2013

Dynamic mechanical analysis and computer simulation of polyurea aerogels

Victoria Ann Prokopf

Department:

DYNAMIC MECHANICAL ANALYSIS AND COMPUTER SIMULATION OF POLYUREA AEROGELS

by

VICTORIA ANN PROKOPF

A THESIS

Presented to the Faculty of the Graduate School of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE IN AEROSPACE ENGINEERING

2013

Approved by

Lokeswarappa R. Dharani, Advisor
Nicholas Leventis
Jeffery S. Thomas

ABSTRACT

Historically, the use of aerogels has been limited due to their poor mechanical properties. However, polyurea aerogels have proven to be mechanically strong under quasi-static conditions. Polyurea aerogels can be created by filling molds of the desired shape with a liquid solution that then creates a solid gel filled with liquid. The liquid can be removed from the gel by supercritical drying. This thesis outlines dynamic testing as well as simulation for polyurea aerogels.

Testing has been conducted in dynamic tension and bending for densities of polyurea aerogels ranging from 0.12 g/cm^3 to 0.31 g/cm^3. In most cases, the mechanical properties were minimally affected when tested over a range of frequencies. In tension the previously observed increase of stiffness with density was not present. In this case the 0.17 g/cm^3 has the lowest storage modulus.

Simulations were performed to develop a better understanding of structure-property response of highly porous polyurea aerogels. Micro-scale effects such as particle stiffness, bond strength, and particle frictional coefficients were incorporated into the macro-scale structure-property relationship for the prediction of the Young's modulus. Compression simulations were performed and compared to the corresponding experiment.

ACKNOWLEDGMENTS

I would like to express my appreciation for everyone who made this project possible. First I would like to acknowledge my advisors Drs. Lokeswarappa Dharani and Nicholas Leventis who provided focus and encouragement. Dr. Jeffery Thomas provided his laboratory and knowledge that led to the work necessary for this project. Ph.D. candidate Chakkaravarthy Chidambareswarapattar, who worked to set up and calibrate the DMA, was instrumental to this project and provided a significant amount of troubleshooting.

The Army Research Office has funded this research under Award W911NF-10-1-0476. Without the Army support this project would not have been possible.

Finally, I would like to thank my family and friends, especially Sean Thoma, who supported me during my time in Rolla and pushed me to keep going.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1 GENERAL

Aerogels are low density solids with high porosity and surface area [1, 2]. The first aerogels were reported by Steven Kistler in 1931. He was attempting to remove the liquid from wet gels without disturbing the solid structure. Initially Kistler attempted to achieve this by allowing the solvent to evaporate, but the surface tension of the receding liquid caused the solid structure to collapse. To resolve this issue Kistler dried the wet gels by a process known as super critical drying. This process involves heating the wet gels while under a pressure greater than the vapor pressure. The liquid was converted to a gas all at once and can be removed without damaging the structure [3]. Initially this process was done with water, but super critical water was found to be such a powerful solvent that the silica would dissolve destroying the structure. The process was modified by replacing the water with alcohol through a series of solvent exchanges.

Aerogels created from silica exhibited low thermal, acoustic, and electrical conductivity [4-6]. Poor mechanical properties such as strength limited the use of silica aerogels [7, 8]. However they proved useful in spacecraft insulation and Cerenkov radiation detectors in nuclear reactors. The problem was addressed by casting a thin conformal polymer coating over the entire microstructure. The resulting aerogels were three time (3x) more dense than native silica aerogels with increased flexibility and over one hundred times (100x) increase in strength. [4]. After undergoing this process silica aerogels could be used in structural applications. Based on the results obtained from this modification, aerogels were developed that consisted of an entirely polymer microstructure such as polyurea [9].

1.2 PRIOR WORK ON PUA

1.2.1 Fabrication of Polyurea Aerogels. Loebs [9] formulated the polymer aerogels (PUA) by reacting triisocyanate Desmodur N3300A and water using triethylamine as a catalyst in a solution of acetone. Aerogels created using this reaction consist of polyurea and are referred to as polyurea aerogels (PUA). Three densities of PUA were investigated; these densities are chemically identical despite a difference in density. The three densities investigated were 0.12 g/cm^3, 0.17 g/cm^3, and 0.33 g/cm^3 which represent the upper end of the densities that are easily produced. Formulas used to produce these densities have been included in Table 1.1 [9].

Table 1.1 Polyurea Aerogel Formulas; Reference [9]

PUA Recipe	N3300a (g)	Acetone (mL)	Water (mL)	Triethylamine (mL)	Linear Shrinkage (%)	Measured Density (g/cm^3)
11g	11	94	1.77	0.38	1.8	0.12
16.5g	16.5	94	1.77	0.38	2.4	0.17
33g	33	94	0.88	0.19	5.2	0.31

The 0.12 g/cm^3 and 0.17 g/cm^3 recipes form a gel in one hour and then are aged in the molds for 24 hours. The 0.33g/cm^3 recipe was modified by decreasing the amount of catalyst used which increased the time to create a solid gel from one hour to four hours

and were aged in the molds for 24 hours. Using the recipe for one hour resulted in samples with voids and cracks. The four hour recipe created a large number of viable samples.

Machining or cutting the samples to create the desired shape tends to deform or tear the finished gels. The desired shapes are created early in the processing phase to avoid any machining. Previous trials found that molds made from polypropylene do not react with the chemicals in PUA. The molds are made up of three pieces, a sheet with cut-outs of the filling neck and the desired shape sandwiched between two solid sheets. The plates are held together by a series of bolts around the perimeter. Washers on the bolts created an even clamping pressure. A layer of silicone grease sealed the molds so they did not leak. The polypropylene sheets used to make samples for dynamic testing were 0.125 inch thick. This method allowed the samples to be easily removed and created wet gels with the desired shape for testing.

1.2.2 Mechanical Characteristics. Testing under quasi-static conditions has been completed previously [9] for tension, compression, shear, and three point bend. In tension the lower two densities had a linear relationship between density and strength, but for the highest density the relationship was exponential. Under compression the relationship between density and strength was similar to the relationship in tension [9]. Data from quasi-static testing proved vital for dynamic testing. The 3-point bend tests did not fail before 5% strain as prescribed by the ASTM standard. Therefore these values are informational only and should not be used in design. However the data collected from this test is needed to complete dynamic testing for 3-point bend. The mechanical properties from this testing are listed in Table 1.2.

1.3 RESEARCH OBJECTIVE

This project represents the second stage in the process of understanding the physical properties of PUA. Dynamic mechanical analysis will provide data for use of PUA in applications where the load is oscillating. Simulation will provide information about the strength and stiffness of the bonds in PUA as well as provide a model to predict the Young's modulus of similar aerogels.

Table 1.2. Quasi-Static Testing Results [9]

Tension	PUA Density	Young's Modulus (MPa)	Yield Stress (MPa)	Failure Stress (MPa)	Failure Strain (%)
	0.12 g/cm^3	24.1 ± 0.5	0.7 ± 0.03	1.1 ± 0.08	12.5 ± 2.3
	0.17 g/cm^3	37.2 ± 1.3	1.0 ± 0.2	1.7 ± 0.1	13.5 ± 3.0
	0.33 g/cm^3	102 ± 7.2	2.93 ± 0.4	3.9 ± 0.2	6.0 ± 0.6
Compression	PUA Density	Young's Modulus (MPa)	Yield Stress (MPa)		
	0.12 g/cm^3	11.7 ± 4.4	0.4 ± 0.01		
	0.17 g/cm^3	19.3 ± 4.2	0.7 ± 0.1		
	0.31 g/cm^3	69.0 ± 17.9	2.4 ± 0.3		
3 Point Bend	PUA Density	Young's Modulus (MPa)	Yield Stress (MPa)		
	0.12 g/cm^3	33.1 ± 2.5	1.03 ± 0.1		
	0.17 g/cm^3	62.7 ± 6.4	1.9 ± 0.1		
	0.31 g/cm^3	137.9 ± 13.1	4.65 ± 0.4		
Shear	PUA Density	Shear Modulus (MPa)	Yield Stress (MPa)	Failure Stress (MPa)	
	0.12 g/cm^3	8.3 ± 0.6	0.2 ± 0.02	0.4 ± 0.03	
	0.17 g/cm^3	11.7 ± 0.7	0.4 ± 0.04	0.7 ± 0.08	
	0.31 g/cm^3	37.9 ± 2.5	1.2 ± 0.2	1.6 ± 0.3	

2. LITERATURE REVIEW

2.1 MICROSTRUCTURE OF POLYUREA AEROGEL

Each density possesses a unique microstructure that plays a critical role in the strength and stiffness characteristics. Figure 2.1 shows scanning electron microscope (SEM) images of the three densities at 200k magnification [9].

The 0.12 g/cm^3 PUA displays a fibrous network with large open spaces between the fibers. By increasing the density to 0.17g/cm^3 the structure maintains the large open areas, but the fibers have small particles along their length. Increasing the density again to 0.33 g/cm^3 the fibrous structure has been replaced by an entirely particulate structure. This structure has very little space between particles unlike the previous two cases.

An increase in the density results in an exponential increase in strength and stiffness. This relationship could be attributed to the decrease in open space limiting the ability of the structure to move from its original position [9].

2.2 COMPUTER SIMULATION

Ma et al. [10] stated that the most feasible way to understand the mechanical properties of a porous material is computer simulation. To investigate these properties Ma et al. [10] employed the diffusion limited cluster-cluster aggregation (DLCA) algorithm. This algorithm begins by dispersing particles in a cube with no overlapping particles. The particles are then allowed to diffuse through the cube; particles bond irreversibly into clusters when they collide. This process continues until all the particles form a single cluster.

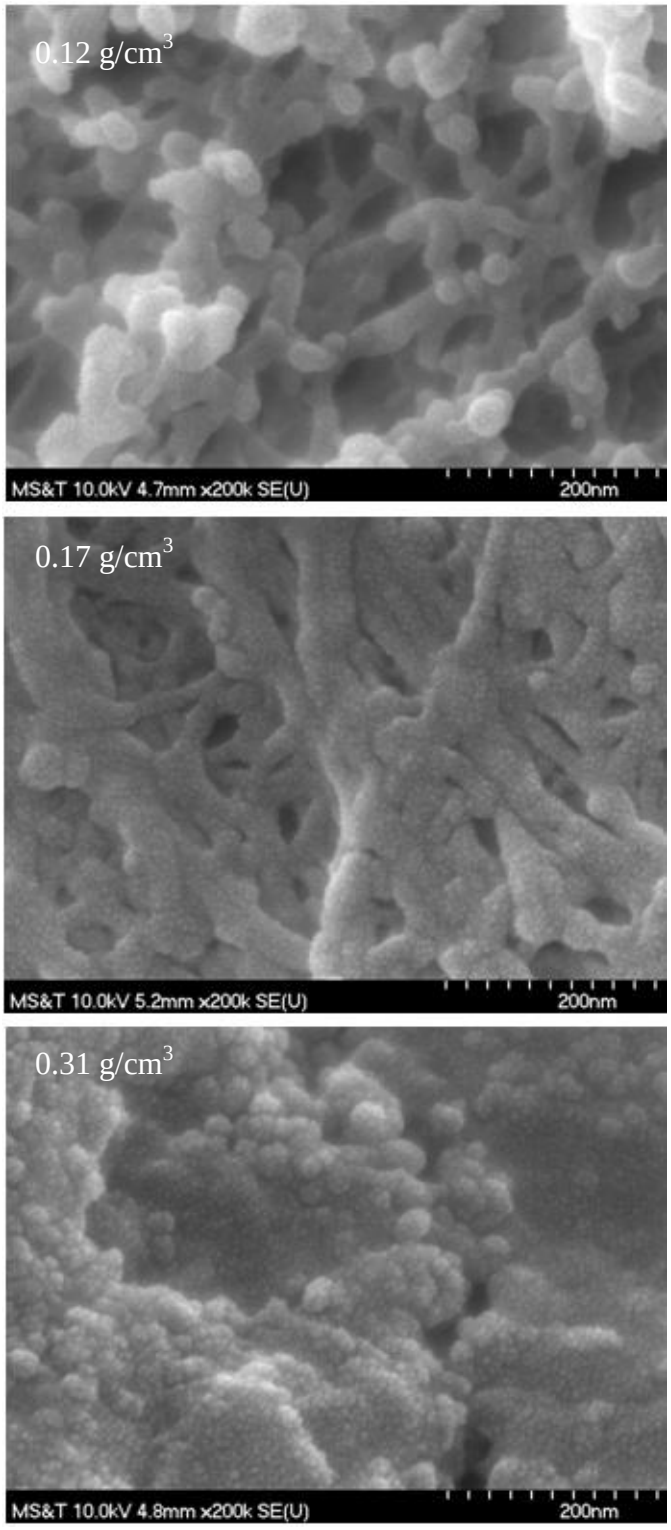

Figure 2.1 SEM Images, 200k x Magnification. Scanning Electron Microscope images have been taken of PUA at a zoom of 200k times. The top photo shows the microstructure of 0.12 g/cm^3 PUA, the middle 0.17 g/cm^3, and 0.31 g/cm^3 is on the bottom [9].

This model can be used in a finite element (FE) program to evaluate properties on both the macroscopic and the microscopic scales. This type of simulation would allow for varying densities. The bonds between particles can be represented by a beam element in an FE program while the particles are rigid nodes similar to Figure 2.2. Dangling masses, branches or particles connected to the main network by a single bond are eliminated to make the created cluster more mechanically efficient. These particles or clusters do not contribute to the mechanical stiffness since their connection to the main network is a single bond and therefore are considered dead weight within the network. The relative density can be calculated by assuming the particles are spheres of unit diameter.

Figure 2.2. Finite Element Model for Aerogel Simulation. The white lines are beam elements that represent particle bonds. The centers of the particles lie on the intersection of these lines and are rigid nodes.

Low density, high porosity materials exhibit a scaling relationship between the elastic modulus (E) and the density (ρ) according to the equation [10]:

$$E \propto \rho^m \tag{1}$$

where m is a scaling exponent. Aerogels typically have scaling exponents greater than 2 suggesting that the porous materials are orders of magnitude more compliant than their solid counterparts. Branches that hang from the main backbone of the network, also known as dangling mass, do not bear load which raises the exponent above 2. Ma et al. [10] attempted to calculate the scaling exponent for silica aerogels using finite element methods. This method resulted in a scaling exponent of 3.6.

Shimpi [11] used a model similar to Ma et al. [10]. The main difference between the two methods was the use of spherical particles by Shimpi rather than cubic. The sphere diameters were determined using the Gaussian distribution function. The particles were tested after every move to determine if they overlapped with another particle. If this occurred the particles were bonded together. This process is shown in Figure 2.3 [11]. The simulation was deemed complete when there was only a single cluster. For this model there were 13800 particles and the cubic space was 400nm in each dimension. To make this model more accurate the dangling masses would need to be removed, but overall the data was similar to that of physical testing in compression.

After creating an aerogel model the positions of the particles were supplied to the Particle Flow Code in 3 Dimensions (PFC3D) input file. Using PFC3D Shimpi [11] determined the mechanical properties of the microstructure by varying the particle

contact stiffness and the normal parallel bond stiffness. The structural parameters are very important to investigate because they dictate the properties of aerogel such as Young's modulus.

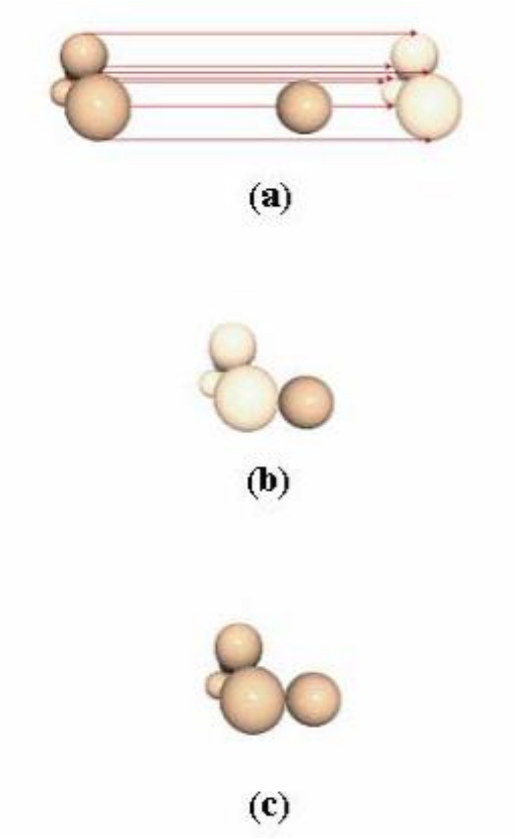

Figure 2.3. A Schematic Showing the Formation of a Cluster in Computer Simulations. When particles collide during the simulation they bond irreversibly and form a cluster [11]

Jullien and Hasmy [12] discuss a modified form of the diffusion limited cluster-cluster aggregation (DLCA) algorithm. This algorithm allows particles to diffuse in a box and when particles meet they form a cluster. Problems with this analysis method arise when the glass transition temperature tends to infinity. To remedy this issue Jullien and Hasmy [12] allow the cluster to deform during aggregation instead of holding the cluster rigid. A cubic box of edge length L containing a unit parameter cubic lattice is examined. Particles are hard cubes of edge length 2 whose centers can move by one unit between sites. Bonding vectors are not allowed to cross. Initially there are N particles in the box that do not overlap with any other particle. Once an overlap is detected a new bond is created as long as neither particle has more than a user prescribed number of bonds. This restriction prevents the formation of a tetrahedron. Particles in a tetrahedron cannot move large distances due to bond restrictions imposed on the model. Infinite clusters can be formed when the box is translated by L in all 3 directions and repeated.

2.3 DYNAMIC MECHANICAL ANALYSIS

Dynamic mechanical analysis (DMA) is an application of an oscillating force to a sample and analyzing the material's response to that force. Properties such as viscosity and stiffness can be obtained with this testing method. The stress applied to the sample at any time can be described by [13]

$$\sigma = \sigma_0 \sin \omega t \tag{2}$$

where t is the time, σ is the stress at time t, ω is the frequency of oscillation, and σ_0 is the maximum stress. The strain of the material can be obtained from

$$\varepsilon(t) = \frac{\sigma_0}{E} \sin \omega t \tag{3}$$

where E is the modulus and ε(t) is the strain at any time t. The phase angle, δ, is the angle between the applied stress and the resultant strain as illustrated in Figure 2.4.

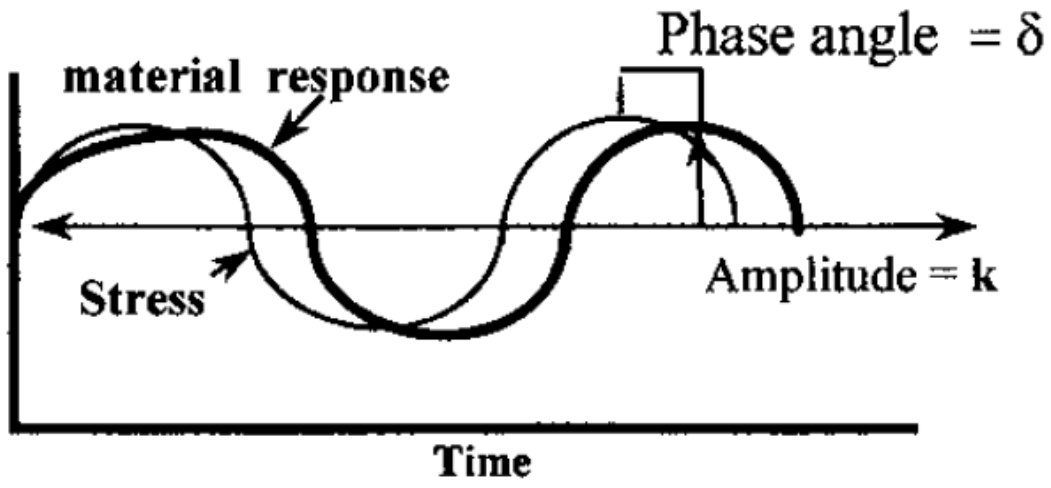

Figure 2.4. A Schematic Showing the Applied Stress Oscillation to a Sample and its Strain Response [13].

According to Menard [13] dynamic mechanical analysis (DMA) can be used to see how modifications can affect the material over a range of frequencies. Complex wave forms can be created by combining several sine waves into a single wave form so that data can be collected for several frequencies in as little as thirty seconds. Temperature and frequency scans can be run together. However since two parameters are changing simultaneously there are concerns over testing accuracy. Menard [13] recommends performing frequency scans isothermally to isolate the effect of frequency on the material. Frequencies outside the scanning range of the DMA can be investigated through free resonance and creep experiments. Creep data would provide data for very low rates of deformation, while free resonance provides results for higher rates of deformation [13]. Not all materials can be analyzed in this way, if the behavior exhibited under creep is not similar to the behavior under DMA then this method cannot be used.

This approach allows for a single modulus to be broken into two components depicted in Figure 2.5, one related to the storage of energy and the other related to the loss of energy. The storage modulus represented by E' is a measure of how well the material stores or returns energy. In an ideal linear elastic material this value would be equivalent to the Young's modulus, E. Energy lost to friction and internal motions is expressed as the loss modulus, E'', also referred to as the viscous or imaginary modulus [13].

The tangent of the phase angle, tan δ, is the most basic property measured through DMA. This is an indicator of how efficiently the material loses energy to molecular rearrangements and internal friction, which is also referred to as the damping. The phase

angle is independent of geometry. As shown in Figure 2.5 the tangent of the phase angle is defined as the ratio of loss modulus to the storage modulus.

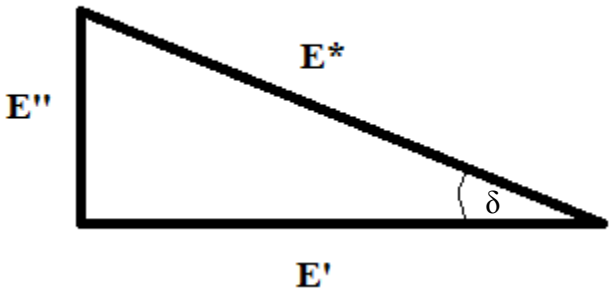

Figure 2.5. Relationship Between the Phase Angle (δ), Storage Modulus (E'), and Loss Modulus (E'').

This information has been used to create a simulation as well as investigate the elastic behavior as a function of frequency of polyurea aerogel.

3. SIMULATION

Simulations were completed to determine the micro-properties such as bond strength and particle stiffness that cannot be calculated experimentally. Further simulations using models of an aerogel with a different density were then completed using the micro-properties found to determine if the model could predict the Young's modulus for the second density. The diffusion limited cluster-cluster aggregation (DLCA) algorithm provided a method for creating polyurea aerogels for simulation. Particle Flow Code 3D (PFC3D) was used to simulate the particle interaction as the micro-parameters were varied.

3.1 DESCRIPTION OF THE DLCA ALGORITHM

The diffusion limited cluster-cluster aggregation (DLCA) Algorithm begins with a cubic volume filled with non-intersecting spheres. The spheres (or clusters) are allowed to move within the cubic volume. The moves are completed incrementally and terminated when an overlap is detected, the edge of the cube is reached, or the maximum number of increments is completed. This process continues until a single cluster forms. The process of aggregation was illustrated in Figure 2.3.

Each sphere has a vector starting at the center extending to a randomly selected point on the surface. This vector represents the positive direction of the dipole; the vector is allowed to rotate with each movement. When two spheres meet the chance of a cluster forming is dependent on the angle between the vectors of the two particles. When the angle between the two vectors is less than forty-five degrees the change that the two

particles form a cluster is zero percent, up to ninety degrees is thirty-three percent, 135 degrees is sixty-six percent chance and 180 degrees is one hundred percent.

3.2 PARTICLE FLOW CODE 3D

Particle Flow Code 3D (PFC3D) published by Itasca [16] models the movement and interaction of rigid spherical particles using the discrete-element method (DEM). PFC3D is classified as a discrete element code because it allows for finite displacements and rotations of particles and recognizes new contacts during calculation. Calculations alternate between the application of Newton's second law to the particles and a force-displacement at the contacts.

The model is composed of distinct spherical particles; these particles interact with one another at contact points. Bonds are created when two particles are in contact with one another. These bonds will break when the strength limit is exceeded. The model is constructed from two parts: the spherical particles or balls and walls.

The calculation cycle is illustrated in Figure 3.1. PFC3D is a time step algorithm that repeatedly applies Newton's second law to the balls, updates wall positions, and applies a force-displacement relationship to all contacts. Contacts can exist between two balls or a ball and a wall. These contacts are created or broken automatically during the simulation. Each time step begins by updating the contacts based on the position of the balls and walls. Each contact has the force-displacement law applied to it, and the contact forces are updated based on the relative motion of the two entities. The velocity and position of each ball are updated based on the resultant forces and moments arising from the contact forces. Wall locations are updated based on the user prescribed velocities.

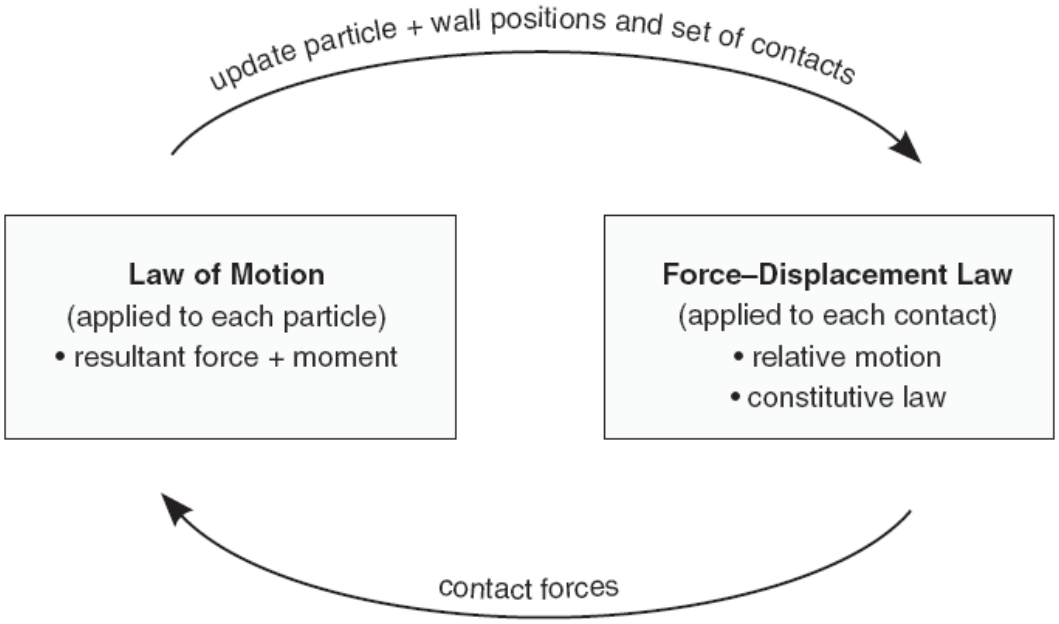

Figure 3.1. Calculation Cycle in PFC3D [16].

3.2.1. Contact Models [16]. Components behave based on stiffness, slip, and bonding. The stiffness model relates the contact forces and relative displacements in both the normal and shear directions. Slip behavior relates shear and normal forces such that two entities may slip relative to the other. If contact bonds exist then slip behavior is not present; contact-bond supersedes slip behavior. Bonding behaviors can be present in two forms, parallel-bonds and contact-bonds. Both bonds act as ‘glue’ joining the two particles. Contact-bonds act only at a point while parallel-bonds are of finite size that acts over a circular or rectangular cross-section between balls. Contact bonds can only

transmit forces but not moments. However, the parallel bonds can transmit both forces and moments. Both bond types may appear in a model simultaneously.

Contact normal bonds behave similar to a point of glue with constant normal and shear stiffness at the contact point. The bond has specified shear and tensile normal strengths. Once the magnitude of the shear or tensile normal strength is exceed the bond breaks. No slip is possible while a bond is present due to the glue-like connection between two particles.

In a parallel bond the two spheres behave as though a cement-like material has been deposited between the two. Forces as well as moments may be transmitted through parallel bonds unlike the contact bonds. Parallel bonds do not preclude the possibility of slip but does not ensure that slip is present. A set of elastic springs can represent a parallel bond; these springs have normal and shear stiffness distributed over the contact plane centered at the contact point. An illustration of the bond is shown in Figure 3.2.

3.2.2. Selecting Material Properties for PFC3D Model. Stiffness and strength must be assigned so that the behavior of the model mimics the behavior of laboratory specimens. In general, the model parameters cannot be directly related to material properties due to the impact of particle size and packing arrangement. Selecting the micro-properties requires a procedure relating the deformability and strength micro-parameters such as particle stiffness, bond stiffness, and bond strengths corresponding to their macro-responses. Through a calibration process a set of micro-parameters can be determined for a particular material; this process is described in Figure 3.3.

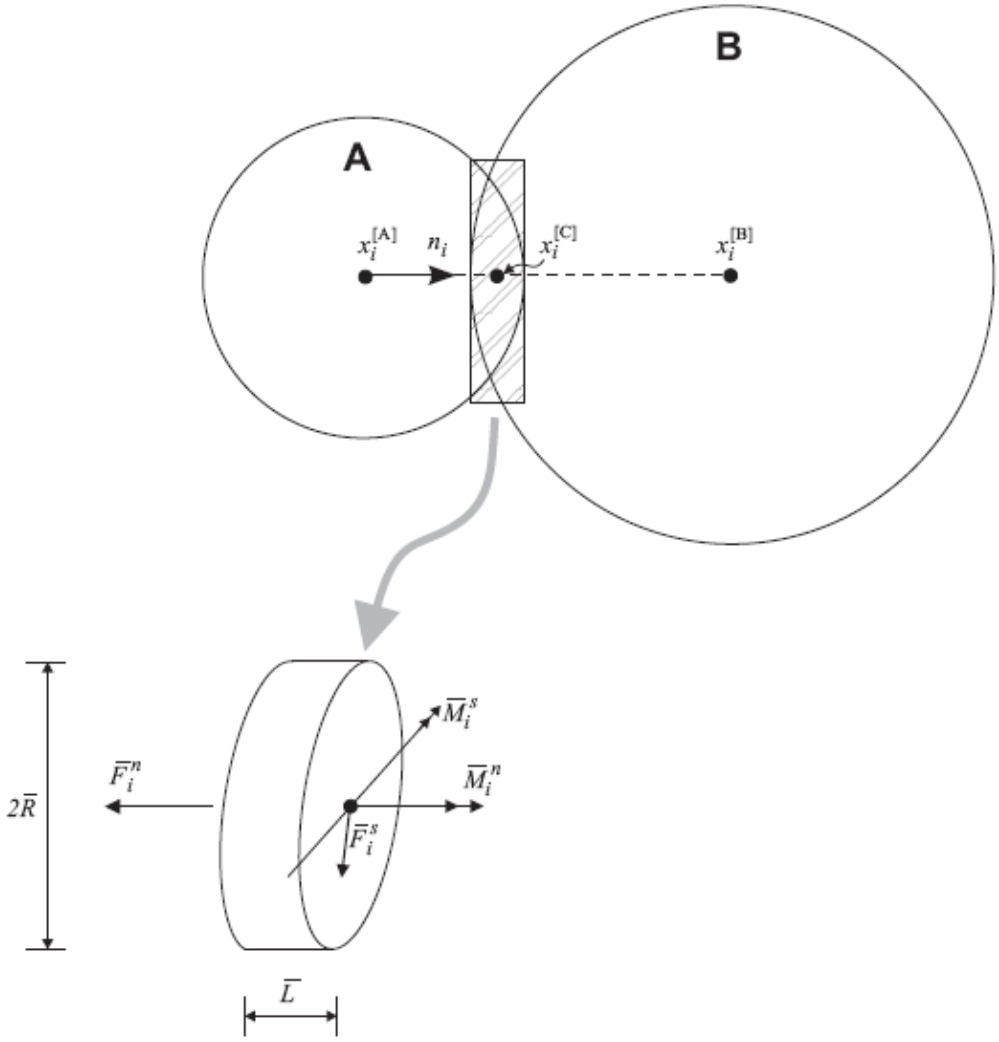

Figure 3.2. Illustration of a Parallel Bond [16].

The initial estimations can be made from the known material properties found through experimental testing. PFC3D provides the following two equations for the first estimate of the particle stiffness and bond strength, respectively [16].

$$E = \frac{K_n}{4R} \tag{4}$$

where E is the Young's modulus from the experimental data, R is the average particle radius, and K_n is the normal stiffness of the particles.

$$\sigma_t = \frac{S_n}{4R^2} \tag{5}$$

where σ_t is the tensile strength measured during laboratory testing and S_n is the normal bond strength. The relationship between the parallel bond stiffness, $\overline{k^n}$, and the modulus, E_p, is given by

$$\overline{k^n} = \frac{E_p}{L} \tag{6}$$

The shear stiffness K and shear bond strength S_s are taken to be equal or a fraction of their normal counterparts. While these equations were designed for a cubic arrangement of particles, they provide useful information for the first estimate of the micro-mechanical parameters [17].

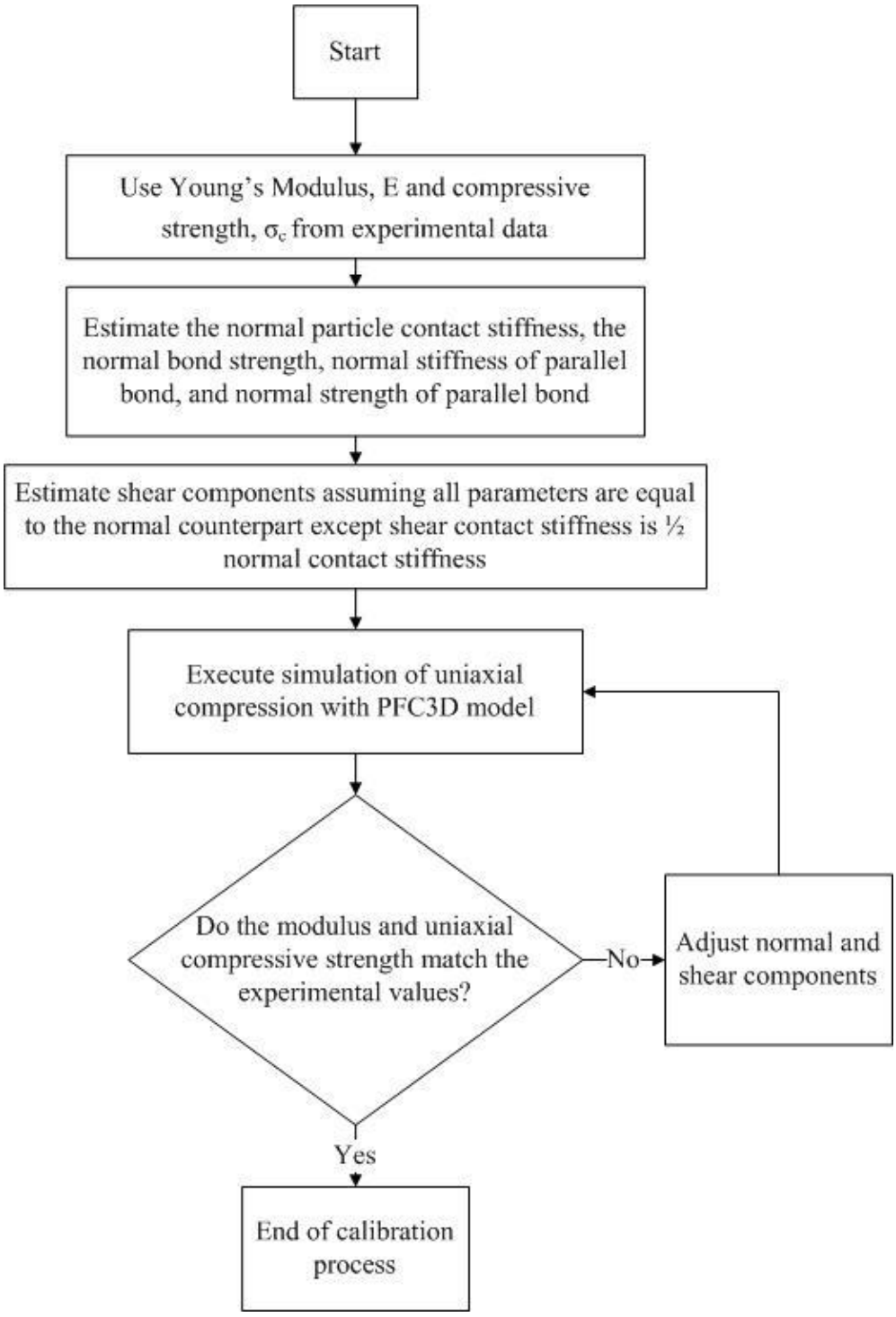

Figure 3.3. Flow Chart of Calibration Process for PFC3D [11].

4. TESTING

Dynamic testing was completed to investigate the change in elasticity over a range of frequencies. Quasi-static testing only provided insight into the static properties of PUA not how the modulus varies over frequencies. This information could be critical if PUA was used in a structure that was not completely static. Difficulties in fabrication led to a modification of the formulas used to create 0.33 g/cm^3 PUA samples. Tension tests were completed by pulling the sample apart with an oscillatory stress. Three point bend tests were completed by placing a specimen on two knife edges 50 mm apart and oscillatory loading in the center of the beam. The same range of frequencies was used for both tests.

All dynamic experiments were conducted using the Dynamic Mechanical Analyzer Q800 (TA Instruments, New Castle, DE) shown in Figure 4.1.

4.1. SAMPLE FABRICATION

Samples were fabricated using the process discussed in Section 1.2.1. The formulation for 0.33 g/cm^3 samples was modified due to difficulty creating aerogels with no voids or cracks. Table 4.1 shows the updated formula for 0.33 g/cm^3 along with the formulas used for the other two densities. The decrease in the amount of catalyst slowed the gelation time allowing any gasses created to escape rather than trap them creating voids in the samples. In addition to a change in the formula, the 0.33 g/cm^3 samples were only aged in the molds for four hours rather than 24 hours as in the other two densities.

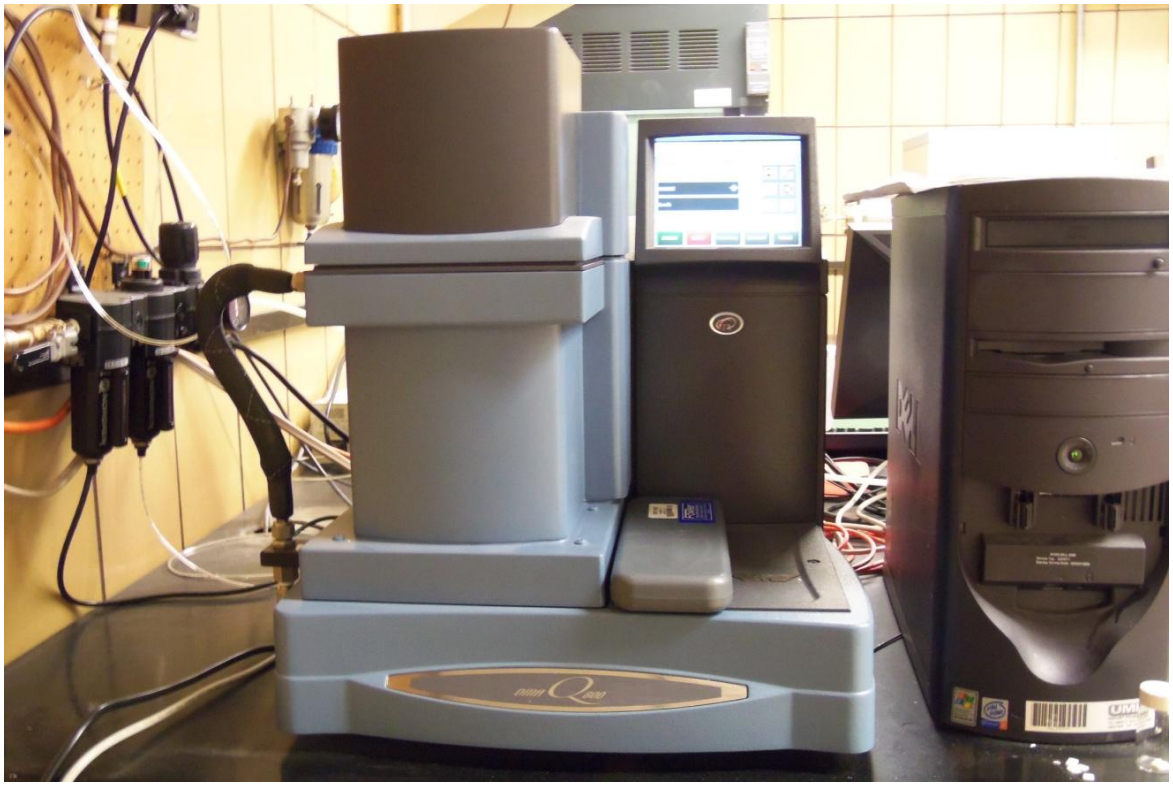

Figure 4.1. DMA Setup.

Table 4.1. Updated Polyurea Aerogel Formulas.

PUA Recipe	N3300a (g)	Acetone (mL)	Water (mL)	Triethylamine (mL)	Linear Shrinkage (%)	Measured Density (g/cm^3)
11g	11	94	1.77	0.38	1.8	0.12
16.5g	16.5	94	1.77	0.38	2.4	0.17
33g	33	94	1.178	0.148	5.2	0.31

4.2. TENSION TEST

ASTM D5026 outlines the procedures standardized for testing plastic specimens in dynamic tension. This standard provides very little guidance on specimen sizes due to the wide variety of dynamic mechanical analyzers available. The Q800 model allows for a rectangular specimen two millimeters thick and six millimeters wide. These dimensions were used to create appropriate test specimens shown in Figure 4.2. Scans were completed with frequencies ranging from 0.5 Hertz to 16 Hertz at 25°C. The stress amplitude of oscillation was 1 N with a preload force of 0.01 N. Using 1 N ensured that deformation would remain in the linear elastic region for the material as prescribed by ASTM D5026 [18]. Five specimens of each density were tested. Figure 4.3 shows a tension specimen loaded and ready to test.

Over the range of frequencies investigated the change in storage modulus for all densities was very small as shown in Figure 4.4. At the lowest frequency, 0.5 Hz, the storage modulus was 40.95 MPa and 25.33 MPa for 0.12 g/cm^3 and 0.17 g/cm^3, respectively. As in quasi-static testing discussed in Section 1.2.2, the 0.33 g/cm^3 was significantly higher than the lower two densities. However, in this test the storage modulus was higher for the 0.12 g/cm^3 than for 0.17 g/cm^3 meaning that 0.12 g/cm^3 PUA has the ability to store more energy than 0.17 g/cm^3. Figure 4.5 shows the tan delta as a function of frequency for all densities. The 0.17 g/cm^3 PUA when loaded in tension will damp the oscillatory motion faster than the other two densities for all frequencies studied. This is due to the mixed microstructure of 0.17 g/cm^3 PUA. The combination of fibrous and particulate microstructure allows the 0.17g/cm^3 to damp the motion more efficiently than the other two densities.

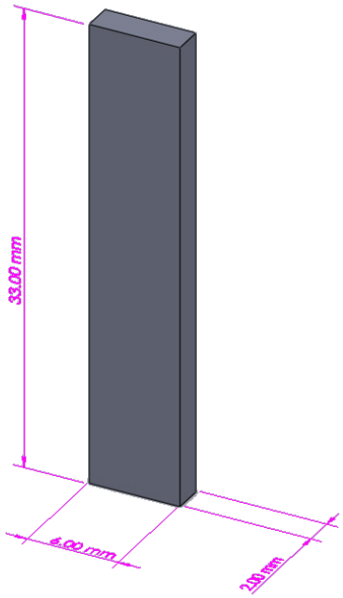

Figure 4.2. Isometric View of a Tension Test Sample Showing the Dimensions for use in the Q800 DMA.

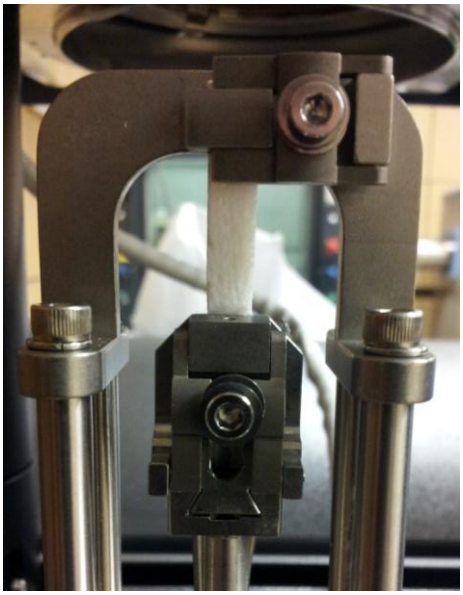

Figure 4.3. Tension Test Setup. The Q800 has been fitted with two clamps; the bottom clamp oscillates during testing.

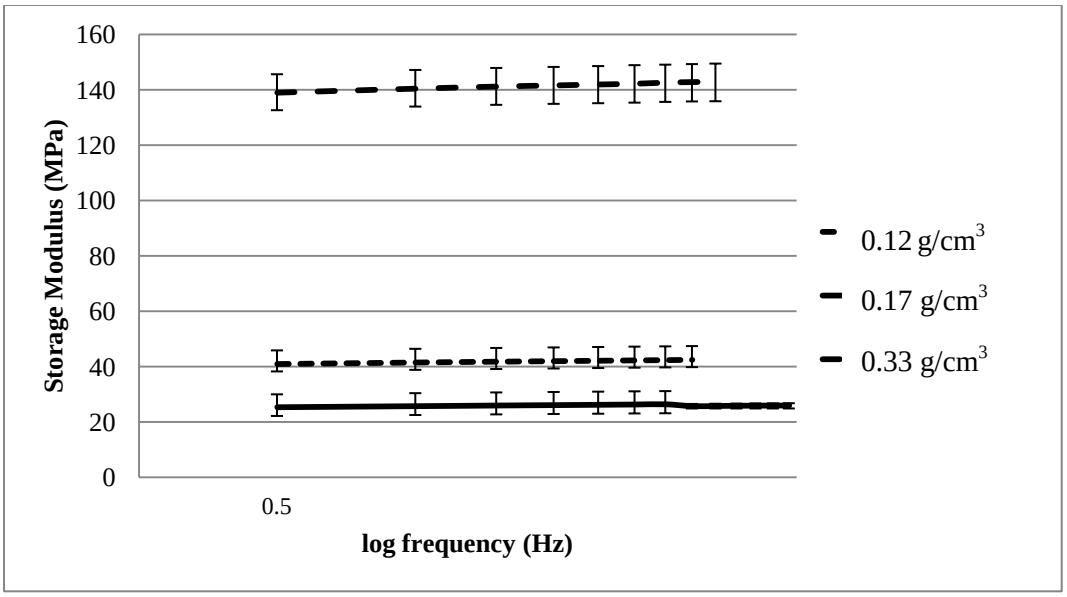

Figure 4.4. Dynamic Tension Test Results for PUA Showing the Variation of Storage Modulus with Frequency for the Three Densities.

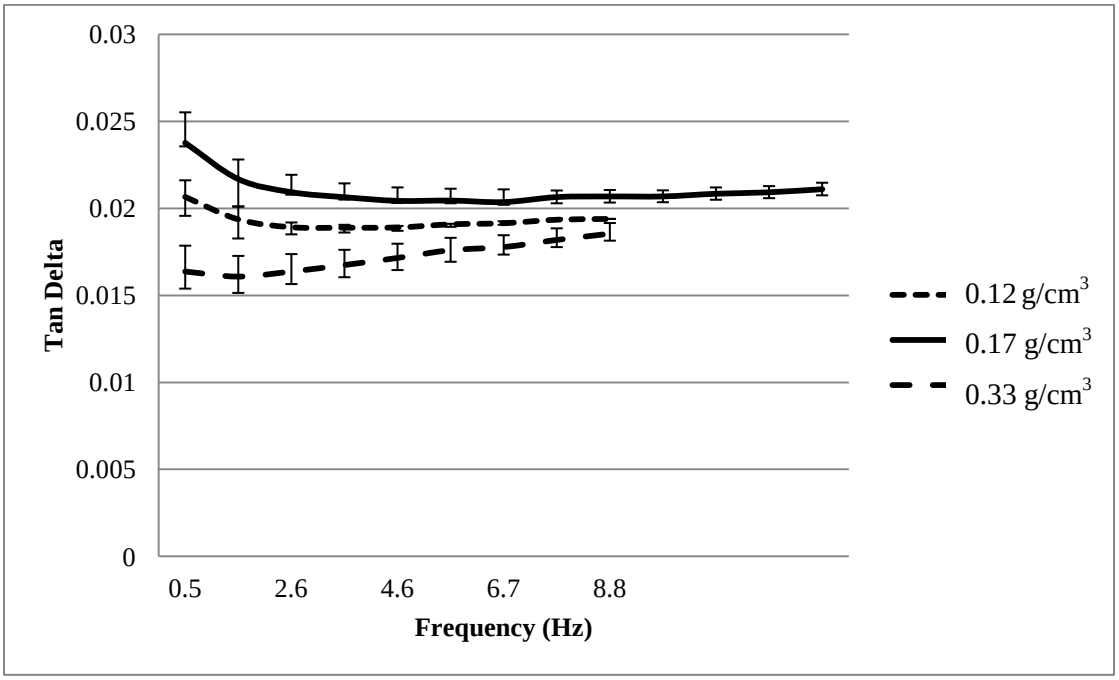

Figure 4.5. Damping Tension Test Results for PUA Showing the Variation of Phase Angle with Frequency for the Three Densities.

4.3. BENDING TEST

ASTM D5023 outlines the procedures for testing plastic specimens in dynamic bending. Specimens have a span to depth ratio of 16 with a 10% of the support span overhang on each end to prevent the specimen from slipping off the supports. Typical specimens are 64 mm x 13 mm x 3 mm rectangular beam as shown in Figure 4.6. Scans were completed with frequencies ranging from 0.5 Hertz to 16 Hertz at 25°C. The amplitude of oscillation was taken to be 5 μm with a preload force of 0.01 N. Using 5 μm ensured that deformation would remain in the linear elastic region for the material as prescribed by ASTM D5023 [19]. Strain amplitude was used rather than a prescribed force due to difficulties with samples yielding in bending. Five specimens were tested. Figure 4.7 shows a specimen loaded into the machine, ready to test.

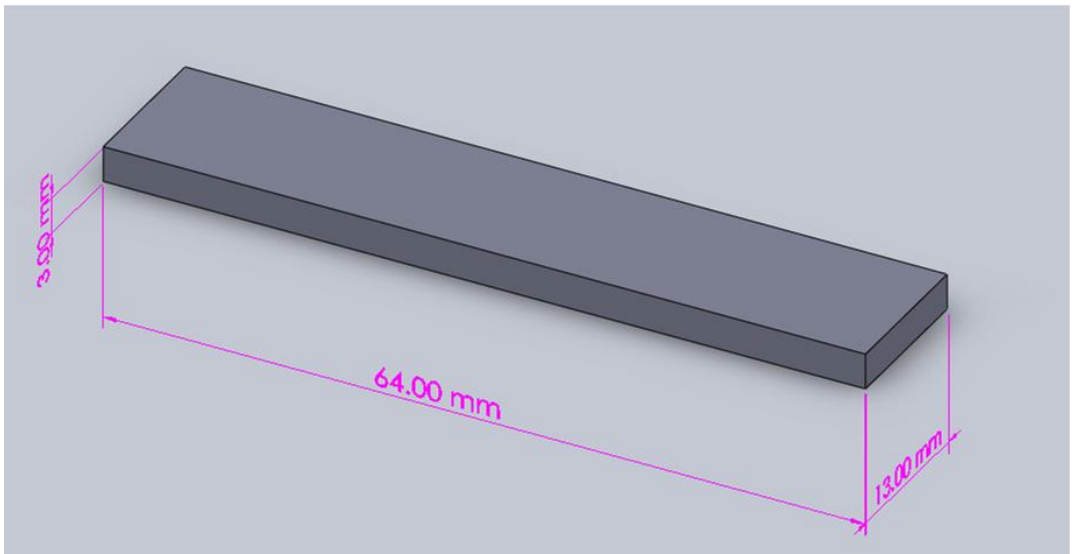

Figure 4.6. Isometric View of a Bending Test Sample Showing the Dimensions Used.

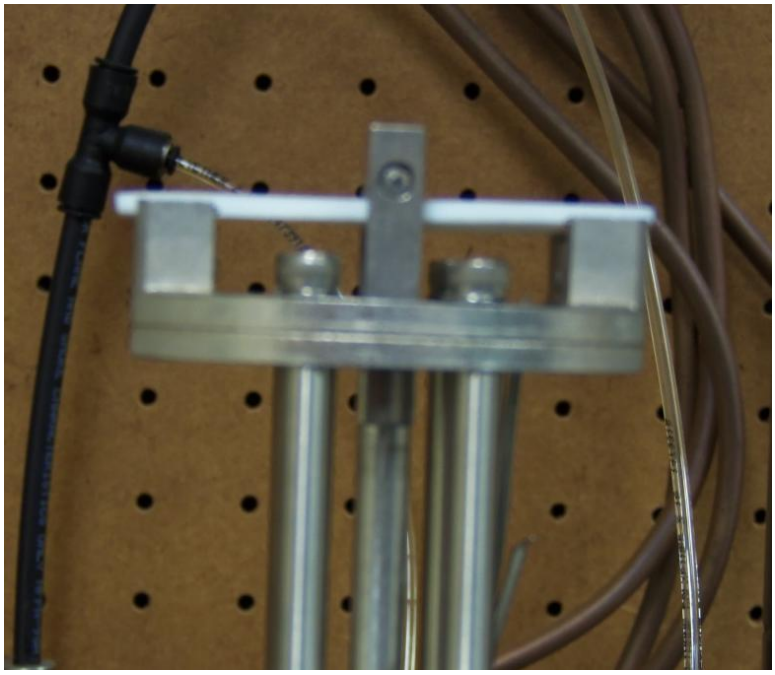

Figure 4.7. Bending Test Setup. The Q800 has been fitted with two stationary supports with a specified span. The center member loads the sample with varying frequencies.

The increase in the storage modulus over the frequency range of 0.12 g/cm^3 and 0.17g/cm^3 was very small as shown in Figure 4.8. At the lowest frequency the storage modulus was 48.44 MPa and 73.24 MPa for 0.12 g/cm^3 and 0.17 g/cm^3, respectively. The highest density also exhibited a similar exponential relationship between density and storage modulus previously discussed. The 0.33 g/cm^3 experiences a greater increase in modulus than the other two densities.

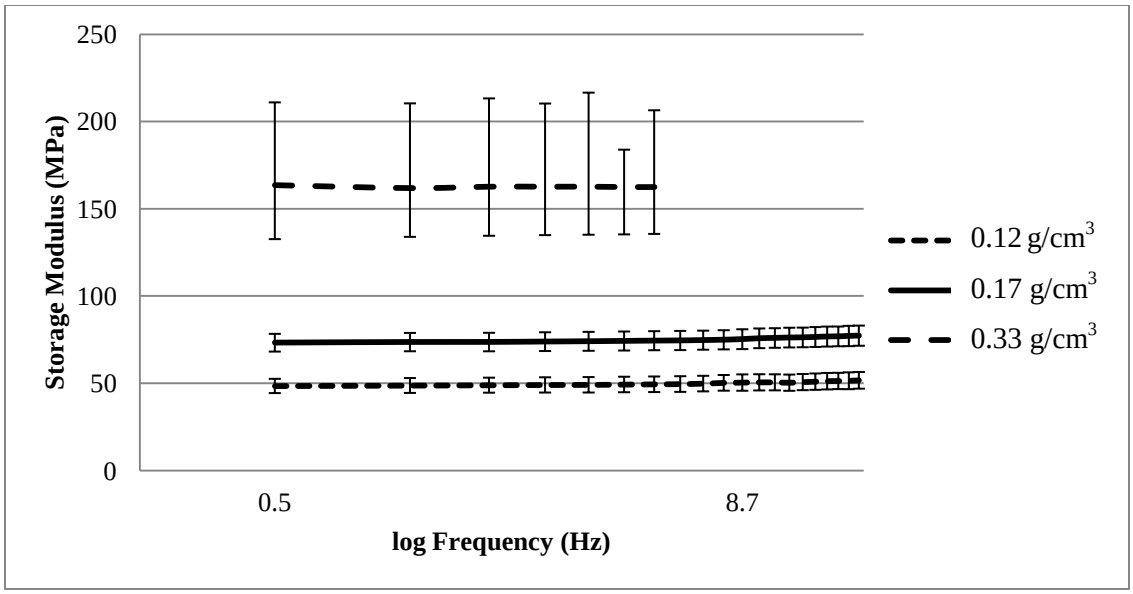

Figure 4.8. Dynamic Bending Test Results for PUA Showing the Variation of the Storage Modulus with Frequency for the Three Densities.

5. DISCUSSION OF RESULTS

Simulations were completed to determine the micro-properties such as bond strength and particle stiffness that cannot be calculated experimentally. These parameters were then used to calculate the Young's modulus of another aerogel with a similar microstructure. This would allow an estimate of the Young's modulus without laboratory testing. The diffusion limited cluster-cluster aggregation (DLCA) algorithm provided a method for creating polyurea aerogels for simulation. Particle Flow Code 3D (PFC3D) was used to simulate the particle interaction as the micro-parameters were varied.

Dynamic mechanical analysis (DMA) showed how the storage modulus (E') changes over a range of frequencies. The phase angle, tan δ, provided information on the damping behavior of the three densities.

5.1 NUMERICAL MODELING

Figure 5.1 shows the model created using the DLCA algorithm discussed previously. The output of the DLCA code, a table consisting of particle radius and its location in 3D space, was supplied to the PFC3D input file. The cube size was chosen based on the largest cell size possible while maintaining computational efficiency. A total of 33 particles were created in a cube with a 350 nm edge length shown in Figure 5.1.

Once the model is created in PFC3D it is calibrated by estimating initial values for the particle stiffness and bond strength then adjusting these values until the modulus in compression of the simulation matches the experimental value. The values of the normal and shear particles' stiffness are estimated using equation 2. The value of R is assumed to be the average particle radius or 36.73 nm. The expected Young's modulus of

the particle-particle contact, the micro-modulus, is equated to the actual modulus, the macro-modulus (E), estimated from the previous compression experiments [9]. The initial estimate for the particle normal stiffness was 2.83 N/m with a shear stiffness of 1.41 N/m. The parallel bond stiffness, k_{np}, was calculated using equation 6. The bond length was assumed to be the diameter of the particles, and the Young's modulus of the parallel bond was taken as the experimental compressive modulus [9]. Initially k_{np} was estimated to be 5.81×10^{13} N/m. The shear stiffness was found to be the same value as the normal stiffness. Setting the shear stiffness to half of the normal stiffness resulted in much lower moduli than needed.

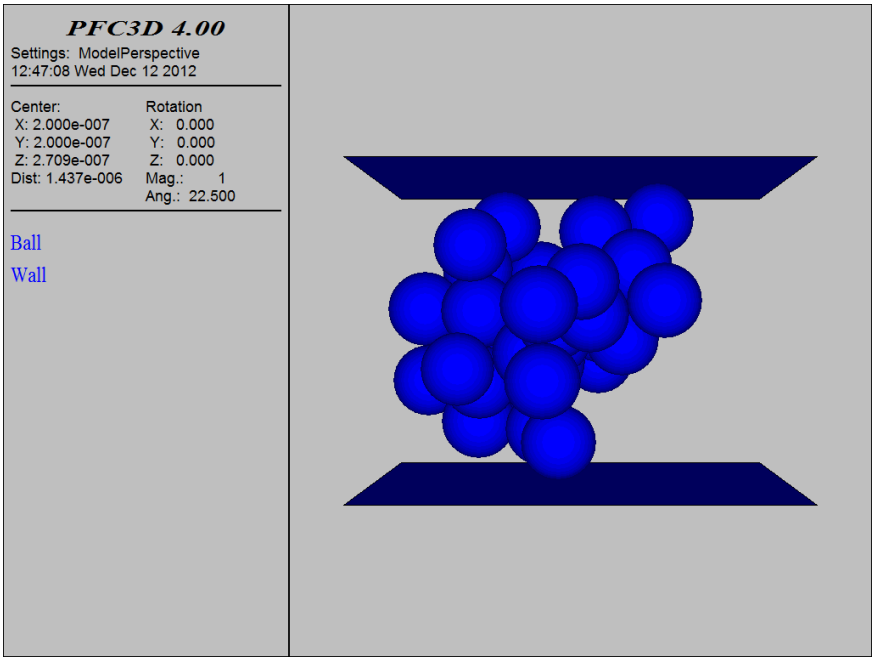

Figure 5.1. Model Created using the DLCA Algorithm. This model is used to determine the micromechanical parameters for PUA.

The strength of the parallel bond was 80 MPa or the stress at which cracks started to appear in the compression samples. Normally this value would be the failure stress, but this value could not be determined from the data. All parameters except the particle stiffness remain constant for the calibration curve shown in Figure 5.2. The compression modulus varied nonlinearly with the particle stiffness. The calibration curve, a curve created by varying particle stiffness, was expected to be linear due to equation 4. Particle stiffness affects the level of equilibrium achieved by PFC3D simulations causing a nonlinear relationship between particles stiffness and Young's modulus. Table 5.1 summarizes the micro-mechanical parameters determined using the calibration process discussed.

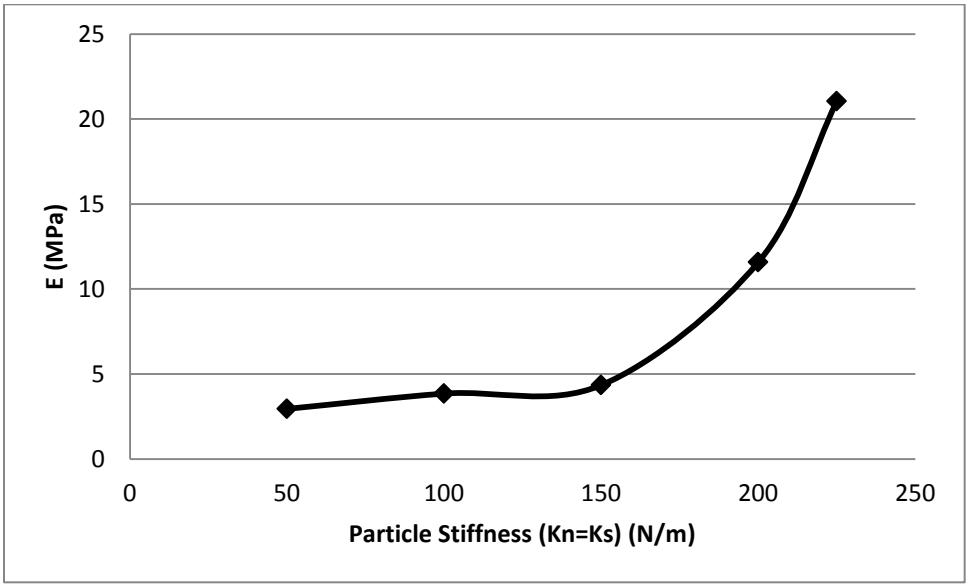

Figure 5.2. Calibration Curve of PFC3D Model Showing the Variation of Particle Compressive Modulus of the Particle Assembly with Particle Stiffness.

Table 5.1. Parameters for PFC3D Model.

	Value	Units
BALL PARAMETERS		
Normal Stiffness	225	N/m
Shear Stiffness	225	N/m
Friction	0.6	
BOND PARAMETERS		
Normal Stiffness	4.05E+15	N/m
Shear Stiffness	4.05E+15	N/m
Normal Strength	80	MPa
Shear Strength	80	MPa

Using these parameters a second model was created using the 0.33 g/cm^3 density. The same parameters were applied to the model as used in the 0.165 g/cm^3 simulation. This model predicted a Young's modulus of 72.4 MPa for 0.33 g/cm^3. Previous testing [9] gave a Young's modulus of 69.0 MPa with a standard deviation of 17.9 MPa. The model can predict the Young's modulus of aerogels with similar microstructures.

5.2 DYNAMIC MECHANICAL ANALYSIS

Frequency scans of polyurea aerogel showed that frequency has a minimal effect of the storage modulus in the frequency range tested. The tension test indicated that the 0.17 g/cm^3 more efficiently damped the oscillatory motion induced by the DMA than the other two densities. Testing has concluded that polyurea aerogels are mechanically strong

and insensitive to mid-range frequencies. PUA could prove to be a useful new material in a wide variety of engineering structures if processes could be developed to keep production costs to a minimum.

6. CONCLUSIONS AND FUTURE WORK

Frequency sweeps were conducted on polyurea aerogel (PUA) in bending and tension. A computer model of the PUA microstructure was developed to investigate the micro-properties of PUA such as bond strength and particle stiffness.

6.1 CONCLUSIONS

The presented results indicate that polyurea aerogels are not particularly sensitive to various mid-range frequencies. The change in storage modulus across the tested range was minimal. The 0.17 g/cm^3 damped the oscillatory motion more effectively in tension than the other two densities.

Compression simulations using PFC3D were completed using the PUA model. Simulations showed that the shear and normal stiffness of the particles were equivalent rather than a ratio as suggested by literature.

6.2 FUTURE WORK

Further investigation into extremely high and low frequencies would need to be completed if these materials were to be used in such cases. Additional testing would be needed to investigate the effects of extreme environments (i.e. very hot/cold or moist) as well as the effect of fatigue over the lifetime of a part. Further inquiry into modeling of tension and bending would increase the confidence in parameters obtained through compression models.

APPENDIX A

COMPUTER CODE FOR DLCA ALGORITHM

```
 %Input file for aerogel model creator

clc
clear all
close all

%User inputs
L = input('What edge length is the cube?  ');

Recipe = input('What recipe should be used? ');

particlecalc

particlenum

%Particle maximum movement distance in each direction
maxmove = L * 0.1;
ww = 0;

cubefill

'cube is filled'

ii = 1;
jj = 1;
n = 0;
combined = zeros(particlenum,particlenum);

particlemove2

finalplot

particles

xlswrite('Particles', particles)

%All data can be found in PUA Particle Excel worksheet
if (Recipe == 1.37)

    maxradius = xlsread('PUAParticle', 'G3:G3');
    minradius = xlsread('PUAParticle', 'H3:H3');
    avgradius = xlsread('PUAParticle', 'E3:E3');
    porosity = xlsread('PUAParticle', 'I3:I3') / 100;

elseif (Recipe == 2.75)

    maxradius = xlsread('PUAParticle', 'G4:G4');
    minradius = xlsread('PUAParticle', 'H4:H4');
    avgradius = xlsread('PUAParticle', 'E4:E4');
    porosity = xlsread('PUAParticle', 'I4:I4') / 100;

elseif (Recipe == 5.5)
```

```
    maxradius = xlsread('PUAParticle', 'G5:G5');
    minradius = xlsread('PUAParticle', 'H5:H5');
    avgradius = xlsread('PUAParticle', 'E5:E5');
    porosity = xlsread('PUAParticle', 'I5:I5') / 100;

elseif (Recipe == 11)

    maxradius = xlsread('PUAParticle', 'G6:G6');
    minradius = xlsread('PUAParticle', 'H6:H6');
    avgradius = xlsread('PUAParticle', 'E6:E6');
    porosity = xlsread('PUAParticle', 'I6:I6') / 100;

elseif (Recipe == 16.5)

    maxradius = xlsread('PUAParticle', 'G7:G7');
    minradius = xlsread('PUAParticle', 'H7:H7');
    avgradius = xlsread('PUAParticle', 'E7:E7');
    porosity = xlsread('PUAParticle', 'I7:I7') / 100;

elseif (Recipe == 33)

    maxradius = xlsread('PUAParticle', 'G8:G8');
    minradius = xlsread('PUAParticle', 'H8:H8');
    avgradius = xlsread('PUAParticle', 'E8:E8');
    porosity = xlsread('PUAParticle', 'I8:I8') / 100;

else

    'Recipe not found'

end

CubeVolume = L ^ 3;

AirVolume = CubeVolume * porosity;

ParticleVolume = CubeVolume - AirVolume;

SingleParticle = 4 / 3 * pi() * avgradius ^ 3;

particlenum = round(ParticleVolume / SingleParticle);

%Fills the cube with non-intersecting spheres

particles = zeros(particlenum,5);

max = L;

[x,y,z] = sphere();
hold on

for ii = 1:particlenum
```

```
  if (Recipe >= 16.5)

    %Creates particles within the cube with a given radius for
particulate
    %material
    particles(ii,4) = random('unif',minradius,maxradius);
    particles(ii,1) = random('unif',particles(ii,4),max-
particles(ii,4));
    particles(ii,2) = random('unif',particles(ii,4),max-
particles(ii,4));
    particles(ii,3) = random('unif',particles(ii,4),max-
particles(ii,4));

    jj = 1;
        while (jj < ii)

            intersectfill

            if (value(ii,jj) <= 0)

                particles(ii,4) = random('unif',minradius,maxradius);
                particles(ii,1) = random('unif',particles(ii,4),max-
particles(ii,4));
                particles(ii,2) = random('unif',particles(ii,4),max-
particles(ii,4));
                particles(ii,3) = random('unif',particles(ii,4),max-
particles(ii,4));

                jj = 1;
                intersectfill
            else
                jj = jj+1;
            end

        end


surf(particles(ii,4)*x+particles(ii,1),particles(ii,4)*y+particles(ii,2
),particles(ii,4)*z+particles(ii,3))

  else
    %Creates particles within the cube with a given radius for fibrous
    %material
    particles(ii,4) = minradius;
    particles(ii,5) = maxradius;
    particles(ii,1) = random('unif',particles(ii,4),max-
particles(ii,5));
    particles(ii,2) = random('unif',particles(ii,4),max-
particles(ii,4));
    particles(ii,3) = random('unif',particles(ii,4),max-
particles(ii,4));

    jj = 1;

    while (jj < ii)
```

```
      intersectfill

      if (value(ii,jj) <= 0)

        particles(ii,4) = minradius;
        particles(ii,5) = maxradius;
        particles(ii,1) = random('unif',particles(ii,4),max-
particles(ii,5));
        particles(ii,2) = random('unif',particles(ii,4),max-
particles(ii,4));
        particles(ii,3) = random('unif',particles(ii,4),max-
particles(ii,4));

        jj = 1;
        intersectfill

      else

          jj = jj+1;

      end

    end


surf(particles(ii,5)*x+particles(ii,1),particles(ii,4)*y+particles(ii,2
),particles(ii,4)*z+particles(ii,3))

  end

end

daspect([1 1 1])
view(45,45)

%Checks for intersecting spheres

if (ii ~= jj)

   %Calculates the distance between centers of the spheres

   magnitudeV(ii,jj) = sqrt((particles(ii,1)-particles(jj,1))^2 +
(particles(ii,2)-particles(jj,2))^2 + (particles(ii,3)-
particles(jj,3))^2);

   if (Recipe >= 16.5)
        dist(ii,jj) = particles(ii,4) + particles(jj,4);

   else

       dist(ii,jj) = particles(ii,5) + particles(jj,5);
```

```
   end

   value(ii,jj) = magnitudeV(ii,jj) - dist(ii,jj);
   value(jj,ii) = magnitudeV(ii,jj) - dist(ii,jj);

end

%Diffuses particles in the cube and joins them when they intersect

%Repeats until only once cluster exists

combined = zeros(particlenum, particlenum);
m = particlenum ^ 2 - particlenum;
dipole = zeros(particlenum,3);
max = L * 0.9;
min = L * 0.1;

while (n < m)

    for ii = 1:particlenum

        move = zeros(3,1);
        %Moves particles up to max distance at 5 percent increments
        %unless they would leave the cube

            %x direction
            xintersect = 0;
            if ((particles(ii,1) < 0) || (particles(ii,1) > L))
                move(1,1) = -particles(ii,1);

            else
                move(1,1) = random('unif',-maxmove,maxmove);
            end
            for qq = 1:20

                xnetwork = 0;

                for bb = 1:particlenum
                    if ((particles(bb,1) < min) || (particles(bb,1) >
max))

                        xnetwork = xnetwork + 1;

                    end
                end

                    particles(ii,1) = particles(ii,1) + 0.05 *
move(1,1);

                    if ((particles(ii,1) > max) || (particles(ii,1) <
min))
                        particles(ii,1) = particles(ii,1) - 0.05 *
move(1,1);
                        break
```

```
                    end

                    xnetwork = 0;

                    for bb = 1:particlenum

                        if ((bb ~= ii) && (combined(ii,bb) == 1))

                            particles(bb,1) = (particles(bb,1) + 0.05 *
(move(1,1)));

                            if ((particles(bb,1) > max) ||
(particles(bb,1) < min))

                                xnetwork = 1;
                            end

                        end
                    end


                for jj = 1:particlenum
                    intersect

                    if (value(ii,jj) <= 0)

                        xintersect = 1;

                    end
                end

                if (xnetwork > 0)

                    break
                end

                if ((particles(ii,1) > max) || (particles(ii,1) < min)
|| (xintersect == 1) || (xnetwork ~= 0))

                    break

                end

            end

            %y direction
            yintersect = 0;
            if ((particles(ii,2) < 0) || (particles(ii,2) > L))
                move(2,1) = -particles(ii,2);
            else
                move(2,1) = random('unif',-maxmove,maxmove);
            end
```

```
            for qq = 1:20

                ynetwork = 0;

                for bb = 1:particlenum

                    if ((particles(bb,2) < min) || (particles(bb,2) >
max))

                        ynetwork = 1;

                    end
                end

                    particles(ii,2) = particles(ii,2) + 0.05 *
move(2,1);

                    if ((particles(ii,2) > max) || (particles(ii,2) <
min))
                        particles(ii,2) = particles(ii,2) - 0.05 *
move(2,1);
                        break
                    end

                    ynetwork = 0;

                    for bb = 1:particlenum

                        if ((bb ~= ii) && (combined(ii,bb) == 1))

                            particles(bb,2) = (particles(bb,2) + 0.05 *
(move(2,1)));

                            if ((particles(bb,2) > max) ||
(particles(bb,2) < min))

                                ynetwork = ynetwork + 1;
                            end

                        end
                    end



                for jj = 1:particlenum
                    intersect

                    if (value(ii,jj) <= 0)

                        yintersect = 1;

                    end
                end
```

```
                if (ynetwork > 0)
                    break
                end

                if ((particles(ii,2) > max) || (particles(ii,2) < min)
|| (yintersect == 1) || (ynetwork ~= 0))

                    break

                end

            end

            %z direction
            zintersect = 0;
            move(3,1) = random('unif',-maxmove,maxmove);
            for qq = 1:20

                if ((particles(ii,3) < 0) || (particles(ii,3) > L))
                    move(3,1) = -particles(ii,3);
                else
                    move(3,1) = random('unif',-maxmove,maxmove);
                end


                    if ((particles(ii,3) > max) || (particles(ii,3) <
min))

                        particles(ii,3) = particles(ii,3) - 0.05 *
move(3,1);
                        break

                    end

                    znetwork = 0;

                    for bb = 1:particlenum

                        if ((bb ~= ii) && (combined(ii,bb) == 1))

                            particles(bb,3) = (particles(bb,3) + 0.05 *
(move(3,1)));

                            if ((particles(bb,3) > max) ||
(particles(bb,3) < min))

                                znetwork = 1;
                            end

                        end
                    end
```

```
                for jj = 1:particlenum
                    intersect

                    if (value(ii,jj) <= 0)
                        zintersect = 1;
                    end

                end

                if (znetwork > 0)
                    break
                end

                if ((particles(ii,3) > max) || (particles(ii,3) < min)
|| (zintersect == 1) || (znetwork ~= 0))

                    break

                end

            end

            theta = random('unif', 0, 2*pi);
            thetaz = random('unif', 0, pi);

            dipole(ii,1) = particles(ii,1) + particles(ii,4) *
cos(theta) * cos(thetaz);
            dipole(ii,2) = particles(ii,2) + particles(ii,4) *
sin(theta) * cos(thetaz);
            dipole(ii,3) = particles(ii,3) + particles(ii,4) *
sin(thetaz);

        join2


    end

    finish

end

%Checks for intersecting spheres

if (ii ~= jj)

   %Calculates the distance between centers of the spheres

   magnitudeV(ii,jj) = sqrt((particles(ii,1)-particles(jj,1))^2 +
(particles(ii,2)-particles(jj,2))^2 + (particles(ii,3)-
particles(jj,3))^2);

   dist(ii,jj) = particles(ii,4) + particles(jj,4);
```

```
   value(ii,jj) = magnitudeV(ii,jj) - dist(ii,jj);
   value(jj,ii) = magnitudeV(ii,jj) - dist(ii,jj);

end

%Moves connected particles together


for jj = 1:particlenum

    if ((ii ~= jj) && (combined(ii,jj) == 0))

        intersect

        %Finds the angle between the dipoles of the two particles
        vector1(1) = dipole(ii,1) - particles(ii,1);
        vector1(2) = dipole(ii,2) - particles(ii,2);
        vector1(3) = dipole(ii,3) - particles(ii,3);
        vector2(1) = dipole(jj,1) - particles(jj,1);
        vector2(2) = dipole(jj,2) - particles(jj,2);
        vector2(3) = dipole(jj,3) - particles(jj,3);

        magnitude1 = sqrt(vector1(1)^2 + vector1(2)^2 + vector1(3)^2);
        magnitude2 = sqrt(vector2(1)^2 + vector2(2)^2 + vector2(3)^2);

        dotproduct =  vector1(1) * vector2(1) + vector1(2) * vector2(2)
+ vector1(3) * vector2(3);

        vectortheta = acos(dotproduct / (magnitude1 * magnitude2));

        if (vectortheta <= pi/4)
            stick = 0;

        elseif ((vectortheta > pi/4) && (vectortheta <= pi/2))
            stick = 33;

        elseif ((vectortheta > pi/2) && (vectortheta <= 3*pi/4))
            stick = 66;

        else
            stick = 100;
        end

        number = random('unif', 0, 100);

    %If intersection occurs the two particles are linked
        if ((value(ii,jj) <= 0) && (number <= stick) && combined(ii,jj)
== 0);

            combined(ii,jj) = 1;
            combined(jj,ii) = 1;
```

```
    %Adds any other connected particles in the existing cluster

            for aa = 1:particlenum

                if ((combined(ii,aa) == 1) && (ii ~= aa) && (jj ~= aa)
&& (combined(jj,aa) == 0))

                    combined(jj,aa) = combined(ii,aa);
                    combined(aa,jj) = combined(ii,aa);

                end

                if ((combined(aa,jj) == 1) && (ii ~= aa) && (jj ~= aa)
&& (combined(ii,aa) == 0))

                    combined(ii,aa) = combined(aa,jj);
                    combined(aa,ii) = combined(aa,jj);

                end
            end
        end
    end
end

%Checks for the completion of the model

n = 0;

for ii = 1:particlenum
    join2
end

for aa = 1:particlenum
    for bb = 1:particlenum

        if (combined(aa,bb) == 1)

            n = n + 1;

        end
    end
end
percent = n / m * 100

%Plots the final figure

figure (2)
[x,y,z] = sphere;
hold on

for ii = 1:particlenum
```

```
surf(particles(ii,4)*x+particles(ii,1),particles(ii,4)*y+particles(ii,2
),particles(ii,4)*z+particles(ii,3))

end
daspect([1 1 1])
view(45,45)
```

APPENDIX B.

RAW TEST DATA

Table B.1. Average Raw Data for 0.12 g/cm^3 in Three Point Bend.

0.12 g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	48.44837
1.3	48.72176
2.1	48.90028
2.9	49.05232
3.8	49.11454
4.6	49.28909
5.4	49.41943
6.2	49.56111
7	49.81604
7.8	50.2405
8.7	50.39384
9.5	50.53472
10.2	50.52974
11	50.32502
11.8	50.67738
12.6	50.92742
13.6	51.18759
14.4	51.31713
15.2	51.43152
16	51.67582

Table B.2. Average Test Data for 0.17 g/cm^3 in Three Point Bend.

0.17g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	73.241185
1.3	73.59673
2.1	73.628005
2.9	73.853825
3.8	74.04009
4.6	74.21713
5.4	74.394585
6.2	74.50198
7	74.69944
7.8	74.905385
8.7	75.263435
9.5	75.752975
10.2	75.9501
11	76.18228
11.8	76.266395
12.6	76.50357
13.6	76.760755
14.4	76.852015
15.2	77.09388
16	77.284115

Table B.3. Average Raw Data for 0.33 g/cm^3 in Three Point Bend.

0.33 g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	163.5795
1.35	161.9324
2.23	162.717175
3.13	162.717175
4.07	162.717175
4.97	162.5263667
5.83	162.5263667

Table B.4. Average Raw Data for 0.12 g/cm^3 in Tension.

0.12 g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	40.95721
1.5	41.51524
2.6	41.81616333
3.6	42.00776333
4.6	42.15958667
5.7	42.28862
6.7	42.38877
7.7	42.48212333

Table B.5. Average Raw Data for 0.17 g/cm^3 in Tension.

0.17 g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	25.3338725
1.5	25.727455
2.6	25.9634525
3.6	26.1178075
4.6	26.2347825
5.7	26.341305
6.7	26.42043
7.7	25.73192
8.8	25.799685
9.8	25.86977
10.8	25.916335
11.8	25.94732
12.8	25.98857

Table B.6. Average Raw Data for 0.17 g/cm^3 in Tension.

0.33 g/cm^3	
Frequency (Hz)	Storage Modulus (MPa)
0.5	138.9143
1.5	140.3805
2.6	141.152625
3.6	141.578025
4.6	141.911725
5.7	142.191675
6.7	142.5984667
7.7	142.7957333
8.8	142.9042667

BIBLIOGRAPHY

1. Chakkaravarthy, C., Larimore, Z., Sotiriou-Leventis, C., Mang, J., & Leventis, N., *One-Step room-temperature synthesis of fibrous polyimide aerogels from anhydrides and isocyanates and conversion to isomorphic carbons.* Journal of Materials Chemistry, 2010(20): p. 9666-9678.

2. Leventis, N., Sotiriou-Leventis, C., Chandrasekaran, N., Mulik, S., Larimore, Z., Lu, H.,Churu, G.,Mang, T. , *Multifunctional Polyurea Aerogels from Isocyanates and Water. A Sturcture-Property Case Study.* Chemistry of Materials, 2010(22): p. 6692-6710.

3. Kistler, S., *Coherent Expanded Aerogels.* Journal of Physical Chemistry, 1932(36): p. 52-64.

4. Zhang, G., Dass, A., Rawashdeh, A., Thomas, J., Councsil, J., Sotiriou-Leventis, C., Fabrizio, E., Ilhan, F., Vassilaras, P., Scheiman, D., McCorkle, L., Palczer, A., Johnstons, J., Meador, M., Leventis, N., *Isocyanate-Crosslinked Silica Aerogel Monoliths: Preparation and Characterization.* Journal of Non-Crystalline Solids, 2004. **350**: p. 152-164.

5. Leventis, N., Palczer, A., McCorkle, L., *Nanoengineered Silica-Polymer Composite Aerogels with No Need for Supercritical Fluid Drying.* Journal of Sol-Gel Science and Technology, 2005. **35**: p. 99-105.

6. Lee, J., Gould, G., Rhine, W., *Polyurea Based Aerogels for a High Performance Thermal Insulation Material.* Journal of Sol-Gel Science and Technology, 2009. **49**: p. 209-220.

7. Roy, S., Shimpi, N., Katti, A., Lu, H., Rahman, M., *Mechanical Characterization and Modeling of Isocyanate-Crosslinked Nanostructured Silica Aerogels*, in *47th AIAA/ASM/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. 2006.

8. Reinheimer, P.G., *Construction of a Multi-Functional Cryogenic Propellant Tank with Cross Linked Silica Aerogels*. 2010, University of Alabama, Tuscaloosa.

9. Loebs, J., *Processing and Mechanical Characterization of Polyurea Aerogels*. 2011, Missouri University of Science and Technology.

10. Ma, H., Roberts, A., Prevost, J., Jullien, R., Scherer, G., *Mechanical Structure-Property Relationship of Aerogels.* Journal of Non-Crystalline Solids, 2000(277): p. 127-141.

11. Shimpi, N., *Mechanical Testing and Numerical Simulation of Structure-Property Relationship of Silica Aerogel*. 2005, Oklahoma State University.

12. Jullien, R., Hasmy, A., *Fluctuating Bond Aggregation: A Model for Chemical Gel Formation.* Physical Review Letters, 1995. **74**(20): p. 4003-4006.

13. Menard, K., *Dynamic Mechanical Analysis*. 1999: CRC Press.

14. Katti, A.R., *Characterization of the Thermo-Mechanical Behavior of Crosslinked Silica Aerogel*. 2001, Oklahoma State University.

15. Hawick, K.A. *Simulating and Visualising Sedimentary Cluster-Cluster Aggregation*. Computational Science Technical Note 2010 [cited 2011 October 19]; Available from: http://www.massey.ac.nz/~kahawick/cstn/.

16. Itasca, *Particle Flow Code in 3 Dimensions, Version 4.0, User's Manual*. 2010, Itasca Consulting Group, Inc.

17. Meakin, P., Family, F., *Structure and Dynamics of Reaction-Limited Aggregation.* Physical Review Letters, 1987(36): p. 5498-5501.

18. A.S.T.M., *Standard Test Method for Plastics: Dynamic Mechanical Properties: In Tension*. 2006.

19. A.S.T.M., *Standard Test Method for Plastics: Dynamic Mechanical Properties: In Flexure (Three-Point Bend)*. 2007.

VITA

Victoria Ann Prokopf was born to parents James and Barbara Prokopf. Victoria graduated from Collinsville High School in May of 2007. In the fall of the same year Victoria was admitted to the University of Missouri – Rolla. Victoria earned her Bachelor of Science degree in Aerospace Engineering with a minor in mathematics from the renamed Missouri University of Science and Technology. After receiving a Bachelor's degree Victoria started her Master of Science degree in Aerospace Engineering in the Summer of 2011. While pursuing her degree Victoria has worked as a Graduate Research Assistant in Leventis Labs and a Graduate Teaching Assistant for IDE 20. Victoria will receive her Master of Science degree in Spring of 2013.