New Open Source Software for Building Molecular Dynamics Systems

Bruce Michael Allen
University of Denver

Follow this and additional works at: https://digitalcommons.du.edu/etd

Recommended Citation
https://digitalcommons.du.edu/etd/747
NEW OPEN SOURCE SOFTWARE FOR BUILDING MOLECULAR DYNAMICS SYSTEMS

A Thesis
Presented to
the Faculty of Engineering and Computer Science
University of Denver

In Partial Fulfillment
of the Requirements for the Degree
Masters of Science

by
Bruce Michael Allen
June 2012
Advisor: Dr. Maciej Kumosa
Abstract

The context of this work is the development of open source software to support researchers to quickly build systems of molecules for molecular dynamics (MD) simulations. The goal is achieved through the integration of three open source programs by judicious modifications and creation of new source code, which allows the creation of molecular models, MD cells and the LAMMPS geometry input files. The software-changes work together supporting an easy and intuitive process for simulation system creation. Creation of multiple MD cells for research simulations becomes quicker and provides needed standardization to the simulation process. The researcher can select from atomistic force fields or modify the software to use other force fields. The software has been validated by generating the glass transition temperature ($T_g$) and the bulk modulus for a cross-linked epoxy system composed of Diglycidyl Ether of Bisphenol-A and Isophorone diamine (DGEBA/IPD). The Large-scale Atomic/Molecular Massively Parallel Simulator (LAMMPS) software is used to conduct the molecule dynamics simulations.
Acknowledgements

Dr. Maciej Kumosa for providing direction and constant support making this work possible in the face of many obstacles –especially March and October 2011

Dr. Paul Predecki for providing the “devil” in the devil’s advocate review of this work, with thanks

Dr. Paul Saxe, Materials Design, Inc. for providing knowledge and perspective on ring perception, atom typing, force fields and software development in the molecular dynamics field

Dr. Craig Knox, Army Research Laboratory, for discussing methods for system quenching and simulated annealing of epoxy systems, and reviewing my early results

Dr. Leandro Martinez, IMECC, for creating the PACKMOL software

Dr. John Carpenter, Cray Research, for creating the MSI2LMP software

Mr. Mark Simms, Nanorex, Inc., for the NanoEngineer-1 software

Mr. Ananyo Bandyopadl, Michigan Technological University, for reviewing my early MD results and providing suggestions

Mr. N. Anderson, AIM Inc., for generating CAR and MDF files from my PDB files for validation testing

Sandia National Laboratory for sponsoring the LAMMPS Workshop, August, 2011

Mr. Bruce Allen for being my dad, for fighting World War II, for being a coal miner and for being a true son of West Virginia
Table of Contents

1. General Introduction .............................................................................................................1

2. Literature Review ..................................................................................................................8
   2.1 Introduction .......................................................................................................................8
   2.2 Background .......................................................................................................................9
   2.3 Force Fields ......................................................................................................................13
   2.4 Target System Definition Diglycidyl ether of bisphenol A and Isophonone (DGEBA/IPD) .................................................................................................................................20
   2.5 Crosslinked Epoxy Systems ............................................................................................23
   2.6 Molecular Dynamics Cell Geometry ................................................................................24
   2.7 Validation Criteria for Open Source Software Performance .........................................28
   2.8 Molecular Dynamics Simulation Procedures ..................................................................30
   2.9 Commercial Software Features ......................................................................................33
   2.10 Open Source Software Features ...................................................................................34
   2.11 Thesis Statement ............................................................................................................36

3. Open Source Software Modifications ..................................................................................37
   3.1 Introduction .......................................................................................................................37
   3.2 NanoEngineer-1 CAD Software Modifications ..............................................................39
   3.3 PACKMOL Software Modifications ..............................................................................42
   3.4 MSI2LMP Software Modifications ................................................................................43
   3.5 Software Validation .........................................................................................................44

4. Molecular Dynamics Simulations .........................................................................................50
   4.1 Simulation setup .................................................................................................................50
   4.2 Molecular Dynamics Cell Configuration .........................................................................52
   4.3 Molecular Mechanics Energy Minimization ...................................................................54
   4.4 System Quench ................................................................................................................54
   4.5 Simulated Annealing .......................................................................................................58
   4.6 Equilibrium and Density Data Collection .......................................................................58
   4.7 Equilibrium and Bulk Modulus Data Collection ............................................................59
   4.8 Simulation Density Results and Glass Transition Temperature ......................................60
   4.9 Simulation Bulk Modulus Results ...................................................................................69

5. Discussion and Conclusions ..................................................................................................71
   5.1 Software Testing ..............................................................................................................71
   5.2 Molecular Dynamics Testing ..........................................................................................71
       5.2.1 Density Results .........................................................................................................71
       5.2.2 Bulk Modulus Results ............................................................................................77

6. Summary and Future Work ...................................................................................................78

7. References ............................................................................................................................81
8. Appendix A – Raw and Simulation Data..............................................................84
9. Appendix B – Software Algorithms ..............................................................84
   Force Field Atom Typing ..............................................................................84
   Bulk Modulus Calculation ..........................................................................84
10. Appendix C – Self-avoiding Walk Software .................................................85
11. Appendix D – Force Field Details ...............................................................98
12. Appendix E – Software Modification Details ...............................................99
13. Appendix F – Acronyms and Symbols .........................................................106
List of Tables

Table 2.1 - CFF91 force field covalent bonding parameters. ..............................................17
Table 2.2 - Software equivalence table substitute chemically equivalent atom types. .......19
Table 2.3 - Second equivalence table not used by MSI2LMP ................................................20
Table 2.4 - 90% Cured epoxy systems at 298K and 1atm (Tack & Ford, 2008), (Wu & Xu, 2006).......................................................................................................................29
Table 4.1 - System two: mass calculations for the 5 Oligomer Ring MD cell. ..........51
Table 4.2 - system 2 mass calculations for the 5 oligomer ring MD cell. .................52
Table 4.3 - Single Round of Simulated Annealing .................................................................58
Table 4.4 - LAMMPS ensembles used to collect density data ........................................59
Table 4.5 - Steps to collect bulk modulus data .................................................................60
Table 4.6 - System 2 range of temperature and density during variation ...............61
Table 4.7 - System two calculated density vs temperature ..............................................61
Table 4.8 - System two simulated vs experimental results ..............................................62
Table 4.9 - System three calculated density vs temperature ...........................................62
Table 4.10 - System four calculated density vs temperature ..........................................62
Table 4.11 - Raw volume data for bulk modulus calculation .........................................69
Table 4.12 - Simulated and experimental bulk modulus ....................................................70
List of Figures

Figure 1.1 – MD Simulation Process Flow .......................................................... 7

Figure 2.1 - Mechanical descriptions of the force field valence terms, cross-coupling terms and Van der Waals (Cerius, 1998) .............................................................. 20

Figure 2.2 - Diglycidyl ether of bisphenol A (DGEBA) Epoxy Monomer Resin ....... 21

Figure 2.3 - Isophorone diamine (IPD) Crosslinker (curing agent). ......................... 21

Figure 2.4 - Synthesis of DGEBA from Bisphenol A ........................................... 21

Figure 2.5 - Amine reaction with DGEBA ............................................................ 22

Figure 3.1 - Beyond Compare file comparison of CAR and MMP file for LAMMPS geometry input file ......................................................................................... 48

Figure 3.2 - Beyond Compare file comparison of CAR and MMP file for LAMMPS geometry input file ......................................................................................... 49

Figure 4.1 - 75% intra-oligomer crosslinked 569 atoms ........................................ 55

Figure 4.2 - System 2: Initial MD cell configuration 2845 atoms ......................... 56

Figure 4.3 - 87.5% intra-oligomer crosslinked 666 atoms .................................. 56

Figure 4.4 - 93.75% Inter-oligomer crosslinking 1138 atoms ............................. 57

Figure 4.5 - System 3 initial MD cell 2845 atoms ............................................. 57

Figure 4.6 - System 4 initial MD cell 3330 atoms ............................................. 58

Figure 4.7 - System 2 least squares 5 point glassy, 6 point viscous Tg calculation .... 64

Figure 4.8 - System 2 least squares 6 point glassy, 5 point viscous Tg calculation .... 65

Figure 4.9 - System 2 least squares 4 point glassy, 7 point viscous Tg calculation .... 66

Figure 4.10 - System 3 least squares 5 point glassy, 6 point viscous Tg calculation .... 67

Figure 4.11 - System 4 least squares 3 point glassy, 8 point viscous T_g calculation .... 68

Figure 4.12 - System 4 longer equilibrium procedure simulation two .................... 69
1.  **General Introduction**

The context of this work is the development of open source software to support researchers in quickly building systems of molecules for molecular dynamics (MD) simulations. The ultimate goal is to turn this proof-of-concept software into releasable software for the MD community to use on a daily basis. The software development lifecycle (SDLC) is composed of gathering requirements, design and coding of software and testing of software against a standard to determine the correctness of the software. Software testing, on a research level, validated the Diglycidyl Ether of bisphenol A and isophorone diamine (DGEBA/IPD) system simulation results against published simulated and experimental results.

Molecule dynamics does for the design of molecular systems what finite element analysis does for building macro structures. Molecular dynamics simulations allow the designer to develop concepts and run “experiments” in the computer: however; as only some attributes of nature can be simulated in a timely fashion calculations do not yet reproduce what occurs in the test tube. The adage, “Garbage in garbage out” applies. Nature never checks herself with computer models and is often cruel towards those who insist on mocking her with computer simulations.

Molecular Dynamics simulations allow researchers to see the interactions at the atomic and molecular scale. Molecular dynamics is the calculation of the atom positions
and velocities in a three-dimensional (3D) space. Movement of atoms in a molecule governs the shape of the molecule. For example, cyclohexane exists in the chair, boat and twisted-boat conformations over time as energy is exchanged with the molecule and its environment. The interactions between molecules are based on nonbonded (e.g. van der Waals and Coulombic) forces. Nonbonded forces are based on atoms attracting and repelling one another based on an equilibrium distance between atoms. As molecules change their conformations the distance between atoms, from one molecule to another, the external forces on all molecules change over time. Molecular dynamics simulations are based on Newton’s equations of motion coupled with a model of how bonded and nonbonded atoms move relative to the forces applied commonly referred to as a force field. The term “force field” is misleading because the “force field” equations really calculate the potential energy of the system of atoms. The physical properties of the system depend on the force field.

Molecules are confined to a 3D-space that is periodic, non-periodic in nature or a combination of the two cases. This work addresses the periodic case. Non-periodic boundary conditions allow research at the surface of liquid and solids with matter in all phases. The volume of a simulation is often referred to as a cell, MD cell, box or system. Molecules are given an equilibrium structure in terms of their potential energy in the gas phase and packed into the MD cell as close as possible without creating large repulsive forces between molecules. Setting pressure and temperature for such a simulation allows the volume of the simulation to change over time. The new calculated volume at 298K and 1atm hopefully will be close to the experimental volume thereby providing realistic density values of condensed matter. Simulations conducted are of a cross-linked epoxy
system, DGEBA/IPD assumed to be in the range of 298K to 600K at 1atm. The high temperatures are mainly required for equilibrating the initial system through simulated annealing and for calculating the glass transition temperature ($T_g$) of DGEBA/IPD, while high pressure (~5000atm) is required to calculate the bulk modulus.

Leveraging open source software (OSS) allows researchers to quickly build molecules and systems of molecules for MD simulations similar to those available in expensive commercial software packages. Significant modification of the open source software was required for this work. Three open source software tools were modified to accomplish this goal: NanoEngineer-1(NE-1), PACKMOL and MSI2LMP. The commercial software companies, accelrys (accelrys, 2012), Materials Design (materials design, 2012), and SCIENCEOMICS (SCIENCEOMICS, 2012) offer their respective products Materials Studio, MedeA and MAPS to quickly accomplish the same for considerable amounts of money and come with significant scientific support and consultation. One of the many goals of this software development effort is to make the software available to the LAMMPS user community via sourceforge (sourceforge, 2012). LAMMPS has developed a significant user-base and many users answer both software and scientific questions on the LAMMPS portal. LAMMPS is short for Large-scale Atomic/Molecular Massively Parallel Simulator developed and maintained by Sandia National Laboratories.

The researcher can select either the published COMPASS (Sun, 1998) or CFF91 (Sun, 1998). Published COMPASS is available in the papers published by Sun and other researchers. COMPASS is short for Condensed-phase Optimized Molecular Potentials for Atomistic Simulation Studies and is owned by accelrys. CFF91 (Maple, Dinur, & Hagler, 1988) is distributed with LAMMPS in the tools directory as well as by accelrys.
CFF91 is short for Consistent Force Field, 1991. Whenever COMPASS is referenced in this work it refers to the published COMPASS force field. There are many other published force fields available for use. Any of the following force fields, AMBER (Flowers, 2012), CHARMM (CHARMM FF Parameters, 2012), OPLS (Wikipedia, OPLS, 2012), DREIDING (Mayo, Olafson, & Goddard III, 1990) and others can be integrated into NE-1. The main purpose of providing COMPASS and CFF91 was to demonstrate the integration of multiple force fields into NE-1. The scientific reason for selecting atomistic force fields for this work is to allow more detailed calculation of physical properties and for future calculation of the mean squared displacement (MSD) of water in a cross-linked epoxy.

LAMMPS is used in this work to simulate the DGEBA/IPD system and is used daily to simulate millions of atoms on systems with thousands of processors. Some LAMMPS simulations run for a year! LAMMPS is executed on an Intel i7 four-core processor and runs in parallel using OpenMPI for this work. Testing with hyper-threading was disappointing as simulation times did not decrease, but may require the USER-OMP package to be installed.

The software heavy processes in purple and the human heavy processes in blue are shown in Figure 1.1. The black arrows show the data flow, in all cases files containing data, from one process/program to another. NanoEngineer-1 uses a file format called molecular machine part file (MMP). NanoEngineer-1, PACKMOL and MSI2LMP (Molecular Simulations, Inc. to LAMMPS – MSI became accelrys) all were modified to understand the modified MMP file format developed for this work. MSI2LMP creates native LAMMPS geometry input files (LGIF). LAMMPS generates thermodynamic
(log), trajectory and restart files. The log file contains thermodynamic data such as temperature, pressure, volume, etc. Trajectory files allow the user to see the movement of molecules over the duration of the simulation since they contain atom positions over time through the use of the Virtual Molecular Dynamics (VMD) open source program. The restart files allow the user to restart a simulation in the event of a power failure. The restart feature saves the user from rerunning the last 6 months of calculations due to a power failure. The human heavy processes are accomplished with a text editor. The red arrows show human decisions and how they affect the input data to Execute LAMMPS process. The user must create LAMMPS command input files (LCIF) by typing them in. LAMMPS has a 929 page user manual detailing commands. The LAMMPS command input file used to equilibrate the DGEBA/IPD system is over 10 pages. The LAMMPS geometry file for a 569 atom oligomer is 210 pages and the 5 oligomer system file is over 2000 pages! The input files and associated data, raw and refined, must be distributed on DVD. The first three purple processes and their associated software represent the scope of this work. LAMMPS execution and the associated human processes in blue are required for all MD simulations and are the same processes the user would carry out using a commercial software package.

Multiple confirmation criteria for validating simulation results are required. In the case of MD calculations viewing the trajectory files generated by LAMMPS provides insight into the correctness of force field information used to create the initial geometry. As an example, running a molecular mechanics (MM) energy minimization will allow a trajectory file to be created, which can be viewed using the VMD software. The trajectory file format is simple and the software modifications to
MSI2LMP generates a trajectory file from NE-1 or PACKMOL data thereby providing another check on geometry structure via the VMD tool. LAMMPS generates trajectory files during simulation when commanded. Initial results in this work showed benzene rings bent in half and other obvious distortions to the DGEBA structure. Obviously, the force field data was not correctly populated into the LAMMPS geometry input file (LGIF). PACKMOL generates the final MD cell geometry file in new MMP file format. This intentional design feature allows the final MD cell geometry to be viewed in NanoEngineer-1 and allows the researcher to manually modify the final MD cell geometry prior to simulation.
Figure 1.1 – MD Simulation Process Flow.
2. Literature Review

2.1 Introduction

There are two major purposes of this literature review. The first is to understand the testing criteria of the software and the second understanding the features of commercial and open source software currently supporting MD simulations though an exhaustive search was not conducted. The software validation or testing criteria consists of two tiers: actual software validation per existing tenants of software engineering and the review of MD simulation results. The software development lifecycle (SDLC) consists of creating requirements, developing a design, coding the software and finally testing the software. The requirements are developed directly from features defined in commercial and open source software. The three selected open source software have existing designs, which will be modified to meet new requirements. Review of the NanoEngineer-1, PACKMOL and MSI2LMP code was a significant part of the literature search.

The testing criteria required review of force fields in terms of how they affect simulation results and how they can be integrated into the software, defining a target epoxy system for simulation in order to confirm the software can model real-world systems, defining crosslinking for the purposes of building an initial system for simulation, defining validation criteria for measuring the open source software’s
performance in terms of calculating physical properties, and identify the necessary simulation procedures that must be applied to the MD simulation to calculate the desired thermodynamic properties.

2.2 Background

Molecular dynamics is the solving of Newton’s equations of motion over a span of time to collect the positions and velocities of the atoms. Atoms translate, rotate and vibrate due to the bond and nonbonded forces inherent in the atoms of a molecule and between molecules. An interesting phenomenon of MD calculations is when atoms are too close together and the repulsive forces cause the atoms to fly away from one another. When all forces on all the atoms individually sum to zero the system of atoms is considered to be at equilibrium (Wu & Xu, 2006). It is critical to not confuse the equilibrium condition with the minimum system energy. The equilibrium condition has many configurations and is defined as zero forces and moments on each atom in the system. Just as a truss system has beams and points where beams are pinned together the same situation occurs in the system of molecules with atoms and bonds, including nonbonded contributions. System minimum energy exists as many minimums on the potential energy surface of the system where bond lengths and angles are the variables in configuration space. There is one global minimum with many local minima.

Equation 2.1 shows the pressure calculation performed by LAMMPS and other MD software for a 3D system. The first term is the kinetic energy and the second term is the virial as discussed by (Wu & Xu, 2006). The number of atoms (N), the Boltzmann constant ($K_B$), temperature (T) and system volume (V) are required for calculating the kinetic energy while the per atom force ($f_i$) and the distance between atom pairs ($r_i$) are
required to calculate the virial. The virial is defined as the derivative of the system internal energy with respect to the strain (Wu & Xu, 2006).

\[ P = \frac{Nk_b T}{V} + \left( \sum_{i}^N r_i \cdot f_i \right)/3V \]

**Equation 2.1 Molecular Dynamics Pressure Calculation**

Other definitions of system equilibrium are defined in the literature to meet the needs of the work conducted. Convergence of the dependent state variable or other property is usually used to define equilibrium. An example would be to define the temperature and pressure of a system and watch the volume of the system converge to a mean value. The problem with atoms moving too far in too short a time is it wastes computation time needed to achieve equilibrium for study. Balancing repulsive and attractive forces per atom is the goal. The natural repulsive and attractive forces are quantified by a force field. Whether covalent or Lennard-Jones in nature the force between atoms is modeled with a polynomial function.

Molecular mechanics (MM) energy minimization is used to create an initial system with a minimum potential energy. Bond lengths, angles, dihedrals (torsion), impropers (out-of-plane) and distances between atoms are adjusted to reach a minimum energy within a molecule. All MD procedures start with MM. Molecular Mechanics energy minimization reduces high bond energy and suffers from the local minimum problem. The minimum energy found is one of many and most likely not the global minimum. LAMMPS provides five minimization methods. The Polak-Ribiere conjugate gradient (CG) provides the added benefit of changing the size of the MD cell as the atoms
are moved when used in conjunction with the LAMMPS “relax box” option to apply an external pressure tensor to the system. This feature helps to bring the system simulation pressure to the user-specified set point. The temperature is assumed to be zero in the Polak-Ribiere method. LAMMPS also minimizes energy based on nonbonded forces between atoms in the system subject to the cutoff specified for atom pairs based on Van der Waals and Coulombic interactions. LAMMPS also implements the fast inertial relaxation engine (FIRE) method to minimize energy. The method uses an NVE (constant number of atoms, volume and energy) ensemble to calculate position, velocity and forces for every atom in the system. Essentially, the system does have a temperature during the FIRE minimization due to the NVE ensemble. System temperature changes in a chaotic manner during this method and, though slow to converge, does provide lower potential energies compared to other methods.

As the MD simulation progresses, energy is exchanged between potential and kinetic energy. Obviously, temperature is another way of looking at the atom velocities of the system. An adiabatic system is isolated from its surroundings and the energy of such a system would ideally remain the same over time. Calculations of such a system fall victim of round off errors and the system energy does change over time. The NVE ensemble is the adiabatic system. When a system is run adiabatically the temperature will converge to some value based on the geometry of the system. Energy can be permanently removed from the system by rescaling the velocities of the atoms. Velocity rescaling is another tool to help prevent atoms from flying away from each other and is performed
after MM energy minimization. Quenching is the rapid cooling of a material. Velocity rescaling is referred to as quenching due to energy being removed from the system.

The constant number of atoms, volume and temperature, NVT, ensemble is used to equilibrate a system to a set point temperature where pressure will be defined in terms of system temperature and volume as if the system were an ideal gas. The virial must also be considered in the pressure calculation of the system as per Equation 2.1. Under NVT the system exchanges energy with a temperature bath. Running the system under the NVE ensemble initially will provide the equilibrium temperature of the system. Knowing the equilibrium temperature then allows one to know if energy will be added to or removed from the system when coupled with the temperature bath using a thermostat to control the flow of energy. The Nose-Hoover thermostat is the default for LAMMPS.

The constant number of atoms, pressure and temperature, NPT, ensemble is used to equilibrate a system to a set point temperature and pressure. The NPT ensemble is used to change the volume of the system. The NPT ensemble is important in the calculation of the system density and the bulk modulus. The NPT ensemble can be thought of as a system connected to a temperature bath for the exchange of energy while the system is maintained in a volume with a movable piston thus allowing the system volume to change over time to maintain a constant system pressure. Temperature and volume are the better behaved state variables in MD simulations while pressure is dependent on system size. “Instantaneous pressure [or stress tensor] of a simulation cell... will have mean squared fluctuations” according to David Case quoting Section 114 of Statistical Physics by Landau and Lifshitz. A system of 60,000 atoms will experience 30bar
instantaneous pressure fluctuations, however, much larger fluctuations are observed in practice. A 54 million atom system is required to reduce the pressure fluctuations to ~1bar. The Nose-Hoover barostat is the default in LAMMPS.

Annealing is the process of heating a material past some critical temperature and then cooling it to some temperature below the critical temperature. In the case of an epoxy the glass transition temperature \( T_g \) is the critical temperature. Annealing, often referred to as simulated annealing, is another process used to prepare a system for MD simulations. Annealing raises the system temperature above \( T_g \) and allows the system to bypass high energy barriers that force MM energy minimization into local minima. A global minimum is not guaranteed by simulated annealing. The rate of heating and cooling can be important in an annealing process (Varshney, Patnaik, Roy, & Farmer, 2008).

2.3 Force Fields
The force field converts geometry into potential energy. The bond lengths, angles, dihedrals, cross-terms, and force-at-a-distance contributions (e.g Van der Waals and Coulombic) are used to compute the potential energy of the system. The consistent family of force fields (CFF) includes CFF91, PCFF and COMPASS. All three were developed by accelrys. CFF91 and PCFF are available for use outside of Materials Studio. (Tack & Ford, 2008) reported system density values closer to experimental values using CFF91 compared to results using COMPASS – see Figure 2.7 (Tack & Ford, 2008). All three
force fields are described with the equations in

\[ E_{pot} = \sum_b [K_2 (b - b_0)^2 + K_3 (b - b_0)^3 + K_4 (b - b_0)^4] \quad (quartic\ bonding) \]

\[ + \sum_{\theta} [H_2 (\theta - \theta_0)^2 + H_3 (\theta - \theta_0)^3 + H_4 (\theta - \theta_0)^4] \]

\[ + \sum_{\varphi} [V_1 [1 - \cos (\varphi - \varphi_1^n)] + V_2 [1 - \cos (2\varphi - \varphi_2^n)] \]

\[ + V_3 [1 - \cos (3\varphi - \varphi_3^n)] + \sum_{\omega} K_\omega \omega^2 + \sum_{b} \sum_{b} F_{bb'} (b - b_0) (b' - b_0) \]

\[ + \sum_{\theta} \sum_{\theta'} F_{\theta\theta'} (\theta - \theta_0) (\theta' - \theta_0') + \sum_{b} \sum_{\theta} F_{b\theta} (b - b_0) (\theta - \theta_0) \]

\[ + \sum_{b} \sum_{\varphi} (b - b_0) [V_1 \cos \varphi + V_2 \cos 2\varphi + V_3 \cos 3\varphi] \]

\[ + \sum_{b} \sum_{\varphi} (b' - b_0') [V_1 \cos \varphi + V_2 \cos 2\varphi + V_3 \cos 3\varphi] \]

\[ + \sum_{\theta} \sum_{\varphi} (\theta - \theta_0) [V_1 \cos \varphi + V_2 \cos 2\varphi + V_3 \cos 3\varphi] \]

\[ + \sum_{\varphi} \sum_{\theta} \sum_{\theta'} \cos \varphi (\theta - \theta_0) (\theta' - \theta_0') \]

Equation 2.2 through Equation 2.7.
Equation 2.2 Consistent Family of Force Fields (CFF) Potential Energy for Covalent Bonding (Cerius, 1998)

\[ E_{pot} = \sum_b [K_2(b-b_0)^2 + K_3(b-b_0)^3 + K_4(b-b_0)^4] \text{ (quartic bonding)} \]

\[ + \sum_\theta [H_2(\theta-\theta_0)^2 + H_3(\theta-\theta_0)^3 + H_4(\theta-\theta_0)^4] \]

\[ + \sum_\varphi [V_1[1 - \cos(\varphi - \varphi_1^0)] + V_2[1 - \cos(2\varphi - \varphi_2^0)] \]

\[ + V_3[1 - \cos(3\varphi - \varphi_3^0)] + \sum_\omega K_\omega \omega^2 + \sum_b \sum_b F_{bb'} (b-b_0)(b'-b_0') \]

\[ + \sum_\theta \sum_\theta' F_{\theta\theta'} (\theta - \theta_0)(\theta' - \theta_0) + \sum_b \sum_\theta F_{b\theta'} (b-b_0)(\theta - \theta_0) \]

\[ + \sum_b \sum_\varphi (b-b_0) [V_1\cos\varphi + V_2\cos2\varphi + V_3\cos3\varphi] \]

\[ + \sum_b \sum_\varphi (b'-b_0') [V_1\cos\varphi + V_2\cos2\varphi + V_3\cos3\varphi] \]

\[ + \sum_\theta \sum_\varphi (\theta - \theta_0) [V_1\cos\varphi + V_2\cos2\varphi + V_3\cos3\varphi] \]

\[ + \sum_\varphi \sum_\theta \sum_\theta' \cos\varphi (\theta - \theta_0)(\theta' - \theta_0') \]
\[ E_{pot} = \sum_{\theta} \left[ K_2 (\theta - \theta_0)^2 + K_3 (\theta - \theta_0)^3 + K_4 (\theta - \theta_0)^4 \right] \] (quartic bonding)

\[ + \sum_{\theta} \left[ H_2 (\theta - \theta_0)^2 + H_3 (\theta - \theta_0)^3 + H_4 (\theta - \theta_0)^4 \right] \]

\[ + \sum_{\phi} \left[ V_1 [1 - \cos (\phi - \phi_1)] + V_2 [1 - \cos (2\phi - \phi_2)] \right] \]

\[ + V_3 [1 - \cos (3\phi - \phi_3)] \] + \sum_{\omega} K_{\omega} \omega^2 + \sum_{b} \sum_{b} F_{b\theta} (b - b_0)(b' - b'_0) \]

\[ + \sum_{\theta} \sum_{\theta'} F_{\theta\theta'} (\theta - \theta_0)(\theta' - \theta_0) + \sum_{b} \sum_{\theta} F_{b\theta} (b - b_0)(\theta - \theta_0) \]

\[ + \sum_{b} \sum_{\phi} (b - b_0) [V_1 \cos \phi + V_2 \cos 2\phi + V_3 \cos 3\phi] \]

\[ + \sum_{b} \sum_{\phi} (b' - b'_0) [V_1 \cos \phi + V_2 \cos 2\phi + V_3 \cos 3\phi] \]

\[ + \sum_{\phi} \sum_{\theta} [V_1 \cos \phi + V_2 \cos 2\phi + V_3 \cos 3\phi] \]

\[ + \sum_{\phi} \sum_{\theta} \sum_{\theta'} \cos \phi (\theta - \theta_0)(\theta' - \theta_0) \]

Equation 2.2 details the potential energy of valence bonded atoms in a molecule and Equation 2.3 through Equation 2.7 defines the potential energy for nonbonded interactions. Figure 2.1 shows the mechanical behavior of the valence and nonbonded
potential energy terms. The potential energy expression in

$$E_{pot} = \sum_b \left[K_2(b - b_0)^2 + K_3(b - b_0)^3 + K_4(b - b_0)^4\right] \text{(quartic bonding)}$$

$$+ \sum_{\theta} \left[H_2(\theta - \theta_0)^2 + H_3(\theta - \theta_0)^3 + H_4(\theta - \theta_0)^4\right]$$

$$+ \sum_{\phi} \left[V_1[1 - \cos(\phi - \phi_0)] + V_2[1 - \cos(2\phi - 2\phi_0)]\right]$$

$$+ V_3[1 - \cos(3\phi - 3\phi_0)] + \sum_\omega K_\omega \omega^2 + \sum_b \sum_b F_{bb}(b - b_0)(b' - b'_0)$$

$$+ \sum_\theta \sum_\theta F_{\theta\theta}(\theta - \theta_0)(\theta' - \theta'_0) + \sum_\theta \sum_\theta F_{\theta\theta}(b - b_0)(\theta - \theta_0)$$

$$+ \sum_b \sum_\phi (b - b_0)[V_1\cos\phi + V_2\cos2\phi + V_3\cos3\phi]$$

$$+ \sum_b \sum_\phi (b' - b'_0)[V_1\cos\phi + V_2\cos2\phi + V_3\cos3\phi]$$

$$+ \sum_\theta \sum_\phi (\theta - \theta_0)[V_1\cos\phi + V_2\cos2\phi + V_3\cos3\phi]$$

$$+ \sum_\theta \sum_\theta \sum_\theta \cos\phi(\theta - \theta_0)(\theta' - \theta'_0)$$

Equation 2.2 accounts for bond lengths ($b$), angles ($\theta$), dihedrals ($\phi$), out-of-plane angles ($\chi$) as well as cross-coupling terms all of which are valence terms. Cross-coupling terms are important for predicting vibration frequencies and structural variations due to conformational changes (Sun, 1998). All force field equations contain equilibrium bond lengths and angles where the potential energy contribution becomes zero. For example equilibrium bond lengths ($b_0$), angles ($\theta_0$), dihedrals ($\phi_0$), out-of-plane angles ($\chi_0$) values are specific to the nature of the participating atoms. The equilibrium bond length ($b_0$)
between a hydrogen and a carbon atom will be different than the same value \((b_{ij})\) between two carbon atoms.

Equation 2.3 accounts for the Coulombic energy \((E_{elec})\) and Equation 2.5 accounts for the Van der Waals \((E_{vdW})\) energy. Equation 2.4 details the calculation of atom charges \((q_i)\) calculated per atom \((i)\) by summing the bond increments \((\delta_{ij})\) for the atoms \((j)\) valence bonded to atom \((i)\).

\[
E_{elec} = \sum_{i>j} \left( q_i q_j / r_{ij} \right)
\]

**Equation 2.3** Coulombic contribution to the potential energy where \(q_i\) is the charge on atom \(i\) (Cerius, 1998)

\[
q_i = \sum \delta_{ij}
\]

**Equation 2.4** Total calculated atom charge based on summation of bond increments (Cerius, 1998)

Equation 2.5 accounts for the Van der Waals energy and uses a soft Lennard-Jones 9-6 function. The nonbonded terms are used for interactions between pairs of atoms separated by two or more atoms and for atoms in different molecules (Cerius, 1998).

\[
E_{vdW} = \sum E_{ij} \left[ 2 \left( \frac{r^0_{ij}}{r_{ij}} \right)^9 - 3 \left( \frac{r^0_{ij}}{r_{ij}} \right)^6 \right]
\]

**Equation 2.5** Van der Waals potential energy contribution (Cerius, 1998)

Equation 2.6 and Equation 2.7 define the 6\(^{th}\)-order combination rules that are used to calculate \(r_{ij}\) values from \(r_i\) values, which are parameters calculated for the force field and
are maintained in a table for a force field. Materials Studio and LAMMPS calculate $E_{\text{elec}}$ and $E_{\text{vdW}}$ using the Equation 2.3 through Equation 2.7.

\[
r_{ij}^0 = \left( \frac{(r_i^0)^6 + (r_j^0)^6}{2} \right)^{\frac{1}{6}}
\]

**Equation 2.6 Off diagonal atom pair parameters for Van der Waals calculation (Cerius, 1998)**

\[
E_{ij} = 2 \sqrt{E_i E_j} \left( \frac{(r_i^0)^3 + (r_j^0)^3}{(r_i^0)^6 + (r_j^0)^6} \right)
\]

**Equation 2.7 6\textsuperscript{th}-order mixing potential energy for unlike atom pairs (Cerius, 1998)**

Force fields, from a software perspective, are large text files of sequential data organized around atom types, bonds, angles, dihedrals, Wilson out-of-plane, cross-coupling terms, Van der Waals and Coulombic parameter values corresponding to the
As an example a generic sp³ carbon is bonded to an aromatic carbon, their atom types are “c” and “cp” respectively. In the case of the bond between the two carbons the force field parameters are shown in Table 2.1.

Table 2.1 - CFF91 force field covalent bonding parameters.
Covalent Bonding

<table>
<thead>
<tr>
<th>Version</th>
<th>Reference</th>
<th>I</th>
<th>J</th>
<th>(b_0) (Å)</th>
<th>(K_2) (Kcal/Å²)</th>
<th>(K_3) (Kcal/Å³)</th>
<th>(K_4) (Kcal/Å⁴)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>1</td>
<td>c</td>
<td>cp</td>
<td>1.5010</td>
<td>321.9021</td>
<td>-521.8208</td>
<td>572.1628</td>
</tr>
</tbody>
</table>

The versions of the parameters are maintained so that new parameters can replace the previous version but older simulations can be repeated using the previous parameters. The reference identifies the publication and author who calculated the parameters. The atom types are given along with the parameters. The parameters in Table 2.1 match the coefficients shown in Equation 2.2 for the quartic bonding term. Table 2.1 shows two atom types common in the DGEBA molecule. Each unique pair of covalently bonded atoms is given a unique bond type in LAMMPS thus allowing the force field parameters shown to be used in a simulation. The parameters are written to the LAMMPS geometry input file (LGIF) prior to the simulation by the MSI2LMP software.

Researchers either assign atom types to atoms in a molecule or allow software to do so. If software assigns atom types the researcher should review the assignments. Every pair of covalently bonded atoms has a bond type defined by the two atom types assigned. Every three covalently bonded atoms have an angle type and for every four covalently bonded atoms has a dihedral angle type. An atom type may cover a broad category such as a generic sp³ Carbon or a specific aromatic Carbon “cp”. Atom type assignment on the surface appears to be straight forward. Sun et. al. went further in defining an equivalence table, which substitutes one atom type for another based on the environment of the atom. The categories are nonbonded, bond, angle, dihedral and out-
of-plane. Table 2.2 shows examples from the COMPASS equivalence force field table (Sun, 1998). The CFF91 force field also contains such a table.
Table 2.2 - Software equivalence table substitute chemically equivalent atom types.

<table>
<thead>
<tr>
<th>Atom Type</th>
<th>Nonbonded</th>
<th>Bond</th>
<th>Angle</th>
<th>Dihedral</th>
<th>Out-of-plane</th>
</tr>
</thead>
<tbody>
<tr>
<td>c3a</td>
<td>c3a</td>
<td>c3a</td>
<td>c3a</td>
<td>c3a</td>
<td>c3a</td>
</tr>
<tr>
<td>c43</td>
<td>c43</td>
<td>c4</td>
<td>c4</td>
<td>c4</td>
<td>c4</td>
</tr>
</tbody>
</table>

According to the data in Table 2.2 an aromatic Carbon (c3a) must have unique parameters for all cases: nonbonded through out-of-plane. The software, MSI2LMP, reports missing parameters to the user. In the case of a Carbon attached to three heavy atoms (c43) it must have unique parameters for the nonbonded case where the other atom is identical. In all other cases the parameters for the generic Carbon (c4) will be used (Sun, 1998). The researcher must be concerned with the software replacing atom types. It is up to the researcher to detect such replacements and decide if the replacement will work as expected. The replacement of critical atom types could lead to unexpected results. Literature researched for this work did not address such concerns.

A second equivalence table, shown in Table 2.3, is also part of CFF91 and addresses atom type substitutions for cross-coupling terms and is used by Materials Studio, but MSI2LMP does not use it. The table contains numbers and symbols that may be used by MS. Being proprietary may make it difficult to integrate into MSI2LMP and may affect the simulation results. Tack and Ford did simulate a five oligomer system using MS with unpublished COMPASS and a 10 oligomer system using LAMMPS with CFF91 (Tack & Ford, 2008). If they had simulated identical systems using the same force field under MS and LAMMPS one could make inferences about how MS handles the missing cross-coupling terms reported by MSI2LMP.
Table 2.3 - Second equivalence table not used by MS12LMP.

<table>
<thead>
<tr>
<th>Atom type</th>
<th>NonB</th>
<th>Bond Inc</th>
<th>Bond</th>
<th>Angle End Atom</th>
<th>Angle Apex Atom</th>
<th>Dihedral End Atom</th>
<th>Dihedral Center Atom</th>
<th>OOP End Atom</th>
<th>OOP Center Atom</th>
</tr>
</thead>
<tbody>
<tr>
<td>C=</td>
<td>C=</td>
<td>C=</td>
<td>C=_3</td>
<td>C=_</td>
<td>C=_</td>
<td>C=_</td>
<td>C=_</td>
<td>C=</td>
<td>C=</td>
</tr>
</tbody>
</table>

Figure 2.1 - Mechanical descriptions of the force field valence terms, cross-coupling terms and Van der Waals (Cerius, 1998).

2.4 Target System Definition Diglycidyl ether of bisphenol A and Isophonone (DGEBA/IPD)

The molecular structures of DGEBA and IPD are shown in Figures 2.2 and 2.3 respectively. The crosslinked epoxy system created from a large collection of DGEBA monomers and crosslinking molecule, IPD, is based on the reaction of the epoxide ring at the two reaction-sites per nitrogen in the IPD structure.
Figure 2.2 - Diglycidyl ether of bisphenol A (DGEBA) Epoxy Monomer Resin.

Figure 2.3 - Isophorone diamine (IPD) Crosslinker (curing agent).

Figure 2.4 shows the reaction between bisphenol A and an epoxide to form the DGEBA monomers or resin. A 100% crosslinked system will contain completely reacted IPD molecules attached to four unique DGEBA molecules. Typically R is –CH$_2$Cl as in epichlorohydrin, then the by-product is HCl. An excess of epichlorohydrin is used to limit the extent of reactions of the type shown in Figure 2.4 (Predecki, 2012).

Figure 2.4 - Synthesis of DGEBA from Bisphenol A.
The geometry of such a network could theoretically go on forever in all three dimensions. Each hydrogen atom per nitrogen on the IPD molecule provides an identical reaction site to react with an epoxide ring on the DGEBA molecule: however; once a DGEBA molecule reacts at one site the reaction rate constant at the sister site will be different for the second reaction. Also, the reaction rate differs per nitrogen due to one being attached directly to the ring and the other is attached to a methyl group attached to the ring. Figure 2.5 shows the reaction between the monomer and the crosslinker.

![Figure 2.5 - Amine reaction with DGEBA.](image)

In Figure 2.6 a secondary reaction does occur between DGEBA molecules to form short polymer chains (Tack & Ford, 2008). The reaction requires one DGEBA monomer to be partially formed thus having a hydroxyl group to react with a completely formed DGEBA.
Figure 2.1 Secondary reaction: DGEBA with Bisphenol A.

2.5 Crosslinked Epoxy Systems

Crosslinking results in an epoxide ring on the DGEBA molecule reacting with an amine group on the IPD molecule. If every one of the four reactive amine sites react with a DGEBA molecule then the cross linking is defined as 100%. A 3D network of DGEBA molecules connected to IPD molecules are formed in the real world. An amorphous network of chains running out from a single IPD molecule would be entirely covalent in bonding. Since many IPD molecules are present in the curing of the DGEBA resin multiple such networks are formed and are intertwined. Such an intertwined collection of networks requires a full atom model where Van der Waals, Coulombic and valence force field data is available. Materials Studio and LAMMPS support the full atom model. In the real world the network spans far longer distances than can be simulated as most simulation MD cells are reported in the 20Å to 100Å range. Computational capability limits the MD cell size.

Another approach to crosslinking was raised by Tack and Ford based on information from Hexion Specialty Chemicals where an oligomer of DGEBF and
DETDA was constructed on a ring topology (Tack & Ford, 2008). The 3-D network in the oligomer case does not extend to the same distances one expects for the 3-D network described above. Not having the Hexion information in hand makes it difficult to evaluate the chemical or structural physical evidence to support this topology.

Regardless of topology or structure of the system the reactions at the amine were considered by some researchers. Once the reaction occurs with the primary amine it becomes a secondary amine with a slower reaction rate and more steric hindrance around the reaction site. Some researchers went as far as developing models for the crosslinking process in order to simulate the formation of crosslinks. Wu and Xu considered the primary and secondary amines to have the same reactivity (Wu & Xu, 2006). Varshney et. al. looked at different and equal reaction rates for the basis of building system MD cells (Varshney, Patnaik, Roy, & Farmer, 2008). The secondary is reported to by 40% of the primary reaction.

2.6 Molecular Dynamics Cell Geometry

The literature contains three major MD cell creation approaches. The first approach is based molecular mechanics where the potential energy of the system is minimized (Theodorou & Suter, 1985) Amorphous Cell, software by accelrys, is known to use the work of Theodorou and Suter. The second approach simulates a reaction between the monomers or resin and the crosslinker molecule or curing agent. The third approach takes the classic energy minimization problem and makes it a packing problem (Martinez, Andrade, Birgin, & Martinez, 2009). All methods have advantages and disadvantages and hybrid approaches could be created from these.
The first method is the ancestor to the current MM energy minimization included in most software like Materials Studio and LAMMPS. LAMMPS offers five MM options. Theodorou’s and Suter’s method was developed in 1985. It limited molecule configurations to rotations about bonds and did not allow bond lengths to change, but it did address Coulombic and Van der Waals forces in the minimization. Regardless of software tools used, additional software must be written by the researcher to crosslink epoxy monomers. The method is static in nature due to the structure being predefined and the atoms are rotated about bonds to seek a minimum energy.

The second method depends upon executing an MD simulation of resin and curing agent molecules and at some user-defined number of time steps into the simulation a test for resin and curing agent molecule proximity is done to identify molecules most likely to react. A radius of 6Å - 10Å from the reaction site is used to form bonds between the epoxide and the amine. The MD cell energy is minimized after every bond formation (Varshney, Patnaik, Roy, & Farmer, 2008). Varshney et. al. conducted simulations where amine reaction kinetics were assumed to be identical and for a primary amine reaction much faster. This resulted in Varshney simulating all primary reactions occurring prior to any secondary reactions. Additional reaction schemes were conducted based on selecting pseudo random amine reactions with the goal being a final crosslinked epoxy where equilibrium could be achieved with less computational time. Yarovsky and Evans used a dynamic approach as well. The MD cell is subjected to MD until the level of crosslinking is achieved; also the ratio of resin to curing agent also helps to define the percent crosslinking (Yarovsky & Evans, 2002). The force field is expected to keep the
molecules from spearing one another and to prevent two or more atoms from having the same position in the MD cell. It would be interesting to see how many simulations would be required to generate a random oligomer as used by Tack and Ford (Tack & Ford, 2008).

The final method is based on a packing optimization approach where molecules are packed into a box with the constraint that any two atoms in different molecules must remain a user defined distance or more apart. The goal is to avoid high repulsive forces and hopefully reduce the computation time to reach equilibrium. Equation 2.8 shows the optimization cost function for such an optimization. The first term is concerned with the minimum distance requirement between atoms \(d_{tol}\) of different molecules (distances between atoms in the same molecule are constrained by the molecule’s geometry) while the second term is concerned with the meeting constraints (system boundaries, etc) where the constraints are linear with respect to each atom.

\[
\sum_{i=1}^{n_{mol}} \sum_{j=1}^{n_{atom}(i)} \left( \sum_{l=i+1}^{n_{mol}} \sum_{m=1}^{n_{atom}(l)} \max\{0, d_{tot}^2 - ||p^{ij} - p^{lm}\|^2\}^2 \right) \\
+ \sum_{i=1}^{n_{mol}} \sum_{j=1}^{n_{atom}(i)} \left( \sum_{z=1}^{r_{ij}} \max\{0, g^z_{ij}(p^{ij})\}^2 \right)
\]

**Equation 2.8 PACKMOL Cost Function: Minimize \(f(c,0)\) for best packed MD Cell.**

The first term is quadratic due to the distance between atom pairs. The first term is linearized by placing atoms into bins. The binning algorithm drastically reduces the number of atom-to-atom comparisons since atoms in different bins cannot be close enough to violate the distance constraint.
The cost function is continuous and first-order differentiable, which is not the case for the potential energy equation terms shown in Equation 2.3 and Equation 2.5. Specifically, the $E_{\text{vdW}}$ and $E_{\text{coul}}$ terms create the problem for the energy minimization solution in method two when atoms are very close to one another ($d_{\text{tot}} = 0$). The packing optimization approach does away with the MM energy minimization initially by creating an MD cell with no large repulsive forces. The MM energy minimization can then be done later with a better behaved starting geometry.

Whether a static or dynamic method is used all MD cell geometry methods require an MM energy minimization first step. Obviously, initial system configuration is as important as the actual MD simulations themselves and for crosslinked epoxy systems often times require MD simulations to build the initial system configuration. The method used by Tack and Ford to create a system of pre-crosslinked oligomers was adopted to reduce the complexity of testing the software (Tack & Ford, 2008). Private correspondence with the author of PACKMOL resulted in the design of a new feature for PACKMOL. The proposed new feature requires adding software changes to implement a dynamic constraint where molecules could be chained together as to simulate the growth of a network in terms of a polymerization of like molecules or the networking of resin monomers and curing agent molecules as per the DGEBA/IPD system. In terms of DGEBA and IPD; once the DGEBA molecule is placed in the MD cell an IPD molecule would be attached at its Nitrogen atom via a covalent bond to the last carbon atom on one end of the DGEBA molecule. The position of the hydrogen on the carbon would serve as the constraint to place one of the nitrogen atoms in the IPD molecule. This dynamic
constraint would create crosslinks and would create long chain polymers when using the same molecule.

Tack and Ford describe important details of the MS Amorphous Cell software (Tack & Ford, 2008). Once the oligomers were placed in space, Amorphous Cell takes a sub-sample of the total system of oligomers. The sub-sample is cubed-shape and a measure of isotropic behavior is used by the software to identify the best candidate sub-sample. The Tack and Ford work shows one sub-sample (Tack & Ford, 2008). A uniform density appears to be another criterion for sub-sample selection. Selecting a sub-sample also varies the number of atoms in the system. Tack and Ford report a system of approximately 10,000 atoms and the MD cell sizes are not reported. This approach may have impacts on system behavior under MM and MD and most likely reduces strain in the sample allowing it to achieve equilibrium sooner. Also, if the MD cell is populated in a biased fashion in terms of types of atoms then the Coulombic partial charge environment will be dependent upon the atoms selected.

2.7 Validation Criteria for Open Source Software Performance
Table 2.4 shows the experimental results for DGEBA/IPD and DGEBF/DETDA systems at 298K and 1atm. Specifics of the epoxy composition in terms of percent crosslinking were not reported. The density of 1.131gm/cc is reported in other research with DGEBA to IPD ratios of 16:8 while the researcher was constructing an initial crosslinked epoxy of 93% crosslinked. Only resin and curing agent are present in the simulation and the simulations cited in the literature assumed the same composition.
Table 2.4 - 90% Cured epoxy systems at 298K and 1 atm (Tack & Ford, 2008), (Wu & Xu, 2006).

<table>
<thead>
<tr>
<th>Property</th>
<th>DGEBA/IPD (Experimental)</th>
<th>DGEBF/DETDA (Experimental)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bulk Modulus (GPa)</td>
<td>5.01</td>
<td>2.9</td>
</tr>
<tr>
<td>Elastic Modulus (GPa)</td>
<td>4.71</td>
<td>2.56</td>
</tr>
<tr>
<td>Shear Modulus (GPa)</td>
<td>1.75</td>
<td>0.946</td>
</tr>
<tr>
<td>Poisson’s Ratio</td>
<td>--</td>
<td>0.383</td>
</tr>
<tr>
<td>Density (gm/cc)</td>
<td>1.131</td>
<td>1.16</td>
</tr>
</tbody>
</table>

Figure 2.7 shows the expected behavior of an epoxy (DGEBF-DETDA) density vs temperature graph. Specifically, the change in slope denotes the glass transition temperature as Van der Waals forces melt and the epoxy becomes viscous. Figure 2.7 was generated by a series of simulations using MS using the unpublished COMPASS force field and LAMMPS using the CFF91 force field (Tack & Ford, 2008).

Figure 2.2 DGEBF-DETDA System Density vs Temperature (Tack & Ford, 2008)
2.8 Molecular Dynamics Simulation Procedures

Repeatable experimental procedures are critical to reproduction of experimental results. Repeatable simulation procedures are required for MD simulations. The literature contains many procedures for the calculation of thermodynamic properties. Simulation procedures contain molecular mechanics energy minimization, the use of NVE, NVT and NPT ensembles to equilibrate the system and finally additional NVT and NPT ensembles are run to collect thermodynamic property data. The majority of researchers used Materials Studio Discover tool to run molecular dynamics. Yarovsky and Evans used the PCFF force field whereas COMPASS was the most commonly used.

The literature reports the need to model the Van der Waals and Coulombic forces between atoms for epoxy simulations. Obviously, nonbonded interactions are weak, but there are many of them making them significant in providing physical properties such as density, and moduli. The glass transition temperature is the temperature where the Van der Waals forces “melt” and the material is held together by entangled molecules and the covalent bonds within molecules (Ashby & Jones, 2005). Both the COMPASS and CFF91 force fields provide for calculating Van der Waals and Coulombic forces (Sun, 1998).

Tack and Ford performed Ewald summation used for long-range Coulombic forces, and Van der Waals forces summation for short-range forces, due to their desire to use a common method across MS and LAMMPS. Ewald summation method is of order $N^{(3/2)}$ and the per atom charge information is maintained across all processors when LAMMPS runs on a multi-processor system resulting in slower performance. LAMMPS also provides a particle-particle-particle-mesh (PPPM) algorithm, which is $\text{Nlog}(N)$
where N is the number of atoms. Also, the atom charge information is maintained on the processor responsible for the atoms assigned to it making the calculation of long-range Coulombic forces and Van der Waals forces faster than the Ewald algorithm. A Nonbonded cutoff distance of 9.5Å was used by several researchers especially with Materials Studio and LAMMPS (Tack & Ford, 2008). A large cutoff results in more atoms in a region for computing nonbonded interactions. It is important to note, regardless of using Ewald or PPPM LAMMPS expects the overall system charge to be zero. LAMMPS reports non-neutral systems at the beginning of a simulation.

Molecular Mechanics (MM) energy minimizations are used by all researchers. Wu and Xu used MM energy minimization in Materials Studio. Their MM was configured to terminate after 10,000 steps or less (Wu & Xu, 2006). Several other researchers mention limiting MM to 10,000 steps. The details of MM are rarely mentioned in the literature reviewed. LAMMPS offers a FIRE MM method, though it is slow to converge, has taken 1,000,000 or more steps to converge finding much lower potential energies compared to MM limited to 10,000 steps.

Quenching refers to taking potential energy out of the system. Quenching works very much like MM energy minimization by reducing the system potential energy. Knox used velocity rescaling after MM energy minimization to quench the system. Atoms that were too close together, very high repulsive force situation or “bad contacts” were removed through quenching with small time steps. By scaling the time step logarithmically (e.g. 0.001fs, 0.01fs, 0.1fs and 1.0fs) for 10,000 steps each prevents atoms from getting lost moving from processor to processor (Knox, 2011).
Simulated annealing possibly helps to get a system to move to a global energy minimum configuration. Though not directly identified as simulated annealing, several researchers did employ it. Yarovsky and Evans were interested in imitating the curing regime of the system and used an NVT ensemble at 600K for 100ps followed by a NVT at 300K for 200ps (Yarovsky & Evans, 2002). Their simulation durations are short compared to other published procedures. Knox described a method of cyclic annealing; running NPT ensemble at 1atm and a cold temperature for 50ps, ramp the temperature from a cold temperature to a hot temperature for 50ps, run hot for 200ps and then ramp from a hot temperature to a cold temperature for 50ps followed by a MM energy minimization. The annealing was repeated five times (Knox, 2011). Wu and Xu prepared their system with the NVT ensemble at 600K for 100ps followed by NPT ensemble at 600K and 0.1MPa (1bar) for 500ps (Fan & Yuen, 2007).

System equilibrium is critical to collecting property data whether strain, stress, or thermodynamic data. Yarovsky and Evans replicated the shrinkage of the epoxy system by using the NPT ensemble at 600K and 1atm for 100ps. This was followed by NPT at 300K for 1ns for property collection (Yarovsky & Evans, 2002). Knox used a NPT 2ns equilibration at the temperature of choice and 1atm prior to collecting property data. Tack and Ford simulated the crosslinked DGEBF/DETDA system to calculate thermodynamic properties (Tack & Ford, 2008). Tack and Ford equilibrated samples by running NVT for a minimum of 20 ps to set the temperature at 298K. Next the NPT ensemble was run for 20ps to set the pressure at 1 atm. Each MD cell was subjected to dynamics using the NPT
ensemble for a time of 100-500 ps to reach equilibrium or steady state (Tack & Ford, 2008).

The MD cells were then subjected to dynamics for measurement from a time of 100-1000 ps (Tack & Ford, 2008). Wu and Xu equilibrated their system using the NVT ensemble for 2ns at 600K (Wu & Xu, 2006). After running a simulation for 500ps they used the last 300ps to calculate properties implying the need for the transient behavior of the system to be overcome (Wu & Xu, 2006). Fan and Yuen used the NPT ensemble at 298K and 398K for 200ps respectively to prepare their systems both at 0.1MPa. They simulated curing from 498K to room temperature, but reported a cooling rate of 10K per 200ps while cooling took 4ns (Varshney, Patnaik, Roy, & Farmer, 2008).

2.9 Commercial Software Features

A review of Materials Studio (MS) literature on the accelrys website, in the papers citing its use and in discussions with company representatives clearly defined the minimum requirements for MD software. Materials Studio provides a CAD capability for drawing molecules, the Amorphous Cell tool allows the user to create an MD cell containing the molecules the user specifies and MS contains its own MD simulation engine, Discover, similar to LAMMPS for performing MD simulations. Materials Studio does automatic atom typing for the user. The most significant problem in atom typing is detecting ring structures in the molecule (Schmidt & Fleischhauer, 1978). Amorphous Cell does not directly create crosslinked systems. Researchers either wrote software outside of MS to do crosslinking or wrote scripts that executed inside MS. Materials Studio provide the license needed to use the COMPASS force field and Discover MD engine.
The optimal force field contains enough atom types to represent the unique chemical environments that an atom lives. Ideally, an aromatic carbon atom type should work in Benzene, Naphthalene, Indole, or Thiophene. Force fields usually have a generic atom type for every element in the force field. COMPASS and CFF91 both contain generic carbon. As an example, if a force field did not have a specific atom type for a primary carbon for an alcohol, a generic carbon atom type could be used in its place.

2.10 Open Source Software Features

LAMMPS was suggested as an MD simulation tool due to its robust collection of capabilities. The majority of the research on cross-linked epoxy systems reviewed for this work cites Materials Studio as the software to use for simulating crosslinked epoxy systems and calculating thermodynamic properties. Apparently, there is currently no commercial or open source software available to build crosslinked systems. LAMMPS has integrated support for mean squared displacement (MSD) and radial distribution functions (RDF) to study the movement of small atoms in an epoxy matrix. LAMMPS contains a deformation feature that allows the researcher to apply normal and shear strains to a system in order to calculate stresses.

NanoEngineer-1 (NE-1) was developed by Nanorex, Inc. and by 2009 the company was out of business. Nanorex employed K. Eric Drexler of nanotechnology fame. NanoEngineer-1 was developed for bio-molecule research and the development of nanomachines. The software did not understand alternative force fields beyond NanoDynamics-1 its associated atom types and the GROMACS molecular dynamics package. NanoEngineer-1 lacked the MD cell concept. NanoEngineer-1 provided a very functional CAD capability necessary for this work. NanoEngineer-1 understands a
variety of file formats. NanoEngineer-1 primarily works with the molecular machine part file (MMP) file format. It can create protein data bank (PDB) files through its export function.

SAW, self-avoiding walk, software, I wrote from scratch starting in 2009. SAW is a self-avoiding walk software that when completed will allow a researcher to create amorphous, semi-crystalline and crystalline crosslinked epoxy structures. Work was stopped on the SAW software after the PACKMOL software was discovered at the LAMMPS workshop in August 2011. Appendix C details the features and progress on the SAW software.

PACKMOL was discussed by users at the August 2011 LAMMPS Workshop sponsored by SANDIA National Labs. Several researchers reported using the software to create initial MD cells for simulation. PACKMOL allows the user to specify any group of molecules, any number of molecules and the size of the MD cell to be populated—limited by hardware resource. Also, the user may specify a pseudo random seed value for the population of the MD cell thus allowing the recreation of the unique MD cell at any time. Changing the random seed may force the user to change the MD cell dimensions. PACKMOL modifications were prototyped. PACKMOL was modified to understand the MMP file format.

Early in the literature review on LAMMPS and MD in general the MSI2LMP tool was cited for converting MS CAR/MDF files into a LGIF. The LGIF has a file extension of lammmps05 denoting the file is generated for LAMMPS input and the year, 2005, denoting the year of the file format. The file formats of CAR and MDF are implicitly
defined in the MSI2LMP open source software and the formats are owned by accelrys, MSI2LMP provides two important capabilities for this work, the first its ability to convert CAR/MDF files to a LGIF and with modifications it reads MMP files generated by either NE-1 or PACKMOL and generates LGIFs.

The tool OpenBabel offers open source ring perception software (Group, 2012) (Open Babel: The Open Source Chemistry Toolbox, 2012). The ring perception software could be integrated into NE-1, perhaps in late 2012.

2.11 Thesis Statement
It is postulated that open source software programs can be integrated through judicious modifications and creation of new source code to create molecule models, MD cells and initial LAMMPS geometry input files. These software changes will work together providing an easy and intuitive process for researchers to create simulation systems. Creation of multiple MD cells for research simulations becomes quicker and provides needed standardization to the simulation process.
3. Open Source Software Modifications

3.1 Introduction
The process work flow for the software user (e.g. researcher) drives the software modifications made to NanoEngineer-1 (NE-1), PACKMOL, and MSI2LMP. See process diagram in Figure 1.1 for the mapping to the processes below. The following computer and user processes were defined:

1. Draw molecule geometry using NE-1 (Process N0)
   a. Select a force field
   b. Manually assign atom types
   c. Perform molecular mechanics energy minimization on the molecule
   d. Save the MMP file to a desired folder

2. Create an MD cell Execute PACKMOL (Process P0)
   a. Specify the following information into the PACKMOL input file
      i. Define MD cell size in Angstroms
      ii. Path and filename of each molecule to be built in the MD cell
      iii. Number of each molecule to be built in the MD cell
      iv. A pseudo random seed to generate a unique MD cell configuration
   b. Execute the PACKMOL program (command line program)
      i. Check that the program completed correctly
ii. Check for the output MMP file

3. Generate a LAMMPS geometry input file (LGIF) and Execute MSI2LMP (Process M0)
   a. Specify the following information to the MSI2 LMP program on the command line
      i. Input MMP file generated by PACKMOL
      ii. Force field class
      iii. Force field filename
   b. Execute the MSI2LMP program (command line program)
      i. Check that the program completed correctly
      ii. Maintain force field parameters (Process F0)
      iii. Check for the output lammps05 file

4. Execute LAMMPS (Process L0)
   a. Create a LAMMPS command input file with the following information (Process PH0)
      i. LAMMPS input geometry file
      ii. Molecular mechanics energy minimization
      iii. Quenching
      iv. Simulated annealing
      v. Ensembles to achieve your scientific curiosity
   b. Execute LAMMPS program (command line program)

5. Evaluate Simulation Results (Process E0)
   a. Compare simulation results with published experimental results
b. Compare simulation results with published simulation results

c. Modify simulation parameters
   
i. MD cell configuration
   
ii. Molecule geometry
   
iii. Force field parameters
   
iv. Ensemble parameters
   
v. LAMMPS engine parameters

3.2 NanoEngineer-1 CAD Software Modifications

NanoEngineer-1 (NE-1) consists of nearly 250,000 lines of Python code and the
software modifications are detailed here. It took significant time to learn the Python
language that NE-1 was written in and the inner workings of the NE-1 software.
NanaoEngineer-1 is dependent on Qt for its graphical user (GUI) and on OpenGL to
display the atoms, bonds and molecules. NanoEngineer-1 is computer-aided design
(CAD) software, which allows the user to draw molecules. See Appendix E for detailed
Screen dumps of the NE-1 software. Figure E.1 shows the NE-1 GUI upon starting the
application. The “Chunks” button, midway on the third tool bar, allows the user to place
atoms on the canvas and connect said atoms to form molecules. The user may draw any
number of molecules in an unbounded three dimensional space. NanoEngineer-1 has an
integrated force field, NanoDynamics-1 (ND-1) for the purpose of molecular mechanics
energy minimization. The ND-1 force field is used to “clean up” a CAD drawing of a
molecule. This works for small molecules, but researchers would be better served by a
true clean up feature. ND-1 is sophisticated enough to set bond lengths, angles and
dihedrals at equilibrium values for a molecule in a vacuum. NanoEngineer-1 required software changes to incorporate manual atom typing for an atomistic force field (e.g. CFF91 and COMPASS). Coding for a rule-based atom typing logic into NE-1 was beyond the scope of this work, manual atom typing was implemented by adding dialogs to the current NE-1 software. The user selects an element and an atom type prior to placing the atom on the drawing canvas. Atoms are easily deleted and replaced with another atom with a different type if required.

Clicking on the chunk button displays the force field chooser including its associated atom type chooser, geometry chooser and hotspot chooser boxes on the left-hand side of the application. The geometry and hotspot chooser boxes support the self-avoiding walk software, which is part of this work and is detailed in Appendix C. Figure E.2 shows the chooser boxes on the left hand side of the application. Starting from a blank canvas the user can select a force field. The atom types for each force field are publically published and are displayed in NE-1 to show how force fields can be integrated into the software. Figure E.3 shows the force field drop down, which displays and force fields available for atom typing.

The atom chooser allows the user to select an element (e.g. carbon, oxygen, etc.) and the force field chooser allows the user to pick a specific type of carbon atom based on its chemical environment. The CFF91 force field is the default force field and default atom type is the highlighted “cp” atom type for an aromatic carbon as seen in Figure E.4. If the COMPASS force field had been selected the “c3a” atom type would represent an
aromatic carbon. CFF91 offers 23 carbon, 10 hydrogen, 24 nitrogen, and 13 oxygen atoms to use based on chemical environment.

NanoEngineer-1 opens, modifies and saves molecular machine part (MMP) files. The NE-1 software was modified to incorporate the numeric atom type per atom stored in the file and the force field used in the molecule stored in the file. The MMP file contains chunks. A chunk is a collection of one or more atoms. One problem with the MMP file is combining chunks into a single chunk representing one molecule. NanoEngineer-1 does provide a button to combine chunks. NanoEngineer-1 supports a clipboard feature. Atoms cut are placed on the clipboard and ultimately live inside the MMP file when saved. This presents problems for the modified PACKMOL software due to the MMP file parser being written in FORTRAN. The Clipboard contents must be deleted by the researcher. Future work on PACKMOL will correct this problem.

Figure E.5 shows a tooltip bubble window over the highlighted atom to inform the user that the atom is an aromatic carbon and it has a numeric atom type of 600. Atom types are represented internally as numbers to avoid parsing problems in the future. Atom types are assigned by chemists and they can and will assign a special character in the future that would break the atom type parsing if the parser had been implemented using atom types such as “cp” or “c3a”. COMPASS uses “c3a” to represent an aromatic carbon. The atom types are captured in the molecular machine part file (MMP). MMP is the native geometry file for NE-1. It was modified to extend the grammar of the “info atom” statement to include the force field ID (e.g. “info atom forcefieldid = 600”) capturing the numeric atom type for every atom in the MMP file. Atom types are read in from an MMP
file and assigned to the correct atom when the user wants to display or modify the molecule.

Figure E.6 shows the DGEBA and IPD molecules as drawn by the user in NanoEngineer-1 in close up. The software allows the user to zoom into details of any molecular model. Figure E.7 shows the modified Preferences dialog. The information displayed in the tooltip bubble window is selected by the user here. Specifically, Atom Type and Numeric Atom Type options were added thus allowing the user the option of displaying the information in the tooltip bubble window displayed in Figure E.6.

3.3 PACKMOL Software Modifications

PACKMOL packs molecules into a user defined 3D space by optimizing the pairwise distances between atoms so as to avoid large Van der Waals repulsive forces (Martinez, Andrade, Birgin, & Martinez, 2009). PACKMOL enforces that a user defined distance constraint between pairs of atoms is not violated. PACKMOL is written in FORTRAN. PACKMOL was modified to read the NE-1 MMP file format, including the contained atom types, and generate a final system of molecules that meet the pairwise distance constraint and guarantees all molecules fit inside the user defined MD cell. PACKMOL works with millions of atoms, but it does require equally vast hardware to accomplish such work. A pseudo-random seed value can be used to generate multiple MD cells for simulation. PACKMOL does some of the work of Materials Studio’s Amorphous Cell software. According to the literature (Tack Ford paper) Amorphous Cell attempts to build isotropic MD cells, this is an advantage for the researcher and would be a great addition to the PACKMOL or SAW software.
3.4 MSI2LMP Software Modifications

MSI2LMP is provided in the LAMMPS software distribution. Its purpose is to create a LAMMPS input file from the atom geometry and atom type information defining an MD cell. MSI2LMP is unsupported software. The commercial product, Materials Studio, creates two geometry files with file extensions CAR and MDF. The CAR file contains atom positions, atom partial changes and atom types and the MDF file contains atom bonding data. MSI2LMP was written to translate an MD cell created with Amorphous Cell into a LAMMPS geometry input file. MSI2LMP understands atom types and atom partial charges. Materials Studio automatically types atoms by means of a rule-base software and calculates atom partial charges using the bond increment method, an integral part of the COMPASS and CFF91 force fields. The bond increment values from Equation 2.3 are part of the force field file (FRC). MSI2LMP reads in the atom partial charges from the CAR/MDF file pair and moves them to the LGIF. The modified MSI2LMP program uses the numerical atom types contained in the new MMP file and the bond increment data in the FRC file populate the LGIF with the correct force field parameters and to calculate atom partial charges as per the Equation 2.3 (Sun, 1998).

Additional modifications to MSI2LMP make simulations easier for the researcher. MSI2LMP now represents atom types numerically when processing the new MMP files. Force field files were converted to contain numeric atom types to support this feature and the force field files require maintenance by the researcher if they change. The software was further modified to recognize the force field being used since out-of-plane parameter generation depends on detecting aromatic atoms. For CFF91 the atom type is “cp” and for COMPASS it is “c3a.” Obviously, there are other elements in addition to
carbon that can be aromatic but so far none have been defined in the published force fields. MSI2LMP currently works with. LAMMPS has the capability to track atoms by molecule. This feature is used by LAMMPS in running dynamics on a specific group of atoms. MSI2LMP was modified to maintain the atom molecule association implicit in the new MMP file format and transfer the relationship to the LGIF. This feature will help in the calculation of the mean-squared displacement (MSD) for a specific molecule, such as water, in an epoxy matrix.

MSI2LMP does not use the second equivalence table discussed earlier. It was easier for this work to use the second equivalence table to manually lookup bond increment equivalence values and add the atom pairs directly to the force field file. Modifying the MSI2LMP software to support partial charge calculations in this way is straightforward.

3.5 Software Validation

NanoEngineer-1 validation consisted of adding features to the software design and visually inspecting the new final MMP file format for the expected modifications. The following modifications were added to the MMP file format:

Info atom – The “info atom” token was enhanced to include the token “forcefieldid =” for every atom specified in the MMP file. Therefore, atom number 25 has a force field ID of 600 which is an aromatic carbon. A three-tuple is used to define its position in space and its atomic number identifies the element.

Example: \textbf{atom 25 (6) (-500, 634, 386) def}

\textit{info atom forcefieldid = 600}
Info atom – The “info atom” token was enhanced to include the token, tri =, for every atom that is a member of a triangle in a molecule. Atom 25 now belongs to triangles 1, 8 and 3. See Appendix C for details.

Example: \texttt{atom 25 (6) (-500, 634, 386) def info atom forcefieldid = 600 info atom tri = (1, 8, 3)}

The “info atom” token was enhanced to include the token, “hotspot =”, for every atom that is a member of a hot spot in a molecule. The hydrogen atom below is a female hot spot. Two atoms in a molecule form a hot spot and one is male and the other female. Male and female hotspots are allowed to connect via a covalent bond. Since hotspots are associated with the SAW software, please refer to Appendix C for details.

Example: \texttt{atom 37 (1) (10000, -4567, 2067) def info atom forcefieldid = 108 info atom hotspot = (1:f) or info atom hotspot = (1:m)}

mmptriangles – token was added to enumerate triangles in an new MMP file format. This line in the MMP file tells NE-1 there are 5 triangles in molecule. See Appendix C for details.

Example: \texttt{mmptriangles = (1,2,3,4,5)}

mmphotspots – token was added to enumerate hot spots in the new MMP file format. This line in the MMP file tells NE-1 there is 1 hot spot in the MMP file. See Appendix C for details.

Example: \texttt{mmphotspots = (1)}
Mmpforcefield – token was added to identify the force field selected by the user for the creation of molecules.

Example: \texttt{mmpforcefield = CFF91}

Validation testing consisted of creating a DGEBA or DGEBA-like molecule and assigning atom types by force field, assigning atoms to triangles and to hot spots and assigning a force field to a molecule through the NE-1 GUI and performing a visual inspection of the MMP file created to insure that the data was correct.

PACKMOL testing required visual inspection of the MMP file created by PACKMOL as well as reviewing the output of MSI2LMP. PACKMOL reads in one or more MMP files as templates for building molecules at many attitudes within the confines of the MD cell and generates a new MMP file with many more molecules present in the MMP file. Specifically, every atom required a correct type and the number of molecules and atoms had to match the number requested for creation. Visual inspection showed, after many software bug fixes, that PACKMOL did generate a correct MMP file. Additional work is needed to have PACKMOL pass on the force field name from the molecule MMP files to the final PACKMOL MMP file. The user can edit this manually.

A critical test for the MSI2LMP software was to validate its results against a standard. The ultimate software validation process is to create a significantly complex molecule and generate a correct output file. The software validation test target is a correct LGIF. Figure E.1 shows the square oligomer used to create an MD cell for the MD simulations carried out in this work. The Oligomer is composed of 4 IPD and 9 DGEBA
molecules consisting of 569 atoms. The MD cell contains 5 oligomers consisting of 2845 atoms. The majority of researchers in the literature use Materials Studio to create MD cells by using the MS CAD features and the Amorphous Cell module in MS. Materials Studio can create the 569 oligomer via CAD or it can import the oligomer in the protein data bank (PDB) file format and export the same file information as a CAR/MDF file pair. The CAR/MDF files for the oligomer were created to generate the LGIF using MSI2LMP. The modified MSI2LMP software was used to create a LGIF from the oligomer MMP file. Figure 3.1 and Figure 3.2 displays the side-by-side comparisons between CAR/MDF and MMP derived LGIF files. The two files are available in Appendix A for review. The two files in theory should be identical. The two files should define an MD cell with identical dimensions. The number of atoms, bonds, angles, dihedrals, impropers and types should be identical. The atom positions and partial charges should be identical. In reality there will be slight differences in the MD cell dimensions and atom positions due to how floating point values are represented between the CAR/ MDF file pair and MMP file formats. Nearly identical comparisons validate the MD cell created using two paths through the MSI2LMP software. The CAR/MDF file pair was created by a colleague.
Figure 3.1 - Beyond Compare file comparison of CAR and MMP file for LAMMPS geometry input file.
Figure 3.2 - Beyond Compare file comparison of CAR and MMP file for LAMMPS geometry input file.
4. **Molecular Dynamics Simulations**

4.1 Simulation setup

The simulation setup is based on both the science of the simulation as well as the needs for successful execution of the LAMMPS software. Periodic boundary conditions (PBC) were used for all three dimensions for all system simulation results reported in this work. Essentially, atoms in a PBC environment move through a wall in the MD cell and are relocated on the other side of the MD cell – similar to how a PAC-MAN moved in the video arcade game. This is done to maintain a constant number of atoms for the ensembles used. The CFF91 force field was used. The PPPM k-space solver was used for the initial simulations. The k-space solver is used by LAMMPS to compute long-range Coulombic interactions. The precision value of 1.0e-6 was selected. The CFF91 Van der Waals forces have a cutoff of 9.5Å while the Coulombic forces, based on calculated partial charges have a 10Å cutoff. A 1fs time-step was used and is considered a standard based on the vibration frequency of the carbon-hydrogen covalent bond. LAMMPS was configured to use a skin distance of 2Å with a “bin” style for storing atom pairs in memory. A larger skin value results in neighbor lists being rebuilt less often. Given the nature of the system being simulated the neighbor lists were rebuilt every time step. Storage was allocated for 10.e6 atom pairs per neighbor page and 1.0e5 neighbors per atom. Initial simulations were conducted using the Ewald algorithm and a 4Å bin, which
was replaced with the PPPM algorithm using a 2Å skin with the added benefit to report bad dynamics, where bad dynamics is defined as atoms getting lost when moving from processor to processor. Short range Coulombic interactions are calculated in real space. Table 4.1 shows the mass calculation for the 75% crosslinked oligomer in Figure 4.1. Similar calculations were done for the other oligomers used.

Table 4.1 - System two: mass calculations for the 5 Oligomer Ring MD cell.

<table>
<thead>
<tr>
<th>Atom</th>
<th>No. of Atoms</th>
<th>Segment Molecule</th>
<th>No. of Segments</th>
<th>No. of Atoms in Segment</th>
<th>Atomic Wt.</th>
<th>Mass (gm/mol)</th>
<th>Total Mass (gm/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>25</td>
<td>DGEBA (n=0)</td>
<td>4</td>
<td>100</td>
<td>1.00794</td>
<td>25.1985</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>21</td>
<td></td>
<td>84</td>
<td></td>
<td>12.0107</td>
<td>252.2247</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>4</td>
<td></td>
<td>16</td>
<td></td>
<td>15.9994</td>
<td>63.9976</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td>14.0067</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Sub-Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>341.428</td>
<td>1365.6832</td>
</tr>
<tr>
<td>H</td>
<td>19</td>
<td>IPD</td>
<td>4</td>
<td>76</td>
<td>1.00794</td>
<td>19.15086</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>10</td>
<td></td>
<td>40</td>
<td></td>
<td>12.0107</td>
<td>120.107</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td>15.9994</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>2</td>
<td></td>
<td>8</td>
<td></td>
<td>14.0067</td>
<td>28.0134</td>
<td></td>
</tr>
<tr>
<td>Sub-Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>167.271260</td>
<td>669.08504</td>
</tr>
<tr>
<td>H</td>
<td>26</td>
<td>DGEBA (n=0)</td>
<td>3</td>
<td>78</td>
<td>1.00794</td>
<td>26.206440</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>21</td>
<td></td>
<td>63</td>
<td></td>
<td>12.0107</td>
<td>252.2247</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>4</td>
<td></td>
<td>12</td>
<td></td>
<td>15.9994</td>
<td>63.9976</td>
<td></td>
</tr>
<tr>
<td>N</td>
<td>0</td>
<td></td>
<td>0</td>
<td></td>
<td>14.0067</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Sub-Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>342.428740</td>
<td>1027.28622</td>
</tr>
</tbody>
</table>
Table 4.2 - system 2 mass calculations for the 5 oligomer ring MD cell.

<table>
<thead>
<tr>
<th>Atom</th>
<th>No. of Atoms</th>
<th>Segment Molecule</th>
<th>No. of Segments</th>
<th>No. of Atoms in Segment</th>
<th>Atomic Wt.</th>
<th>Mass (gm/mol)</th>
<th>Total Mass (gm/mol)</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>46</td>
<td>DGEBA (n=1)</td>
<td>1</td>
<td>46</td>
<td>1.00794</td>
<td>46.365240</td>
<td></td>
</tr>
<tr>
<td>Carbon</td>
<td>39</td>
<td></td>
<td></td>
<td></td>
<td>12.0107</td>
<td>468.4173</td>
<td></td>
</tr>
<tr>
<td>Oxygen</td>
<td>7</td>
<td></td>
<td></td>
<td>7</td>
<td>15.9994</td>
<td>111.9958</td>
<td></td>
</tr>
<tr>
<td>Nitrogen</td>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
<td>14.0067</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>Sub-Total</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>626.77834</td>
<td>626.77834</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td></td>
<td></td>
<td>569</td>
<td></td>
<td>3688.8328</td>
<td></td>
</tr>
</tbody>
</table>

Total MD Cell Mass (gm) 3.0627e-20

4.2 Molecular Dynamics Cell Configuration

Several systems, each requiring an MD cell, were constructed for simulation. Results for four systems are reported in this work. The first is a no crosslinked system composed of 45 DGEBA monomers and twenty IPD molecules. All DGEBA monomers and IPD molecules are free to move subject to the dynamics of the system. System one is essentially a liquid and therefore no MD cell geometry is presented. System one has a density of 1.0 gm/cc at 298K and 1atm.

System two is composed of five oligomers where each oligomer is composed of seven DGEBA (n=0) monomers and one DGEBA (n=1) dimer and four IPD molecules in a flat square geometry shown in Figure 4.1. The oligomer is 75% crosslinked within itself, intra-crosslinked, thus making system one 75% crosslinked. This oligomer topology was used by Tack and Ford with their DGEBA/DETDA system (Tack & Ford, 2008). The Tack and Ford MD cell consisted of five oligomers, which were determined
to be significant and not overtaxing of the computer resources. PACKMOL calculated a 55Å cell length provided the smallest MD cell for the five oligomers for system two. A user-defined tolerance for repulsive forces was used by PACKMOL to insure that no atoms in the MD cell were closer than 2Å. Note: the initial MD cell in Figure 4.2 shows the large initial MD, which was reduced to the equilibrated final MD cell, 30.5Å per side, at 298K and 1atm. The five oligomers are squeezed down into the smaller volume. The final LAMMPS geometry input file was generated using MSI2LMP to create the lammps05 file. The LAMMPS geometry input files are shown in Appendix A. System two contains 2845 atoms and is shown in Figure 4.2.

System three is composed of five oligomers where each oligomer is composed of nine DGEBA (n=0) monomers and 1 DGEBA (n=1) dimer and four IPD molecules in a flat square geometry shown in Figure 4.3. The oligomer is 87.5% crosslinked within itself, intra-crosslinked, making the system 87.5 crosslinked and contains 666 atoms. PACKMOL was used to create the initial MD cell. The user-defined tolerance for repulsive forces was initially 2 Å, but was later increased to 3Å since the initial simulations were failing due to bad dynamics (AKA “missing atoms”). Adding two DGEBA monomers was required to build the necessary oligomer. Its geometry was later changed to reduce stain in the oligomer. The second configuration did not suffer the “missing atoms” error the initial system did. The System three MD cell was cubic 67Å per side as determined by PACKMOL. System three contains 3330 atoms.

System four was composed of two inter-crosslinked oligomers; see Figure 4.4, and one of the oligomers shown in Figure 4.1. The system is based on crosslinking two
oligomers at three positions on each oligomer thereby creating an inter-crosslinked oligomer composed of 1138 atoms. PACKMOL was used to create the initial MD cell consisting of two big oligomers and one small oligomer. The system four initial MD cell was cubic, 83.5 Å per side as determined by PACKMOL. System four required additional MM energy minimization due to manual construction of the big oligomer using NanoEngineer-1, which was required to crosslink the two oligomers and introduced additional strain in the geometry.

4.3 Molecular Mechanics Energy Minimization
All four system’s energy was minimized using the Polak-Ribiere conjugate gradient method provided in LAMMPS in conjunction with the “box/relax” feature. The box/relax feature allows the minimization to occur at a user specified pressure, which was 1atm. All systems simulated experienced a reduced volume during the minimization due to the application of the external pressure. The FIRE method was also used with system four. Several rounds of Polak-Ribiere and FIRE were used to lower the potential energy. Visual inspection of the MD cell showed that significant strain had been removed from the oligomers employed. The VMD tool uses average bond lengths to calculate bond placements when rendering the system geometry. System four started with many “unbonded atoms” due to the distance between atoms and after the significant minimization process the bonds were displayed since the atoms were now close enough to render the bonds.

4.4 System Quench
All systems were quenched by running an NVT ensemble at 298K with time steps of 0.001fs, 0.01fs, 0.1fs, 0.2fs, and 0.4fs for 10ps at each time step for several simulations
while velocity rescaling was used for several simulations. The short time steps allowed rapid atom movements to occur without atoms getting lost moving from one processor to another.

Figure 4.1 - 75% intra-oligomer crosslinked 569 atoms.
Figure 4.2 - System 2: Initial MD cell configuration 2845 atoms.

Figure 4.3 - 87.5% intra-oligomer crosslinked 666 atoms.
Figure 4.4 - 93.75% Inter-oligomer crosslinking 1138 atoms.

Figure 4.5 - System 3 initial MD cell 2845 atoms.
4.5 Simulated Annealing

Five rounds of simulated annealing were conducted to condition all systems to calculate densities at several temperatures. Table 4.3 shows the simulated annealing details for each of five rounds. One important issue not addressed is how fast can the MD cell be heated or cooled in simulated annealing. Fan and Yuen did some work looking at different cooling rates (Fan & Yuen, 2007).

Table 4.3 - Single Round of Simulated Annealing.

<table>
<thead>
<tr>
<th>Step</th>
<th>Start Temp (K)</th>
<th>End Temperature (K)</th>
<th>Duration (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>298</td>
<td>298</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>298</td>
<td>600</td>
<td>50</td>
</tr>
<tr>
<td>3</td>
<td>600</td>
<td>600</td>
<td>200</td>
</tr>
<tr>
<td>4</td>
<td>600</td>
<td>298</td>
<td>50</td>
</tr>
</tbody>
</table>

4.6 Equilibrium and Density Data Collection

The equilibrium of all four systems after simulated annealing was accomplished by running a series of NPT ensembles. The NPT ensemble was used to equilibrate the
systems and to ramp the temperature of the systems to the next sampling temperature. Table 4.4 shows the details.

Table 4.4 - LAMMPS ensembles used to collect density data.

<table>
<thead>
<tr>
<th>Step</th>
<th>Start Temperature (K)</th>
<th>End Temperature (K)</th>
<th>Duration (ps)</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>298.0</td>
<td>298.0</td>
<td>10</td>
<td>Collect MD cell dimensions and average temperature</td>
</tr>
<tr>
<td>2</td>
<td>298.0</td>
<td>328.2</td>
<td>100</td>
<td>Ramp temperature</td>
</tr>
<tr>
<td>3</td>
<td>328.2</td>
<td>328.2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>328.2</td>
<td>358.4</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>358.4</td>
<td>358.4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>358.4</td>
<td>388.8</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>388.8</td>
<td>388.8</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>388.8</td>
<td>419.0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>419.0</td>
<td>419.0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>419.0</td>
<td>449.2</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>449.2</td>
<td>449.2</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>449.2</td>
<td>479.4</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>479.4</td>
<td>479.4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>479.4</td>
<td>509.6</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>509.6</td>
<td>509.6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>509.6</td>
<td>539.6</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>539.6</td>
<td>539.6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>539.6</td>
<td>570.0</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>570.0</td>
<td>570.0</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>570.0</td>
<td>600.2</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>600.2</td>
<td>600.2</td>
<td>10</td>
<td></td>
</tr>
</tbody>
</table>

4.7 Equilibrium and Bulk Modulus Data Collection

The bulk modulus was calculated by linearly increasing (i.e. “ramping”) the pressure of system two from 1atm to 5001atm at 298K using the NPT ensemble followed
by another NPT ensemble at 298K and 5001atm. Table 4.5 defines the steps used. The pressurization duration could be analyzed in the same manner as heating and cooling of the MD cell to determine which procedure reflects real physical processes.

Table 4.5 - Steps to collect bulk modulus data.

<table>
<thead>
<tr>
<th>Step</th>
<th>Start Pressure (atm)</th>
<th>End Pressure (atm)</th>
<th>Duration (ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1.0</td>
<td>1.0</td>
<td>100</td>
</tr>
<tr>
<td>2</td>
<td>5001.0</td>
<td>5001.0</td>
<td>100</td>
</tr>
</tbody>
</table>

4.8 Simulation Density Results and Glass Transition Temperature

Table 4.6 shows the calculated minimum and maximum density values based on minimum and maximum MD cell dimensions generated during the simulation of system two. The distribution of density values can be used to identify equilibrium of the system. The systems simulated have a transient and a steady state regime. The literature did not specifically address this and averages presented in this work are for the entire time of the simulation. If a time constant could be derived for the systems then data after the first time constant could be averaged and only the minimum and maximum values after the transients are banished could be used to test for equilibrium. The density data in Table 4.6 was generated by running at the specified temperature for 100ps and collecting a data point every 0.1ps resulting in 1000 data points for every calculated density. The temperatures in Table 4.6 are calculated from 1000 temperatures sampled during the simulation for a specific temperature set point. The same procedure was applied to all systems.
Table 4.6 - System 2 range of temperature and density during variation.

<table>
<thead>
<tr>
<th>Average Temperature (K)</th>
<th>Maximum Temperature (K)</th>
<th>Minimum Temperature (K)</th>
<th>Maximum Density (gm/cc)</th>
<th>Minimum Density (gm/cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>298.26</td>
<td>307.81</td>
<td>288.82</td>
<td>1.060</td>
<td>1.025</td>
</tr>
<tr>
<td>328.05</td>
<td>340.21</td>
<td>317.27</td>
<td>1.067</td>
<td>1.035</td>
</tr>
<tr>
<td>358.47</td>
<td>368.88</td>
<td>346.28</td>
<td>1.062</td>
<td>1.032</td>
</tr>
<tr>
<td>388.72</td>
<td>406.61</td>
<td>374.03</td>
<td>1.061</td>
<td>1.024</td>
</tr>
<tr>
<td>418.93</td>
<td>432.50</td>
<td>402.7</td>
<td>1.052</td>
<td>1.007</td>
</tr>
<tr>
<td>449.12</td>
<td>469.56</td>
<td>433.77</td>
<td>1.035</td>
<td>0.989</td>
</tr>
<tr>
<td>479.44</td>
<td>495.73</td>
<td>463.58</td>
<td>1.032</td>
<td>0.990</td>
</tr>
<tr>
<td>509.72</td>
<td>527.25</td>
<td>491.7</td>
<td>1.02</td>
<td>0.980</td>
</tr>
<tr>
<td>540.02</td>
<td>557.47</td>
<td>520.94</td>
<td>1.01</td>
<td>0.963</td>
</tr>
</tbody>
</table>

Table 4.7 - System two calculated density vs temperature.

<table>
<thead>
<tr>
<th>Temperature (K)</th>
<th>Lx (Å)</th>
<th>Ly (Å)</th>
<th>Lz (Å)</th>
<th>Volume (cc) X 1.0e20</th>
<th>Density (gm/cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>297.55</td>
<td>30.75</td>
<td>30.76</td>
<td>30.77</td>
<td>2.8954</td>
<td>1.056</td>
</tr>
<tr>
<td>328.02</td>
<td>30.74</td>
<td>30.76</td>
<td>30.77</td>
<td>2.9106</td>
<td>1.0510</td>
</tr>
<tr>
<td>358.12</td>
<td>30.78</td>
<td>30.79</td>
<td>30.80</td>
<td>2.9197</td>
<td>1.0458</td>
</tr>
<tr>
<td>388.63</td>
<td>30.83</td>
<td>30.84</td>
<td>30.85</td>
<td>2.9344</td>
<td>1.0427</td>
</tr>
<tr>
<td>418.66</td>
<td>30.79</td>
<td>30.80</td>
<td>30.81</td>
<td>2.9222</td>
<td>1.0470</td>
</tr>
<tr>
<td>449.02</td>
<td>30.93</td>
<td>30.94</td>
<td>30.95</td>
<td>2.9634</td>
<td>1.0325</td>
</tr>
<tr>
<td>479.55</td>
<td>31.07</td>
<td>31.08</td>
<td>31.09</td>
<td>3.0042</td>
<td>1.0185</td>
</tr>
<tr>
<td>509.85</td>
<td>31.39</td>
<td>31.14</td>
<td>30.42</td>
<td>3.023</td>
<td>1.0122</td>
</tr>
<tr>
<td>539.95</td>
<td>31.32</td>
<td>31.34</td>
<td>31.35</td>
<td>3.0783</td>
<td>0.99404</td>
</tr>
<tr>
<td>570.57</td>
<td>31.46</td>
<td>31.47</td>
<td>31.48</td>
<td>3.1164</td>
<td>0.98187</td>
</tr>
<tr>
<td>599.86</td>
<td>31.56</td>
<td>31.57</td>
<td>31.58</td>
<td>3.14645</td>
<td>0.9725</td>
</tr>
</tbody>
</table>
Table 4.8 - System two simulated vs experimental results.

<table>
<thead>
<tr>
<th>System Two Simulated vs Experimental Results at 298 K and 1 atm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simulated Density (gm/cc)</td>
</tr>
<tr>
<td>1.056 ± 0.02</td>
</tr>
</tbody>
</table>

Table 4.9 - System three calculated density vs temperature.

<table>
<thead>
<tr>
<th>System Three 87.5% Intra-oligomer crosslinked System Based on Average Volumes Mass: 3.6296e-20 gm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (K)</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>297.42</td>
</tr>
<tr>
<td>328.07</td>
</tr>
<tr>
<td>358.47</td>
</tr>
<tr>
<td>388.71</td>
</tr>
<tr>
<td>418.99</td>
</tr>
<tr>
<td>449.40</td>
</tr>
<tr>
<td>479.40</td>
</tr>
<tr>
<td>509.56</td>
</tr>
<tr>
<td>539.72</td>
</tr>
<tr>
<td>569.80</td>
</tr>
<tr>
<td>600.18</td>
</tr>
</tbody>
</table>

Table 4.10 - System four calculated density vs temperature

<table>
<thead>
<tr>
<th>System Four 93.75% Intra-oligomer crosslinked System Based on Average Volumes Mass: 3.06e-20 gm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature (K)</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>294.18</td>
</tr>
<tr>
<td>328.20</td>
</tr>
<tr>
<td>358.30</td>
</tr>
<tr>
<td>388.86</td>
</tr>
<tr>
<td>419.02</td>
</tr>
<tr>
<td>448.92</td>
</tr>
<tr>
<td>479.32</td>
</tr>
</tbody>
</table>
Figure 4.7 shows a graph of the average density data plotted, in blue, against the average temperatures from Table 4.7 for system two. The average temperatures and densities were arrived at via the same method described for Table 4.6. The graph in Figure 4.7 shows least squares fit line, in red, of the first five temperature vs. density values while the remaining points were used to generate the least squares fit, in green, of the remaining six temperature vs. density values. The first five points were selected to represent the glassy material prior to Tg while the remaining points represent the viscous material.
Figure 4.7 - System 2 least squares 5 point glassy, 6 point viscous Tg calculation.

The system two graphs, shown in Figures 4.8 and 4.9, are for the six and four point glassy data respectively and show variation in the glass transition temperature. The glass transition temperatures range from a low of 423K to 448K. Tack and Ford reported 417K for the DGEBA/DETDA system, which is a good target for this class of epoxy using the CFF91 force field.
Figure 4.8 - System 2 least squares 6 point glassy, 5 point viscous Tg calculation.
Theory and experimentation show that the glass transition temperature should increase with crosslinking. Simulation results presented for systems one and two clearly show that crosslinking results in higher densities and increases the glass transition temperature. System one, with a simulated density of 1.0 gm/cc has no crosslinks while system two is composed of intra-oligomer crosslinks and has a density of 1.05 gm/cc at 298K and 1atm. A quick investigation of this phenomenon was warranted given the software developed for this work allows for rapid creation of MD cells for simulation.

System three is an intra-oligomer crosslinked system like system two, but it contains two additional DGEBA monomers. Figure 4.3 and Figure 4.5 show the geometry of the oligomer and system geometry for system three respectively. The system three calculated density data is shown in Table 4.9 and the least squares fit for the first five points, red line, for the glassy regime is shown in Figure 4.10.
System four is an inter-oligomer crosslinked system since the crosslinks extend from one oligomer to another. In order to create system four two oligomers used in system 2, Figure 4.1, were crosslinked in three positions in their structures to form a larger oligomer as seen in Figure 4.4 making the large oligomer 93.75% crosslinked. System four contains two large oligomers and a single small oligomer. Figure 4.6 shows the geometry of system four. The calculated temperature and density data for system four are in Table 4.10. The least squares fit for the first three points, red line, for the glassy regime is shown in Figure 4.11 in Figure 4.12 for system four. The density values are lower for system four compared to systems two and three in the first simulation and are considerably higher in the second simulation due to the longer equilibrium process. Thirty MM energy minimizations were used to remove stress from the initial structure.
The density appears to remain constant up to the glass transition temperature and then a rapid reduction of density occurs for the first simulation of system four. The second simulation, Figure 4.12, shows a steady decrease in density up to 479K and then a rapid decrease.

Figure 4.11 - System 4 least squares 3 point glassy, 8 point viscous T_g calculation.
The raw volume data is shown in Table 4.11 and calculated bulk modulus is shown in Table 4.12. The simulated value is higher than the experimental, but compares favorably.

Table 4.11 - Raw volume data for bulk modulus calculation

<table>
<thead>
<tr>
<th>Pressure (atm)</th>
<th>Lx (Å)</th>
<th>Ly (Å)</th>
<th>Lz (Å)</th>
<th>Volume (cc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>30.86</td>
<td>30.87</td>
<td>30.88</td>
<td>2.94e-20</td>
</tr>
<tr>
<td>5001.0</td>
<td>24.76</td>
<td>29.82</td>
<td>35.92</td>
<td>2.65e-20</td>
</tr>
</tbody>
</table>
Table 4.12 - Simulated and experimental bulk modulus.

<table>
<thead>
<tr>
<th>ΔP (MPa)</th>
<th>ΔV (cc)</th>
<th>V₀ (cc)</th>
<th>Simulated Bulk Modulus (GPa)</th>
<th>Experiential Bulk Modulus (GPa)</th>
</tr>
</thead>
<tbody>
<tr>
<td>506.6</td>
<td>2.90e-21</td>
<td>2.94e-20</td>
<td><strong>5.13</strong></td>
<td><strong>5.01</strong></td>
</tr>
</tbody>
</table>
5. Discussion and Conclusions

5.1 Software Testing

The LAMMPS geometry input files (LGIF) shown in Figures 3.1 and 3.2 are nearly identical. The round off of the atom coordinates between Materials Studio and NanoEngineer-1 are negligible and result from NE-1 using truncated integers to represent atom coordinates in the MMP file while the CAR/MDF file pairs use floating point representations. The differences were tracked down to errors in the conversion of the MMP to PDB file, which is a bug in the NE-1 software. Conversion of the MMP file to the PDB format was not perfect and required an unbonded hydrogen atom to be manually reattached to the DGEBA molecule prior to the creation of the CAR/MDF file pair. The results show the same atom, bond, angle and dihedral types as well as force field parameters.

Clearly, the LGIF is created correctly. The LGIF relies on a correctly formed MMP file whether generated by NE-1 or by PACKMOL.

5.2 Molecular Dynamics Testing

5.2.1 Density Results

The low density results for system two can be explained by a combination of factors pertaining to the simulation procedures, MD cell geometry and force field data. It is entirely possible that longer equilibration times could result in higher density values. Many additional simulations were run, but the density values were not significantly
different from the ones presented in chapter four using the same procedures as outlined in chapter four. A single simulation required three days which limited the number of simulations. Ewald simulations took even longer. Equilibration times are low to accommodate the hardware resource available. An initial simulation used Van der Waals and Coulombic cutoffs of 8Å and 10Å respectively with density values slightly above 1 gm/cc and lower and much lower than the 1.05 gm/cc in the reported results. Raising the Van der Waals cutoff to 9.5Å increased densities. The cutoff was not further raised since 9.5Å is reported in the literature.

System two had an initial system cell length of 55Å. By applying 1atm pressure and executing the simulated annealing procedure the volume shrinks to a cube of approximately 30.5Å per side. This shrinking of the crosslinked epoxy stands out as a significant test that the simulations are in fact working correctly. The results reported treated system two as isotropic throughout the simulation procedures. System two was simulated as anisotropic to look for densities closer to experimental values. The MD cell changed from cubic to a rectangular prism, but the density values did not significantly change. The final result is a very nearly cubic system as reported in Table 4.6. Squeezing the system from 1atm to 5001atm and back to 1atm was employed after simulated annealing in hopes of seeing an increase in density values. The system volume returned to its pre-squeezed volume, which shows how well the Van der Waals and Coulombic portions of the simulation were working. The reported density at 298K is about 7% to 8% under the experimental value of 1.131 gm/cc. Figure 4.2 shows the initial MD cell configuration prior to simulations.
The MD cell geometry is critical to the calculation of system properties. Tack and Ford used MS to create their oligomer geometry. Materials Studio automatically typed the atoms and the Amorphous Cell (AC) tool created the MD cell used. As reported, their simulations consisted of approximately 10,000 atoms in variable sized MD cells based on attempting to create an MD cell that was isotropic. Tack and Ford created the MD cell file as a CAR/MDF file pair. The CAR/MDF file pair contained atom partial charges as calculated by MS. Modifications made to MSI2LMP accomplished the same result of a neutral charged system and assignment of partial charges to atoms. Materials Studio’s unique method of creating an MD cell may be responsible for the different results achieved in the Tack and Ford work compared to this work. The MD cell for system two uses all five small oligomer structures and PACKMOL intertwines the oligomers. Upon compression to 1atm the system may contain internal strain, which helps to maintain a larger volume at temperature resulting in low density.

Additional simulations were run for a system with no crosslinks (liquid) and for a system where each oligomer ring was cut to allow the molecule more degrees of freedom. The liquid system achieved a density of 1 gm/cc, while using a linear molar ratio mixing model one would expect a density of the liquid of 1.10 gm/cc. This “macro-level mixing” approach does not account of decreases in total volume upon mixing at the nano-level. The low density could be due to each DGEBA and IPD molecule has a larger volume due to its Van der Waals force field. System two has more covalent bonds bringing the molecules closer together. The severed oligomer system did achieve a density of 1.05
gm/cc at 298K and 1atm and no internal strain in the MD cell due to the oligomer ring structure was uncovered. The two variant systems had nearly the same number of atoms as system two. The goal of using the severed ring and no crosslinked model was to construct two MD cells that were more homogeneous at 298K and 1atm and would result in the same mass being packed into a smaller volume. This did not occur given the simulation procedures employed.

Another area of investigation is that all systems modeled had DGEBA monomers waving like flags from the IPD molecules. All of these DGEBA monomers were modeled as unreacted. The epoxide ring was intact and this would not be the case in the nearly 100% cured epoxy. There are 20 such epoxide rings present in system two and 30 in system three. Follow simulations will be done with the rings open and hydroxyl groups formed. One ancillary and very preliminary simulation with a water molecule in system two showed it to be captured by the oxygen on an epoxide. The MD cells used by Tack and Ford appear to be sub-sampled from the larger system of oligomers and therefore may not have the same chemical environment as the systems used in this work.

Tack and Ford reported consistently low density values for their DGEBF/DETDA systems using CFF91 (Tack & Ford, 2008). Their work does not mention the missing force field parameters reported by the MSI2LMP conversion tool for this work. It is not known if their work included software and/or force field modifications. Additional modifications to MSI2LMP could be done to include the use of the second equivalence table provided in the CFF91 force field. Specifically, more cross-coupling parameters may be populated into the LGIF. All missing force field terms
reported by MSI2LMP for this work were “fixed” by manually applying the two equivalence tables to fill in missing force field data. Cross-term parameters were by-in-large missing.

System three increased the intra-crosslinking from 75% to 87.5% by adding two additional DGEBA monomers to the existing small oligomer. The hope was to achieve the same small volumes as seen for system two but with the higher mass for system three thereby increasing the density of system there and perhaps attempt to show the low density of system two could be due to voids. However, the “flag” DGEBA monomers were considered for the low density, but additional simulations with opened epoxide rings in the form of a methyl and a hydroxyl group showed no change in the density. Replacing DGEBA with DGEBF in one additional simulation resulted in higher densities. The missing Methyl groups most likely contributed to the increase in density. Also, it was expected that the $T_g$ value would increase in value for system three. Figure 4.7 and Figure 4.10 show the density vs temperature for systems two and three respectively. Both show slow drop in density in the glassy regime up to $T_g$. Both systems rely on Van der Waals forces to hold the oligomers together. The results show that crosslinking within the oligomer only raises $T_g$ just so far. System size could also contribute to not seeing crosslinking increasing $T_g$. Bandyopadhyay clearly shows an increase in $T_g$ with increasing crosslinking, but the systems used dynamic crosslinking methodology and over 25,000 atoms (Bandyopadhyay, 2001).

System four was created by crosslinking two small oligomers to form a large oligomer that was 93.75% crosslinked. The oligomer had 1138 atoms twice the number
of the small oligomer. See Figure 4.1 and Figure 4.4 for the small and large oligomer structures. System four was constructed with two large and one small oligomer. The expectation was to see the $T_g$ increase above the 423K seen for system two. The $T_g$ value dropped to 380K as seen in Figure 4.11 and increased to 479K based on simulation two shown in Figure 4.12. Interestingly, the densities below $T_g$ remained constant and appear to defy the increase in temperature that affected systems two and three, while simulation two of system four shows the density gradually changing from 1.17gm/cc to 1.09gm/cc. The additional covalent bonding across oligomers may have given the system better temperature stability from 298K to 380K and this may be the case for the second simulation for both the glassy and viscous regions. The densities are lower for system four simulation-one compared to systems two and three for temperatures below $T_g$. This is due to how system four simulation-one was constructed and equilibrated. Stress was introduced into the large oligomer by manually crosslinking the small oligomers. All systems were treated to the same set of procedures that did not discriminate based upon the needs of the specific system. System four must be better equilibrated through longer MM, quenching and annealing as pointed to by simulation two. Simulation two does point to inter-oligomer crosslinking being required to raise $T_g$.

Additional simulations will be run to lengthen the cooling ramp during annealing. The 50ps cooling time was too short and lengthened to 100ps for simulation two. The data collection procedure was modified to start the system under simulation at 600K and slowly cool the system to 298K. Trapping volume in the structure is a concern
with the rapid cooling. A networked topography oligomer containing 2845 atoms will be simulated. Additional test systems may not have separate oligomers.

5.2.2 Bulk Modulus Results

The bulk modulus for system two was higher than expected. Previous runs using different simulation parameters resulted lower values as low as 4.85 GPa. The bulk modulus calculation was very simple and simulations were not run in order to calculate it for systems three and four.
6. Summary and Future Work

The software created for this work and the results obtained support the thesis and provide the necessary evidence of success. This work initially cited the results of systems one and two. Systems three and four results were conceived created and simulated in one week, the week just before this work was to be distributed for review, where much of that time (5 days) was devoted to computation on the computer. These last two systems and the time taken to generate results stand as evidence that the system of software created does allow for quick creation of MD cells for simulation. Open source software is a valuable tool in the area of molecular dynamics research. LAMMPS, OpenBabel, VMD and other open source software is making a contribution to research. The modifications to NanoEngineer-1, PACKMOL and MSI2LMP could do the same with additional effort to move the software beyond the proof of concept stage. The self-avoiding walk software, SAW, discussed in Appendix C will generate systems with varying degrees of crosslinking and structures that are amorphous to crystalline once the software is complete. Unfortunately, work on SAW was abandoned in favor of completing modifications on PACKMOL. This resulted in molecular dynamics density, glass transition temperature and bulk modulus in line, within 7%, with published experimental results.
Future work will consist of productizing the software used in this work and release as open source software including the SAW software once it is complete. Force fields are the heart of molecular dynamics research. This work has illuminated the need for access to open source force fields. An open source force field on the level of COMPASS is needed in the public domain. Force fields are expensive to produce due to their reliance on skilled chemists to use Ab Initio quantum mechanical software packages to calculate force field terms and cross-coupling terms. Also, they must perform laboratory work to gather data for the nonbonded portions of the force field. There were many commercial product features not incorporated into this work due to time and resources. The following features are candidates for future work:

Ring perception and automatic atom typing – this feature is the heart of automatic atom typing. Once the software knows where the rings are in a molecule it can determine if the ring is aromatic, and apply rule-based software to type the atoms in the molecule.

Automatic identification of triangles – ring perception will allow for automatic collision detection in the SAW software.

Better management of molecular dynamics simulation procedures and LAMMPS control input files. The current approach to simulations it error prone and additional software that can manage processes by concept such as quenching or annealing is easier than having to type in commands over and over.

Shear and elastic modulus calculations

Migration of water in crosslinked epoxy systems.

Aging of crosslinked epoxy systems
Molecular dynamics is an exciting field of study and many researchers can benefit from the work started here.
7. References


8. **Appendix A – Raw and Simulation Data**

All Simulation data and results are on a DVD and available upon request.

9. **Appendix B – Software Algorithms**

   **Force Field Atom Typing**

The force field ID for an atom is calculated from knowing the atomic number of the atom and the atom type, which is based on the chemical environment of the atom. The atomic number is used as follows:

   \[ \text{Forcefieldid} = \text{atomic number} \times 100 + \text{atom type}; \]

The atom type is an integer: 0\leq \text{atom type} \leq 99. This allows Carbon, or any other element, to have 100 atom types. The atomic number is shifted to the 100s position in the forcefieldid. This algorithm works for all atomistic force fields.

   **Bulk Modulus Calculation**

The bulk modulus (B) is a function of the change in pressure (\(\Delta P\)), change in volume (\(\Delta V\)) and initial volume (\(V_0\)) of the MD cell.

   \[ B = \frac{\Delta P}{\Delta V} \frac{1}{V_0} \]

   **Figure B.1 Bulk modulus calculation**
10. Appendix C – Self-avoiding Walk Software

A self-avoiding walk (SAW) is one where a polymer chain is constructed in such a manner that it does not collide with itself. The light cycle race in the movie TRON is a perfect example of a self-avoiding walk. The paths of the light cycles defined an area on the ground that neither light cycle could touch. In the case of a 3D crosslinked epoxy atoms cannot occupy the same space – obviously not avoiding.

Self-avoiding walk (SAW) software was new, original software created along with the software modifications to NanoEngineer-1, PACKMOL and MSI2LMP to create MD cells using a static method. The concept behind SAW is one of growing polymer chains based on the geometry of a molecular template. The square DGEBA/IPD oligomer used to populate a MD cell using PACKMOL is a template. It specifies the geometry of the oligomer regardless of the “attitude” of the oligomer. The oligomer can be translated and rotated in 3D but the atoms remain in the same positions relative to one another. The oligomer is ring-shaped. When the hole faces the viewer and then rotated about the vertical axis the molecule turns so only a side is seen thereby changing the molecule’s attitude. This occurs with any axis through the molecule with respect to the viewer. The MM energy minimization is run on the template molecule prior to being used by SAW to create an MD cell. The hope is that the molecule will occupy the smallest amount of space possible so that a higher initial density can be achieved.

The SAW software creates a user-defined number of starting points inside the MD cell and in a round robin fashion adds an atom to the molecule to be built. Each starting point or “build spot” is given a pseudo random set of angles to rotate the molecule to be built
and the build spot position is pseudo randomly generated. As atoms are added to each molecule being constructed at a build spot its position is checked against the user-defined boundaries of the MD cell to prevent atoms from leaving the MD cell. Any one molecule being built that collides with the MD cell boundary or another molecule is erased from the MD cell once building of the molecule is retried some user-defined number of times. Figure c.1 shows the hotspot chooser feature added to NanoEngineer-1. The hotspot is identified in the toolbar on the left-hand side of the application. Two atoms compose a hotspot. By clicking on the hotspot button the atoms that compose the hotspot are highlighted.
Figure C.1 Hotspot Chooser: Green terminal Hydrogen atoms are the Hotpots.
Figure C.2 Geometry Chooser: Highlighted green Carbon Atoms form a triangle for collision detection.
A molecule can be just a molecule or it can contain hot spots. A plain molecule such as water is built at a pseudo random position in the MD cell. A solvated protein would be surrounded by hundreds of water molecules. Each water molecule is built at a build spot. A molecule, such as DGEBA or IPD, will ideally be attached to the opposite molecule. DGEBA attaches to IPD and IPD attaches to DGEBA and a network is formed. A chain of atoms will extend from each build spot out into random directions. Assuming multiple build spots there will be multiple networks in the same MD cell. A DGEBA molecule will have two hot spots on it while an IPD molecule contains four hot spots. SAW uses a master control file (MCF) to define the names of molecule templates, the size of the MD cell, and many other parameters needed to build an MD cell. Also, NanoEngineer-1 was modified to allow the user to define hot spots on the molecule template. Essentially, the software walks the list of atoms in the molecule template and calculates the necessary translation and rotation from the template coordinate frame to the MD cell coordinate frame. When the software encounters a hot spot it adds the hot spot to the list of build spots and atoms are added to the new molecule in the same round robin fashion. The number of new build spots can increase with each new molecule built, but the MD cell boundaries and collisions with neighboring chains of molecules will result in the eventual termination of the software. The user may specify a desired MD cell density to achieve as a software termination method. Also, a convergence of the MD cell density to some tolerance value, e.g. 1.0e-6, is also provided to the user via the MCF. See Appendix A for a sample MCF file.
All MD cell creation software suffers from the problem of molecules colliding. Some researchers discuss the need to solve ring spearing or simply atoms being too close together or an atom being located within the bond between two atoms. As seen in the literature review in chapter two the collision check must be done between two atoms, which is quadratic in time. The SAW software compares atoms with atoms, atoms with bonds (i.e. line segments) and atoms with planes. It also compares bonds with planes. A pair of covalently bonded atoms form a line segment and a third atom should not fall inside the line segment. A bond should not pass through a plane. Planes are found in ring systems. Any ring will have (n-2) number of planes where n is the number of atoms. Benzene and cyclohexane both have 6 atoms and therefore 4 triangles. The difference between the two molecules is the triangles in benzene are nearly in the same plane compared to cyclohexane. The SAW software must know the location of all planes in the molecule template for collision detection. NanoEngineer-1 was modified to allow the user to define planes in the structure of the molecule. Figure C.2 shows the geometry chooser. The highlighted Carbon atoms form a triangle to test for collisions. The highlighted triangle is one of four triangles defined and identified in the toolbar on the left-hand side of the application. By clicking on the button for a triangle the atoms in the triangle are highlighted. This information is stored in the MMP file and is read by the SAW software. The manual triangle creation was adopted due to the complexity and time required to write and integrate a ring perception algorithm into NanoEngineer-1.

Figure C.3 shows a small monomer for software testing purposes based on DGEBA. The two terminal hydrogens constitute a hot spot in the monomer. The software will omit rendering the hydrogens and place a carbon atom in its place and render another
monomer to form a polymer chain. The three carbon atoms are highlighted in green to show the eighth triangle defined in the monomer for collision detection. **Figure C.4** shows the hotspot defined in the toolbar on the left-hand side of the application. Right-clicking on the hot spot in the toolbar allows the user to open it and define the molecules that can be attached to this hot spot and the percentage to be used by the software to determine the frequency of attachment. DGEBA12 is defined as the molecule (this is its file name) and the number 100 represents 100% of the time attach another monomer to this hot spot. A specification of IPD:50, DGEBA:50 programs SAW to pseudo-randomly select on or the other monomer for attachment. Hot spot and triangle information is stored per atom in the MMP file in addition to a two tokens “mmptriangles” and “mmphotspots”, which account for all hot spots and triangles in the MMP file. Just as in the general case a molecule containing rings is represented by a graph so too are the triangles used for collision detection. The graph is maintained in the MMP file.
Figure C.3 Two terminal Hydrogen atoms for the hotspot and the eighth triangle for collision detection for the monomer.
Figure C.4 HotSpot one is shown in the toolbar on the left and its contents in the open dialog box.

SAW implements collision detection by bisecting the MD cell normal to all three axes. This process creates eight sub-cubes in the MD cell. Further bisection of the sub-cubes creates smaller sub-cubes. Figure C.5 shows the first bisection. If the bisection is carried out to five levels 32,768 sub-cubes are created inside the MD cell. Atoms, bonds and rings are tested by the software for a specific sub-cube when an atom is added to the sub-cube thus reducing the computation time required. Similar to PACKMOL; the quadratic algorithm becomes linear. PACKMOL performs a less stringent test for
collisions due to the geometry of the molecules are predetermined and their attitudes are modified during the minimization of the cost function. SAW has the added burden of building a molecule at the position in space where a previous molecule was completed and creating a bond between the two molecules as shown in Figure C.8. The next molecule is given pseudo random attitude therefore; there is the possibility of the next molecule colliding with the previous molecule to which it is attached. This is like a piece of yarn that folds back onto itself. Folding back is permitted, but collisions are not.

SAW was initially developed to create amorphous structures and after some testing of the software it could be used to create crystalline and semi-crystalline structures as well.

Figure C.5 Bisection of the MD cell normal to all 3 axes for collision detection. Eight sub-cubes are formed
Figure C.6 Twenty DGEBA-like polymer chains rendered by SAW.
Figure C.7 Close up of rendered DGEBA-like polymer chains.
Figure C.8 Highlighted carbon-carbon bond SAW created bond between two monomers
11. Appendix D – Force Field Details

The files for the CFF91 and the numeric version of the CFF91 force fields are on the DVD.
12. Appendix E – Software Modification Details

Figure E.1 NanoEngineer-1 Blank Canvas.
Figure E.2 NanoEngineer-1 Drawing Chunk Mode.
Figure 3.3 Force field chooser allows user to select a set of atoms types for a specific force field
Figure E.4 CFF91 force field is selected and the aromatic carbon (cp) atom type is selected
Figure E.5 Oligomer with aromatic carbon highlighted in yellow – tooltip shows atom details.
Figure E.6 Oligomer showing the DGEBA and IPD molecules
Figure E.7 Preferences tooltip modification allows display of alphanumeric atom types
13. Appendix F – Acronyms and Symbols

3D – three dimensions
AP – Amidopolyamine
AMBER - Assisted Model Building with Energy Refinement
DREIDING – force field
CAD - computer aid design
CHARMM - Chemistry at Harvard Macromolecular Mechanics
CFF91 - Consistent Force Field for the year 1991
COMPASS - Condensed-phase Optimized Molecular Potentials for Atomistic Simulation Studies
DDA - dicyandiamide
DDM – 4, 4’-diaminodiphenyl methane
DETA - Diethylenetriamine
DGEBA - diglycidyl either of bisphenol A
FEA – Finite Element Analysis
FEM – Finite Element Methods
FF – force field
IMP – Imidazoline Polyamine
IPD - isophoronone diamine
K_B – Boltzmann’s Constant
LAMMPS - Large-scale Atomic/Molecular Massively Parallel Simulator
GPa - Giga-Pascal
imp_gen – an open source tool created by the author for the purpose of creating a LAMMPS input file from a variant version of the Nanorex Molecular Machine Part file
LGIF – LAMMPS geometry input file
MPa – Mega-Pascal

MS - Materials Studio

MD - Molecule Dynamics

MM – Molecular Mechanics (usually refers to minimizing the potential energy)

MMP – Molecular Machine Part file. Contains the molecule(s) created in the NanoEngineer-1 software tool

MSD – mean squared displacement

NanoEngineer-1 – Molecular CAD software created by Nanorex

Nanorex – A company that GPL’d NanoEngineer-1 software

OPLS - Optimized Potentials for Liquid Simulations force field

OPLS-aa all atomic force field

OPLS-ua united atom force field

OSS – Open source software

Qeq – The charge equilibration approach

SAW – Self-avoiding walk or also referred to as a random walk from mathematics. Also open source software created by the author to create an MD cell.

T_g – Glass Transition Temperature

VMD - Visual Molecular Dynamics