.
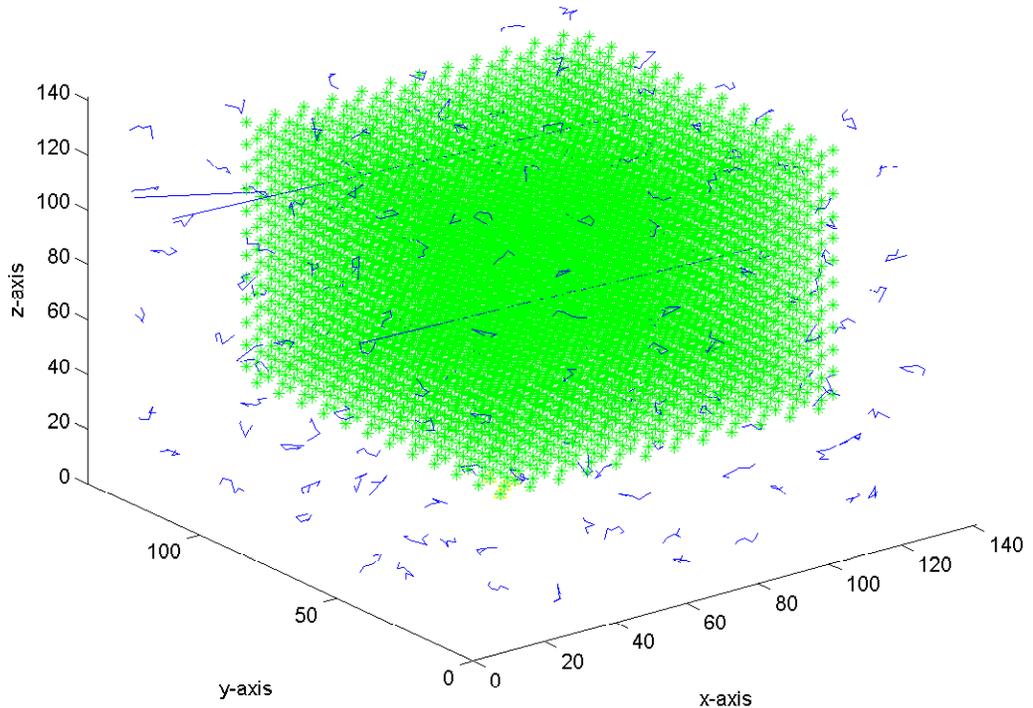
Figure 7.   Random Generate Polymer with Aluminum center contained in a Volume = 141X141X141. Number of polymers in each chain in x, y, and z direction is 6 with 6 beads in each chain. Total number of atoms in structure is 8160 atoms.

D. MOLECULE INTERACTION

The Lennard-Jones potential is used to determine the forces acting on each particle. The forces are used in solving the particle trajectories as they evolve in time. In this program, the potential file contains the following information: atomic number, atomic mass, lattice spacing, potential cut-off range for each atom, grid of inter-atomic force, grid of atom electron density and the spherically averaged atomic density [12]. To determine the embedding energy of each atom, the inter-atomic force is interpolated for each atom site and multiplied with the atom density. The cohesion energy of the system is determined by summing the embedding energy over all the host atoms. Lastly, the force is computed by multiplying the electron density at each atom with the embedded energy.

E. MODEL

Within the model created, the number of monomers (beads) on each chain may be changed by the user. This input is multiplied by the number of polymers chains in the x, y, and z directions to determine the number of polymers in the system.

Each system was equilibrated to an average dimensionless temperature of 300 and a time step equal to 0.01. This process was completed through a constant energy molecular dynamics code where the user was required to enter the number of atoms contained in the system.

With the use of the equations contained in section III-C above, the polymer chains were created and cross-linking introduced. A molecule was selected at random and then connected to another random molecule within the same polymer chain. A standard distance was used for the molecules along with a combination of the potentials for polymers described above. Molecules contained in the same chain were only linked once, and no cross-linking of neighboring atoms was permitted.

F. AUTO- CORRELATION

All of the viscometric functions are derived from elements of the stress tensor. The stress tensor can be determined from the pressure tensor, so only one of these need

be calculated by the program. In an equilibrium molecular dynamics simulation, Newton's equations describing the motion of the atoms in the model system are solved as a function of time. The viscosity η is then given as an integral of the stress-stress autocorrelation functions determined during the simulation,

$$\eta = \frac{V}{k_B T} \lim_{\tau \to \infty} \int_0^\tau dt \left\langle P_{xy}(t) P_{xy}(0) \right\rangle$$

where V is the volume of the system, k_B is Boltzmann's constant, T is temperature, and t is time. The quantity $P_{xy}(t)$ is the value of the xy component of the traceless symmetric stress tensor at time t, and so $P_{xy}(t)P_{xy}(0)$ is the stress-stress autocorrelation function measured during the course of the simulation.

IV.   RESULTS AND DISCUSSION

A.   INTRODUCTION

In this chapter the results of all the simulations are presented and discussed. Before these simulations were carried out, extensive work was required to ensure the program functioned correctly.

Dimensionless units were used to compute the stress tensor in the xy, yz, and xz directions. Several variables remained constant throughout each simulation to observe the polymer behavior. The temperature remained constant at a rate of 300 with a time step of 0.01. The number of atoms (NATOMS) was initially computed by the number of beads or molecules in a chain multiplied by the number of polymer chains in each direction. The density was then calculated by dividing the NATOMS by the cube's volume.

All simulations ran a minimum of three times. An average of the results was recorded and plotted. In all of the simulations completed, there is a noticeable homogeneous trend. These results are presented in the sections to follow.

B.   VARYING DENSITY AND VOLUME PROPORTIONALLY

The first simulation required proportionally varying the x and y dimensions of the cube along with the number of atoms in the same directions. The cube length in the z direction was set to a constant and the number of polymer chains extending in this direction was two. The number of monomers in a chain remained constant at 20. We noted that the normalized shear modulus approaches a constant value, as seen in Figure 8 when NATOMS becomes more than 8000.
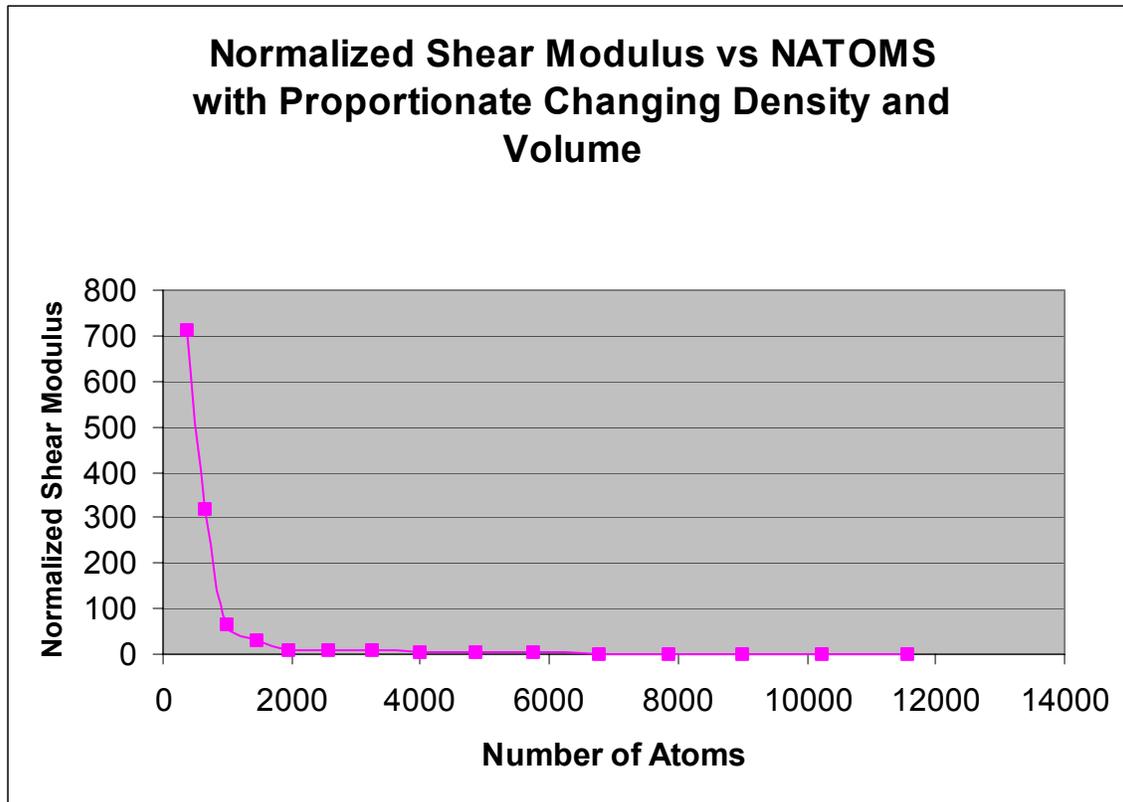
Figure 8.   Plot of the normalized shear modulus as a result of proportionally varying the density with the volume of the cube.

C.   CONSTANT VOLUME

During this simulation, the volume was held constant, and the number of monomers per chain was held at 20. The number of polymer chains which extended in the z direction remained constant at Nz=2; however, the x and y directions were increased by one unit starting from Nx=Ny=3 after a series of runs were completed. After the total number of atoms becomes about 9000, the shear modulus remains roughly constant as seen in Figure 9.

Shear Modulus vs NATOMS with Constant Volume and Varying Density
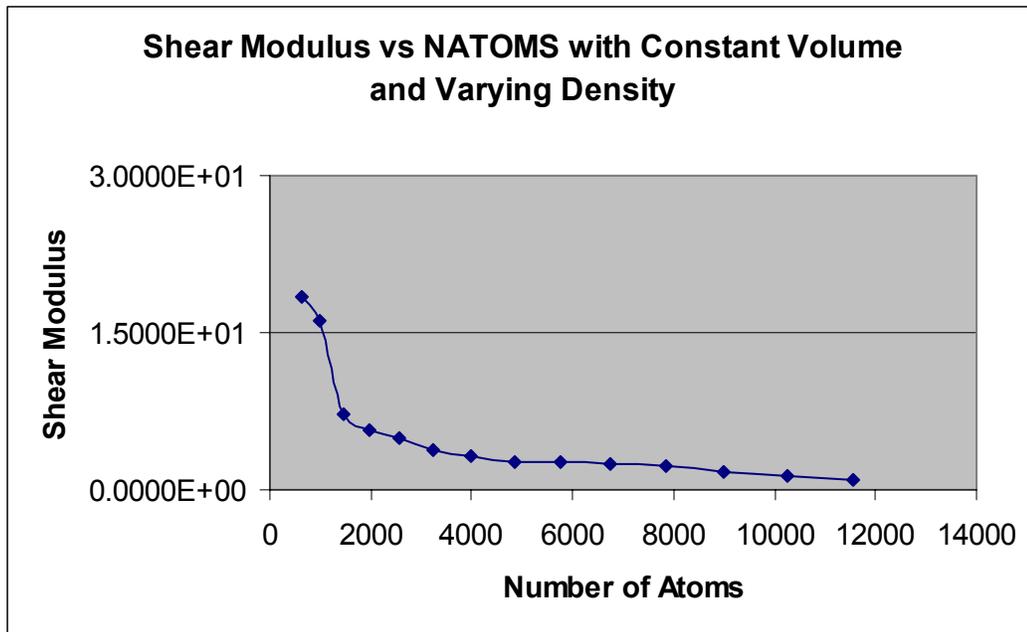
Figure 9.   Plot of the normalized shear modulus determined from a cube of constant volume (112.5X112.5X15) as a function of varying density.

D.   CONSTANT VOLUME AND DENSITY WITH VARIANCE IN NUMBER OF MOLECULES PER CHAIN

This simulation required holding both the volume and density constant. The one varying parameter was the number of monomers per given chain (N). The increments were in units of 5 beginning with N=5…45. The results show that the normalized shear modulus begins to approach a constant value as NATOMS becomes more than 8000 as seen in Figure 10.
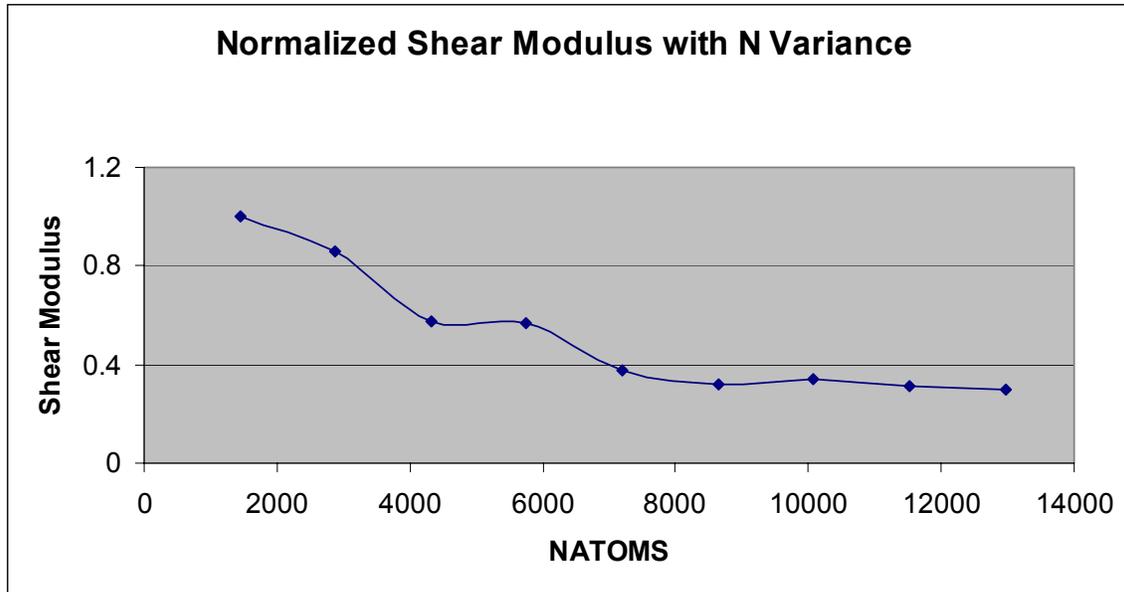
Figure 10.   Plot of the normalized shear modulus determined from a cube with constant volume and polymer density as a function of molecules variance.

E.   CONSTANT VOLUME WITH METALLIC CORE

A face center cube (FCC) structure was inserted in the center of a cube with a constant volume.   The FCC was used to simulate the characteristics of aluminum. There were six polymer chains extending in each direction with a variance in the number of monomers per given chain for the first simulation.   The varying parameters for this simulation included the size of the FCC and the number of polymers interacting with this metallic structure.   Thus, the number of monomers per given chain decreased proportionally with the expansion of the metallic structure to maintain a constant density.

Ideally, this simulation was created to represent a composite material. As the metallic core size increased, the polymers' space became more confined due to the constant volume of the cube. This means the volume fraction of the metallic particle increases in a polymer composite. As shown in Figure 11, the normalized shear modulus continuously increases along with the increasing size of the metallic core.
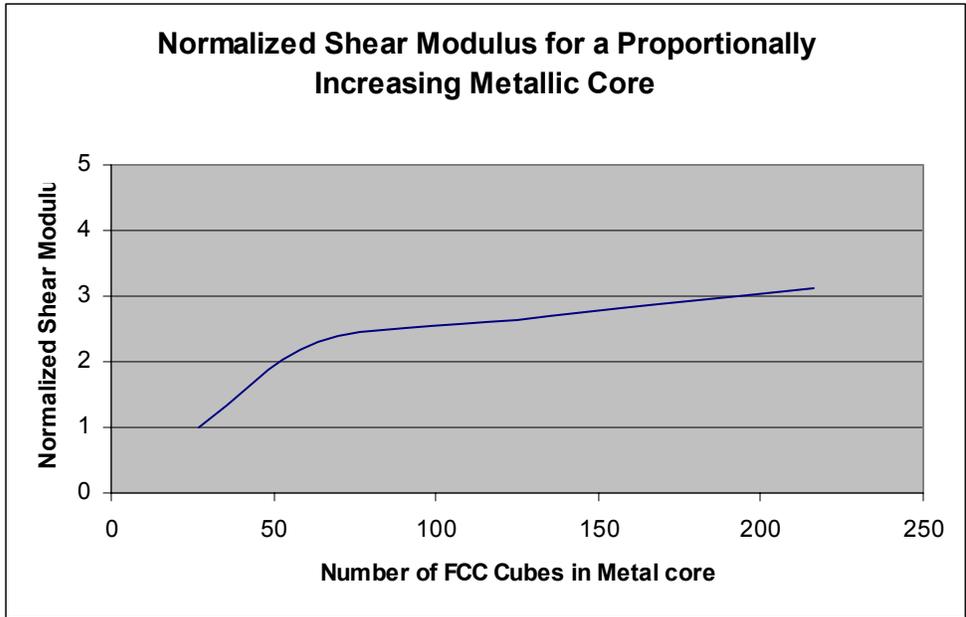
Figure 11.   Plot of the average normalized shear modulus for a cube with a constant volume and increasing metallic core size under constant polymer density.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

Atomistic simulations have been carried out on a wide range of polymers. The subject of scale of microstructure on bulk properties is the basis of this study for polymer matrix nano-composites. Extremely small, rigid particles can be distributed in a soft matrix to engineer specific physical properties. The purpose of this is to provide insights on theoretical methods usage which describes the mechanical properties (stress/strain performance) of nano-materials composed of polymers and metallic structures. Ongoing experimental studies are designed to understand the role of the nano-scale particles in changing the composite mechanical properties. Theoretical modeling of nano-mechanical interactions is applied to examine a variety of unique small scale effects.

The goal of this thesis was to develop insights as to how a molecular level structure is connected to the bulk properties of materials assuming homogeneity. A molecular dynamics program was used to take a closer look at the effects of different sizes, densities, and numbers of molecules per chain on the shear moduli.

Based on the numerous simulations conducted, a common occurrence emerged across the board. The normalized shear modulus decreased to a constant value as the total number of atoms in the model increased. The results show that approximately 8000 or more monomers are needed in order to represent the polymer behavior, regardless of number of polymer chains, number of monomers per chain, or density of the polymers. For the particulate composite, the shear modulus increased along with the size of the metallic particle (i.e. particle volume fraction). In this simulation, if the number of polymers is small, the composite shear modulus could not be determined properly. Thus, it is critical to contain a minimum number of polymers in the molecular dynamic simulation in order to obtain reliable predictions.

B. RECOMMENDATIONS

This project is an important issue that can be very beneficial for polymer material design. Engineers believe the role of polymer composites is rapidly growing due to their

strength, lightweight, chemical stability and tailor-ability [13]. This thesis provides baseline information for obtaining information on the mechanical properties of polymers and bulk material. One important issue that requires further investigation is the cross-linking density relationship. In this thesis, the cross-linking density remained constant throughout all simulations.

Another area for research to consider is the function responses with different types of metals. This will prove beneficial in the creation and modification of other composites.

APPENDIX A - MATLAB CODE

A. GENERATION OF POLYMERS

```
clc;
clf;
clear;

% center postion of the polymer unit
xc=0; yc=0; zc=0;

% dimensions of cube
Lx=28.2*5;
Ly=28.2*5;
Lz=28.2*5;


%distance between molecules within polymer
lc=2.0;

% allowable minimum distance between any two atoms
toler = 2.0;

%Increments in each direction
Nx=6;  % no. of polymer chains in x-dir
Ny=6;  % no. of polymer chains in y-dir
Nz=6;  % no. of polymer chains in z-dir

% number of molecules (bead) in a chain
N=125;


%Spacing in each direction
dx = Lx/Nx;
dy = Ly/Ny;
dz = Lz/Nz;

%FCC Parameters

NUNIT1=4;
NUNIT2=NUNIT1;
NUNIT3=NUNIT1;

DIST = 4.05;

Lx1=(2/5*Lx)+dx/8;
Ly1=(2/5*Ly)+dy/8;
Lz1=(2/5*Lz)+dz/8;
Lx2=(3/5*Lx)+dx/8;
Ly2=(3/5*Ly)+dy/8;
Lz2=(3/5*Lz)+dz/8;

% Lx1=(Lx-(NUNIT2/2))/2 -(NUNIT2*DIST)/2 + dx/8;
% Ly1=Lx1;
```

```
% Lz1=Lx1;
% Lx2=(Lx-(NUNIT2/2))/2 + (NUNIT2*DIST)/2 + dx/8;
% Ly2=Lx2;
% Lz2=Lx2;

Lx11=Lx1+dx/10;
Ly11=Ly1+dy/10;
Lz11=Lz1+dz/10;

NATOM=0;
NMOLE=0;

%Loop for seed points for molecular positions
ij=0;
dummy=0;
for k=1:Nz
   for j=1:Ny
      for i=1:Nx
           xx=(i-1)*dx + dx/2 + xc;
           yy=(j-1)*dy + dy/2 + yc;
           zz=(k-1)*dz + dz/2 + zc;
        if ( (xx>Lx1 & xx<Lx2) & (yy>Ly1 & yy<Ly2) & (zz>Lz1 & zz<Lz2) );
           dummy=dummy+1;
        else
           NMOLE=NMOLE+1;
           ij=ij+1;
           x(ij)=(i-1)*dx + dx/2 + xc;
           y(ij)=(j-1)*dy + dy/2 + yc;
           z(ij)=(k-1)*dz + dz/2 + zc;
        end
      end
   end
end

NATOM=NMOLE*N


figure(1)

%Create FCC

%        Assign initial positions to id 1 for the first 256 atoms
%   based on face-centered cubic lattice total atoms
id=1;
%....    set lattice distance based on cube of side + CUBE

   jatom = 1;
%     NUNIT1=4;
%     NUNIT2=4;
%     NUNIT3=4;
%     DIST = 4.05;
   atomid(jatom) = id;
%....     assign positions of first four atoms
     xnew(jatom) =  Lx11 ;
     ynew(jatom) =  Ly11 ;
     znew(jatom) =  Lz11 ;
```

```
        xnew(jatom+1) = Lx11;
        ynew(jatom+1) = Ly11+DIST;
        znew(jatom+1) = Lz11+DIST;
        xnew(jatom+2) = Lx11+DIST;
        ynew(jatom+2) = Ly11;
        znew(jatom+2) = Lz11+DIST;
        xnew(jatom+3) = Lx11+DIST;
        ynew(jatom+3) = Ly11+DIST;
        znew(jatom+3) = Lz11;
      plot3(xnew,ynew,znew,'y^')
    hold on
% ....     replicate first four positions over NUNITS
    jatom=0;
  for I=1: NUNIT1
      for J=1: NUNIT2
        for K=1: NUNIT3
            for IJ=1:4
                xnew(IJ+jatom) = xnew(IJ) + 2*DIST*(K-1);
                ynew(IJ+jatom) = ynew(IJ) + 2*DIST*(J-1);
                znew(IJ+jatom) = znew(IJ) + 2*DIST*(I-1);
                atomid(IJ+jatom)=id;
            end
            jatom=jatom+4;
             plot3 (xnew,ynew,znew,'g*')
            hold on
        end
      end
  end

    NMET=jatom
    NATOM_TOTAL=(NATOM+NMET)
%Generate Random numbers for specified point
id=20;
for imole=1:NMOLE
  % create the first atom's position of each chain
  id=id+1;
  x1=x(imole); y1=y(imole); z1=z(imole);
  rn=rand(1,3);
  rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
  jatom = jatom + 1;
  atomid(jatom)=id;
  xnew(jatom)=x(imole) + dx*rn1(1,1)*0.5;
  ynew(jatom)=y(imole) + dy*rn1(1,2)*0.5;
  znew(jatom)=z(imole) + dz*rn1(1,3)*0.5;

  % create the remaining (N-1) atoms' positions
   for iatom=2:N
     iflag=0;
     jatom = jatom + 1;
     rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
     mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
     uvx=rn1(1,1)/mag;
     uvy=rn1(1,2)/mag;
     uvz=rn1(1,3)/mag;
     atomid(jatom)=id;
     xnew(jatom) = xnew(jatom-1) + uvx*lc;
```

```matlab
                ynew(jatom) = ynew(jatom-1) + uvy*lc;
                znew(jatom) = znew(jatom-1) + uvz*lc;

                while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
                    (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
                    (znew(jatom)>Lz1 & znew(jatom)<Lz2))

                 iflag = iflag+1;

                 if (iflag > 100)
                    break
                 end

                rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
                mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
                uvx=rn1(1,1)/mag;
                uvy=rn1(1,2)/mag;
                uvz=rn1(1,3)/mag;
                atomid(jatom)=id;
                xnew(jatom) = xnew(jatom-1) + uvx*lc;
                ynew(jatom) = ynew(jatom-1) + uvy*lc;
                znew(jatom) = znew(jatom-1) + uvz*lc;
              end
            end

            %Check whether two atoms' positions are within the tolerance

          if (imole==1)
            for ipatom=2:jatom
                testid=1;
                iter=1;
                while (testid > 0.1 & iter <= 50 )
                iter=iter+1;
                testid=0;
                 for itest=1:(ipatom-1)
                    dist                =                sqrt((xnew(ipatom)-xnew(itest))^2+(ynew(ipatom)-
ynew(itest))^2+(znew(ipatom)-znew(itest))^2);
                    if dist < toler;
                     testid=testid+1;
                    end
                   end

                  if (testid > 0.1)
                   rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;

                   % Magnitude
                   mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2);
                   uvx=rn1(1,1)/mag;
                   uvy=rn1(1,2)/mag;
                   uvz=rn1(1,3)/mag;
                   xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
                   ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
                   znew(ipatom) = znew(ipatom-1) + uvz*lc;
                   iflag=0;
                     while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
                         (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
```

```matlab
                                      (znew(jatom)>Lz1 & znew(jatom)<Lz2))

                         iflag=iflag+1;

                         if (iflag > 100)
                            break
                         end
                         rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
                         mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
                         uvx=rn1(1,1)/mag;
                         uvy=rn1(1,2)/mag;
                         uvz=rn1(1,3)/mag;
                         atomid(jatom)=id;
                         xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
                         ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
                         znew(ipatom) = znew(ipatom-1) + uvz*lc;
                      end
                  end

               end
               if (iter > 49)
                  break
               end
            end
         else
            for ipatom=(imole-1)*N+1:jatom
               testid=1;
               iter=1;
               while (testid > 0.1 & iter < 50 )
               iter=iter+1;
               testid=0;
                for itest=1:(ipatom-1)
                    dist              =                        sqrt((xnew(ipatom)-xnew(itest))^2+(ynew(ipatom)-
ynew(itest))^2+(znew(ipatom)-znew(itest))^2);
                    if dist < toler;
                       testid=testid+1;
                    end
                  end

               if (testid > 0.1)
                  rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;

                  %Magnitude
                  mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2);

                  uvx=rn1(1,1)/mag;
                  uvy=rn1(1,2)/mag;
                  uvz=rn1(1,3)/mag;
                  xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
                  ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
                  znew(ipatom) = znew(ipatom-1) + uvz*lc;
                  iflag=0;

                     while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
                         (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
                         (znew(jatom)>Lz1 & znew(jatom)<Lz2))
```

31

```
                        iflag=iflag+1;

                        if (iflag > 100)
                           break
                        end

                        rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
                        mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
                        uvx=rn1(1,1)/mag;
                        uvy=rn1(1,2)/mag;
                        uvz=rn1(1,3)/mag;
                        atomid(jatom)=id;
                        xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
                        ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
                        znew(ipatom) = znew(ipatom-1) + uvz*lc;
                     end

                end
              end
              if (iter > 49)
                 break
              end
          end
        end
              istart=jatom-N+1;
              iend=jatom;
               plot3(xnew(istart:iend),ynew(istart:iend),znew(istart:iend))
              axis([0 Lx 0 Ly 0 Lz])
              xlabel('x-axis'); ylabel('y-axis'); zlabel('z-axis');
              hold on
        end


        for iatom=NMET:NATOM_TOTAL
          xxx=xnew(iatom); yyy=ynew(iatom); zzz=znew(iatom);

          if ( (xxx>Lx11 & xxx<Lx2) & (yyy>Ly11 & yyy<Ly2) & (zzz>Lz11 & zzz<Lz2))
             plot3(xxx,yyy,zzz,'ro');
          else
%          plot3(xxx,yyy,zzz,'o');
          end
        end


        hold off


        fidin=fopen('molecule_grid.txt','w');
              for ii=1:NATOM_TOTAL
              fprintf(fidin,'%5.1f          %12.5f          %12.5f          %12.5f          %10.1f
\n',ii,xnew(ii),ynew(ii),znew(ii),atomid(ii));
              end
              fclose(fidin);

        % center postion of the polymer unit
```

```
xc=0; yc=0; zc=0;

% dimensions of cube
Lx=28.2*5;
Ly=28.2*5;
Lz=28.2*5;


%distance between molecules within polymer
lc=2.0;

% allowable minimum distance between any two atoms
toler = 2.0;

%Increments in each direction
Nx=6;  % no. of polymer chains in x-dir
Ny=6;  % no. of polymer chains in y-dir
Nz=6;  % no. of polymer chains in z-dir

% number of molecules (bead) in a chain

N=12;


%Spacing in each direction
dx = Lx/Nx;
dy = Ly/Ny;
dz = Lz/Nz;

%internal volume
Lx1=(1/5*Lx+1/4*28.2)+dx/8
Ly1=(1/5*Ly+1/4*28.2)+dy/8;
Lz1=(1/5*Lz+1/4*28.2)+dz/8;
Lx2=(4/5*Lx-1/4*28.2)+dx/8
Ly2=(4/5*Ly-1/4*28.2)+dy/8;
Lz2=(4/5*Lz-1/4*28.2)+dz/8;

Leng = Lx2-Lx1

Lx11=Lx1+dx/10;
Ly11=Ly1+dy/10;
Lz11=Lz1+dz/10;

%NATOM=N*Nx*Ny*Nz
%NMOLE=Nx*Ny*Nz;
NATOM=0;
NMOLE=0;

%Loop for seed points for molecular positions
ij=0;
dummy=0;
for k=1:Nz
   for j=1:Ny
      for i=1:Nx
            xx=(i-1)*dx + dx/2 + xc;
            yy=(j-1)*dy + dy/2 + yc;
```

33

```matlab
        zz=(k-1)*dz + dz/2 + zc;
      if ( (xx>Lx1 & xx<Lx2) & (yy>Ly1 & yy<Ly2) & (zz>Lz1 & zz<Lz2) );
         dummy=dummy+1;
      else
         NMOLE=NMOLE+1;
         ij=ij+1;
         x(ij)=(i-1)*dx + dx/2 + xc;
         y(ij)=(j-1)*dy + dy/2 + yc;
         z(ij)=(k-1)*dz + dz/2 + zc;
      end
    end
  end
end

NATOM=NMOLE*N

figure(1)

%Create FCC

%       Assign initial positions to id 1 for the first 256 atoms
%   based on face-centered cubic lattice total atoms
id=21;
jatom=0;
FCC = 260;
%....    set lattice distance based on cube of side + CUBE
for i_mole=1:FCC
    jatom = jatom + 1;
    NUNIT1=4;
    NUNIT2=4;
    NUNIT3=4;
    DIST = 5.8*2;
%     DIST = 17.625;
    atomid(jatom) = id;
%....    assign positions of first four atoms
    xnew(jatom) =  Lx11 ;
    ynew(jatom) =  Ly11 ;
    znew(jatom) =  Lz11 ;
    xnew(jatom+1) = Lx11;
    ynew(jatom+1) = Ly11+DIST;
    znew(jatom+1) = Lz11+DIST;
    xnew(jatom+2) = Lx11+DIST;
    ynew(jatom+2) = Ly11;
    znew(jatom+2) = Lz11+DIST;
    xnew(jatom+3) = Lx11+DIST;
    ynew(jatom+3) = Ly11+DIST;

    znew(jatom+3) = Lz11;
   plot3(xnew,ynew,znew,'y^')
   hold on
% ....    replicate first four positions over NUNITS
  for I=1: NUNIT1
     for J=1: NUNIT2
       for K=1: NUNIT3
          for IJ=1:4
             xnew(jatom+4) = xnew(IJ) + 2*DIST*(K-1);
```

```matlab
                ynew(jatom+4) = ynew(IJ) + 2*DIST*(J-1);
                znew(jatom+4) = znew(IJ) + 2*DIST*(I-1);
              end
              plot3 (xnew,ynew,znew,'g*')
              hold on
            end
        end
    end

  end
%Generate Random numbers for specified point


for imole=1:NMOLE
  % create the first atom's position of each chain
  id=id+1;
  x1=x(imole); y1=y(imole); z1=z(imole);
  rn=rand(1,3);
  rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
  jatom = jatom + 1;
  atomid(jatom)=id;
  xnew(jatom)=x(imole) + dx*rn1(1,1)*0.5;
  ynew(jatom)=y(imole) + dy*rn1(1,2)*0.5;
  znew(jatom)=z(imole) + dz*rn1(1,3)*0.5;

  % create the remaining (N-1) atoms' positions
   for iatom=2:N
     iflag=0;
     jatom = jatom + 1;
     rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
     mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
     uvx=rn1(1,1)/mag;
     uvy=rn1(1,2)/mag;
     uvz=rn1(1,3)/mag;
     atomid(jatom)=id;
     xnew(jatom) = xnew(jatom-1) + uvx*lc;
     ynew(jatom) = ynew(jatom-1) + uvy*lc;
     znew(jatom) = znew(jatom-1) + uvz*lc;

     while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
         (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
         (znew(jatom)>Lz1 & znew(jatom)<Lz2))

       iflag = iflag+1;

       if (iflag > 100)
          break
       end

     rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
     mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
     uvx=rn1(1,1)/mag;
     uvy=rn1(1,2)/mag;
     uvz=rn1(1,3)/mag;
     atomid(jatom)=id;
     xnew(jatom) = xnew(jatom-1) + uvx*lc;
```
35

```matlab
    ynew(jatom) = ynew(jatom-1) + uvy*lc;

    znew(jatom) = znew(jatom-1) + uvz*lc;
  end
end

  %Check whether two atoms' positions are within the tolerance

 if (imole==1)
   for ipatom=2:jatom
      testid=1;
      iter=1;
      while (testid > 0.1 & iter <= 50 )
      iter=iter+1;
      testid=0;
      for itest=1:(ipatom-1)
        dist= sqrt((xnew(ipatom)-xnew(itest))^2+(ynew(ipatom)- ynew(itest))^2+(znew(ipatom)-
                          znew(itest))^2);
        if dist < toler;
          testid=testid+1;
         end
        end

      if (testid > 0.1)
        rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;

        % Magnitude
        mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2);
        uvx=rn1(1,1)/mag;
        uvy=rn1(1,2)/mag;
        uvz=rn1(1,3)/mag;
        xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
        ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
        znew(ipatom) = znew(ipatom-1) + uvz*lc;
        iflag=0;
           while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
               (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
               (znew(jatom)>Lz1 & znew(jatom)<Lz2))

            iflag=iflag+1;

            if (iflag > 100)
               break
            end
            rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
            mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
            uvx=rn1(1,1)/mag;
            uvy=rn1(1,2)/mag;
            uvz=rn1(1,3)/mag;
            atomid(jatom)=id;
            xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
            ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
            znew(ipatom) = znew(ipatom-1) + uvz*lc;
          end
        end
```

```
        end
      if (iter > 49)
         break
      end
   end
else
   for ipatom=(imole-1)*N+1:jatom
      testid=1;
      iter=1;
      while (testid > 0.1 & iter < 50 )
      iter=iter+1;
      testid=0;
       for itest=1:(ipatom-1)
         dist = sqrt((xnew(ipatom)-xnew(itest))^2+(ynew(ipatom)- new(itest))^2+(znew(ipatom)
                -znew(itest))^2);
         if dist < toler;
          testid=testid+1;
         end
        end

       if (testid > 0.1)
        rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;

        %Magnitude
        mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2);

        uvx=rn1(1,1)/mag;
        uvy=rn1(1,2)/mag;
        uvz=rn1(1,3)/mag;
        xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
        ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
        znew(ipatom) = znew(ipatom-1) + uvz*lc;
        iflag=0;

           while((xnew(jatom)>Lx1 & xnew(jatom)<Lx2) & ...
              (ynew(jatom)>Ly1 & ynew(jatom)<Ly2) & ...
              (znew(jatom)>Lz1 & znew(jatom)<Lz2))

            iflag=iflag+1;

            if (iflag > 100)
               break
            end

            rn=rand(1,3); rn1(1,1)=rn(1,1)-0.5; rn1(1,2)=rn(1,2)-0.5; rn1(1,3)=rn(1,3)-0.5;
            mag=sqrt(rn1(1,1)^2 + rn1(1,2)^2 + rn1(1,3)^2); %magnitude
            uvx=rn1(1,1)/mag;
            uvy=rn1(1,2)/mag;
            uvz=rn1(1,3)/mag;
            atomid(jatom)=id;
            xnew(ipatom) = xnew(ipatom-1) + uvx*lc;
            ynew(ipatom) = ynew(ipatom-1) + uvy*lc;
            znew(ipatom) = znew(ipatom-1) + uvz*lc;
          end

     end
```

```
        end
          if (iter > 49)
             break
          end
      end
    end
          istart=jatom-N+1;
          iend=jatom;
          plot3(xnew(istart:iend),ynew(istart:iend),znew(istart:iend))
%           axis([0 Lx 0 Ly 0 Lz])
          xlabel('x-axis'); ylabel('y-axis'); zlabel('z-axis');
          hold on
end

%Plot to see internal cube
   j_atom=1; j_atom2=1; j_atom3=1; j_atom4=1; j_atom5=1; j_atom6=1;
   mole = 1;
   distance = 4.05*2;
   internal_natom=0;
for mole = (1:9)
   %....   assign positions of first four atoms
          xmole(j_atom) =  Lx1 + 6*2;
          ymole(j_atom) =  Ly1 + 6*2;
          zmole(j_atom) =  Lz1 + 6*2;

          xmole(j_atom+1) = xmole(j_atom);
          ymole(j_atom+1) = ymole(j_atom);
          zmole(j_atom+1) = zmole(j_atom);

          xmole(j_atom+2) = xmole(j_atom+1);
          ymole(j_atom+2) = ymole(j_atom+1);
          zmole(j_atom+2) = zmole(j_atom+1);

          xmole(j_atom+3) = xmole(j_atom+2);
          ymole(j_atom+3) = ymole(j_atom+2);
          zmole(j_atom+3) = zmole(j_atom+2);

          xmole(j_atom+4) = xmole(j_atom+3);
          ymole(j_atom+4) = ymole(j_atom+3);
          zmole(j_atom+4) = zmole(j_atom+3);

          xmole(j_atom+5) = xmole(j_atom+4);
          ymole(j_atom+5) = ymole(j_atom+4);
          zmole(j_atom+5) = zmole(j_atom+4);

          xmole(j_atom+6) = xmole(j_atom+5);
          ymole(j_atom+6) = ymole(j_atom+5);
          zmole(j_atom+6) = zmole(j_atom+5);

          xmole(j_atom+7) = xmole(j_atom+6);
          ymole(j_atom+7) = ymole(j_atom+6);
          zmole(j_atom+7) = zmole(j_atom+6);

          xmole(j_atom+8) = xmole(j_atom+7);
          ymole(j_atom+8) = ymole(j_atom+7);
          zmole(j_atom+8) = zmole(j_atom+7);
```

```matlab
%              xmole(j_atom+9) = xmole(j_atom+8);
%              ymole(j_atom+9) = ymole(j_atom+8);
%              zmole(j_atom+9) = zmole(j_atom+8);

            plot3(xmole,ymole,zmole, 'k.');
            hold on
    for IM=1:9
      for JM=1:9
        for KM=1:9
          for IJM=1:9
             internal_natom=internal_natom+1;
              xmole(j_atom+9) = xmole(IJM) + distance*(KM-1);
              ymole(j_atom+9) = ymole(IJM) + distance*(JM-1);
              zmole(j_atom+9) = zmole(IJM) + distance*(IM-1);
           end
           plot3 (xmole,ymole,zmole,'k.')
           hold on
         end
       end
       end
 end
 internal_natom
for iatom=1:NATOM
  xxx=xnew(iatom); yyy=ynew(iatom); zzz=znew(iatom);

  if ( (xxx>Lx11 & xxx<Lx2) & (yyy>Ly11 & yyy<Ly2) & (zzz>Lz11 & zzz<Lz2))
     plot3(xxx,yyy,zzz,'ro');
  else
%       plot3(xxx,yyy,zzz,'o');
  end
end


hold off


fidin=fopen('molecule_grid.txt','w');
      for ii=1:NATOM
        fprintf(fidin,'%5.1f %12.5f %12.5f %12.5f %10.1f
                \n',ii,xnew(ii),ynew(ii),znew(ii),atomid(ii));
      end
      fclose(fidin);
```

THIS PAGE INTENTIONALLY LEFT BLANK

B.   MOLECULAR DYNAMICS

```
%          filename:  mdss.m       version 5.5
%
%
%      SSFLAG:  input of most material data
%      SSPARM:  chnage of some parameters (RDEL, VMAX, VDEL) only if necessary
%      SSATOM:  input of atomic or moclecular masses
%          atomid < 10: metals using EAM,
%          10 < atomid < 20: Hydrocarbon using Brenner potential,
%          atomid > 20: polymer using Lennard-Jones potential
%
%
%
    clear
    clc
    global X Y Z X1 Y1 Z1 X2 Y2 Z2 X3 Y3 Z3 X4 Y4 Z4 X5 Y5 Z5 FX FY FZ;
    global IN RL LIST NPOINT NGOFR GR LG;
    global ALFA0 ALFA1 ALFA3 ALFA4 ALFA5;
    global fid1 fid2 fid3 fid4;
    global propty
    global P11 P12 P13 P22 P23 P33;
    global nrin drin nrhoin drhoin;
    global frhoin zrin rhorin rhor frho frhoar frhoar1 frhoar2 frhoar3 frhoar4 frhoar5 frhoar6;
    global rhorar rhorar1 rhorar2 rhorar3 rhorar4 rhorar5 rhorar6 z2rar z2rar1 z2rar2 z2rar3 z2rar4
           z2rar5 z2rar6;
    global nmat atomid amass radcut;
    global STRSXX STRSXY STRSXZ STRSYY STRSYZ STRSZZ
    global dimsz1 dimsz2 dimsz3 dimsz4

    dimsz1=500000;
    dimsz2=30000;
    dimsz3=450000;
    dimsz4=510;
    X=zeros(dimsz1,1);  Y=zeros(dimsz1,1);  Z=zeros(dimsz1,1);
    X1=zeros(dimsz1,1); Y1=zeros(dimsz1,1); Z1=zeros(dimsz1,1);
    X2=zeros(dimsz1,1); Y2=zeros(dimsz1,1); Z2=zeros(dimsz1,1);
    X3=zeros(dimsz1,1); Y3=zeros(dimsz1,1); Z3=zeros(dimsz1,1);
    X4=zeros(dimsz1,1); Y4=zeros(dimsz1,1); Z4=zeros(dimsz1,1);   radcut=zeros(50,1);
    X5=zeros(dimsz1,1); Y5=zeros(dimsz1,1); Z5=zeros(dimsz1,1);   amass=zeros(50,1);
    FX=zeros(dimsz1,1); FY=zeros(dimsz1,1); FZ=zeros(dimsz1,1);   atomid=zeros(dimsz1,1);
    IN=zeros(7,1);      RL=zeros(26,1);     propty=zeros(50,10); LG=zeros(10,1);
    LIST=zeros(dimsz2,1); NPOINT=zeros(dimsz2,1); NGOFR=zeros(dimsz2,1);
    GR=zeros(dimsz2,1);
    ALFA0=zeros(1,1); ALFA1=zeros(1,1); ALFA3=zeros(1,1); ALFA4=zeros(1,1);
    ALFA5=zeros(1,1);
    frhoin=zeros(dimsz4,2); zrin=zeros(dimsz4,2); rhorin=zeros(dimsz4,2); rhor=zeros(dimsz4,2);
    frho=zeros(dimsz4,2); frhoar=zeros(dimsz4,2); frhoar1=zeros(dimsz4,2);
    frhoar2=zeros(dimsz4,2); rhoar3=zeros(dimsz4,2); frhoar4=zeros(dimsz4,2);
    frhoar5=zeros(dimsz4,2); frhoar6=zeros(dimsz4,2);
    rhorar=zeros(dimsz4,2); rhorar1=zeros(dimsz4,2); rhorar2=zeros(dimsz4,2);
    rhorar3=zeros(dimsz4,2);
    rhorar4=zeros(dimsz4,2); rhorar5=zeros(dimsz4,2); rhorar6=zeros(dimsz4,2);
    z2rar=zeros(dimsz4,2,2);
    z2rar1=zeros(dimsz4,2,2); z2rar2=zeros(dimsz4,2,2); z2rar3=zeros(dimsz4,2,2);
    z2rar4=zeros(dimsz4,2,2);
```

41

```matlab
    z2rar5=zeros(dimsz4,2,2); z2rar6=zeros(dimsz4,2,2);
    P11=zeros(1,1); P12=zeros(1,1); P13=zeros(1,1); P22=zeros(1,1); P23=zeros(1,1);
    P33=zeros(1,1);
    STRSXX=zeros(dimsz1,1); STRSXY=zeros(dimsz1,1); STRSXZ=zeros(dimsz1,1);
    STRSYY=zeros(dimsz1,1);
    STRSYZ=zeros(dimsz1,1); STRSZZ=zeros(dimsz1,1);

    nrin=1; drin=0.0; nrhoin=1; drhoin=0.0;


%     IN(1),KSAMPL),(IN(4),NATOM),(IN(6),NSTEP)
%     EQUIVALENCE(LG(1),LENER),(LG(2),LSCALE)
%     EQUIVALENCE (RL(16),STPSQH)

%.... initialization

    fid1=fopen('result1.txt','w');
    fid2=fopen('result2.txt','w');
    fid3=fopen('result3.txt','w');
    fid4=fopen('result4.txt','w');  % stress for each atom
    [KRDF,KWRITE,MAXSTP,MAXEQB,NDEAD,DENER]=SSFLAG;


    SSPARM;
%  SSFCC;
    SSATOM;
    SSPOTN;
    SSIVEL;
    SSEVAL;
    SSX2SC;

    KSAMPL=IN(1);
    NATOM=IN(4);
    STPSQH=RL(16);
    LENER=LG(1);
    LSCALE=LG(2);

%.... equilibration
    for NSTEP = 1:MAXEQB
     IN(6)=NSTEP;  NSTEP
     SSPDCT;
     SSEVAL;
     SSCORR;
     SSPRTY;
        if (rem(NSTEP,KWRITE)==0)
          SSMNTR(1);
      end
       if (NSTEP < (MAXEQB-NDEAD))
          if (LSCALE < 0 | LENER < 0)
          SSCALE(DENER);
        end
      end
    end

    SSRSET
```

```
%.... production
    for NSTEP = 1: MAXSTP
     IN(6)=NSTEP;  NSTEP
    SSPDCT;
    SSEVAL;
    SSCORR;
    if (rem(NSTEP,KSAMPL)==0)
      SSPRTY;
    end
    if (rem(NSTEP,KWRITE)==0)
      SSMNTR (KSAMPL);
    end
    if (rem(NSTEP,KRDF)==0)
      SSGOFR;
    end
    end

    fclose(fid1);
    fclose(fid2);
    fclose(fid3);
```

THIS PAGE INTENTIONALLY LEFT BLANK

C. AUTOCORRELATION FUNCTION

```
clc
clear
nt=21;
    fidinat=fopen('stress.txt');
    stress=fscanf(fidinat,'%g %g %g %g %g', [7,nt]);
    stress=stress';
    fclose(fidinat);
LX=141;
LY=141;
LZ=141;
VOL=LX*LY*LZ;

dt=0.01;

h=1/(2*(nt-1));
Sxy=0; Syz=0; Sxz=0;
Sxy=stress(1,3)^2+stress(nt,3)^2;
Sxz=stress(1,4)^2+stress(nt,4)^2;
Syz=stress(1,6)^2+stress(nt,6)^2;

for i = 2:(nt-1);
   Sxy=Sxy+2*stress(i,3)^2;
   Sxz=Sxz+2*stress(i,4)^2;
   Syz=Syz+2*stress(i,6)^2;
end

Sum=(Sxy+Sxz+Syz)*h/(3*VOL)
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

1.   **Schonhals A, Goering H, Brzezinka K W, and Schick Ch** (ed) 2003   *The molecular dynamics of polymers in random nanometer confined spaces investigated by relaxational and scattering techniques* (Berlin, Germany: Federal Institute of Materials Research and Testing)

2.   **Smith G D, Bedrov D, Li L, and Byutner O** (vol) 117, no. 20 *A molecular dynamics simulation study of the viscoelastic properties of polymer nanocomposites* (Journal of Chemical Physics, 9478, (2002))

3.   **Rapaport, D.C.**, *The Art of Molecular Dynamics,* p. 13, Cambridge University Press, 1995.

4.   **Barsky S J and Plischke M,** *Elastic Properties of Randomly Cross-linked Polymers.* Phys. Rev. E **54**, pp 5370 (1996).

5.   **Daw M.S., Baskes M.I., Bisson C.L., Wolfer W.G.** Sandia Report 86-8660, *Application of the Embedded Atom Method to Fracture, Dislocation Dynamics, and Hydrogen Embrittlement,* pp. 7, March 1986.

6.   **Daw M.S., Baskes M.I., Bisson C.L., Wolfer W.G.** Sandia Report 86-8660, *Application of the Embedded Atom Method to Fracture, Dislocation Dynamics, and Hydrogen Embrittlement,* pp. 8, March 1986.

7.   **Nandan, B , Kandpal, L.D. and Mathur, G.N.,** *Glass transition behavior of poly(ether ether ketone)/poly(aryl ether sulphone) blends: dynamic mechanical and dielectric relaxation studies.* Polymer 44, pp 1267-1369 (2003)

8.   **Jones, R.A.L. and Richards, R.W.,** *Polymers at Surfaces and Interfaces* Cambridge University Press, Cambridge, 1999.

9.   **Karim, A. and Kumar, S.** *Polymer Surfaces, Interfaces and Thin Films,* World Scientific, Singapore, 2000.

10.   **Fleer, G. J., Stuart, C., Scheutjens, J.M.G.M., Cosgrove, T., and Vincent, B.** *Polymers at Interfaces.* Chapman & Hall, London, 1993.

11.   **Rapaport, D.C.**, *The Art of Molecular Dynamics Simulation.*   Cambridge University Press, Cambridge, 1995. pp 222-233.

12.   **Daw M.S., Baskes M.I., Bisson C.L., Wolfer W.G.** Physical Review B vol(29) n. 12, *Embedded Atom Method: Derivation and Application to Impurities, Surfaces, and other Defects in Metals.* pp. 6443-6445, 15 June 1984.

13.   http://cmp.ameslab.gov/cmsn/grestpro.html, August/2003.

BIBLIOGRAPHY

A.   BOOKS

Askerland, D.R., and Phule, P.P., The Science and Engineering of Materials, 4th ed., Brooks/Cole-Thomson Learning, 2003.

Biales, N. M., Mechanical Properties of Polymers, John Wiley & Sons, Inc., New York, 1971.

Flinn, R.A. and Trojan, P.K., Engineering Materials and Their Applications, 2nd ed., Houghton Mifflin Company, 1981.

Rapaport, D.C., The Art of Molecular Dynamics, Cambridge University Press, 1995.

Rosen, S. L., Fundamental Principles of Polymeric Materials, John Wiley & Sons, Inc., New York

Williams, J. G., Stress Analysis of Polymers, John Wiley & Sons, Inc., New York, 1973.

B.   PERIODICALS

Daw, M.S. and Baskes, M.I., "Semiempiracal, Quantum Mechanical Calculation of hydrogen Embrittlement in Metals," Physical Review B, v. 50, n. 17, 25 April 1983.

Daw, M.S. and Baskes, M.I., "Embedded Atom Method: Derivation and Application to Impurities, Surfaces, and Other Defects in Metals," Physical Review B, v. 29, n. 12, 15 June 1984.

Daw, M.S., "Model of Metallic Cohesion: The embedded-Atom Method," Physical Review B, v. 39, n. 11, 15 April 1989-I.

Plimpton, S.J. and Hendrickson, B.A., "Parallel Molecular Dynamics With Embedded Atom Method," MRS Proceedings, 291, 1993.

C.   TECHNICAL REPORTS

Sandia Report 86-8660, Application of the Embedded Atom Method to Fracture, Dislocation Dynamics, and Hydrogen Embrittlement, by M.S. Daw, M.I. Baskes, C.L. Bisson and W.G. Wolfer, March 1986.

Thesis, <u>Determination of Bulk Mechanical Properties of Nanostructures from Molecular Dynamic Simulation</u>, by R. Duff, June 2003.

D.   WEB ADDRESSES

efunda. "Finding the Shear Modulus and Bulk Modulus," [http://www.efunda.com/formulae/solid_mechanics/mat_mechanics/elastic_constants_G_K.cfm]. 1 August 2003

efunda. "Finding Young's Modulus and Poisson's Ratio," [http://www.efunda.com/formulae/solid_mechanics/mat_mechanics/elastic_constants_E_nu_cfm]. 1 August 2003

efunda. "The Stress Tensor (or Stress Matrix)," [http://www.efunda.com/formulae/solid_mechanics/mat_mechanics/stress.cfm] 3 September 2003

"Engineering Material Properties," [http://www.apo.nmsu.edu/Telescopes/SDSS/eng.papers/19950926_ConversionFactors/19950926_Mproperties.html]. 15 April 2003

INITIAL DISTRIBUTION LIST

1.   Defense Technical Information Center
        Ft. Belvoir, VA

2.   Dudley Knox Library
        Naval Postgraduate School
        Monterey, CA

3.   Young W. Kwon
        Professor and Chair
        Mechanical Engineering and Energy Processes Dept.
        Southern Illinois University
        Carbondale, IL