UNIVERSITY OF TWENTE.

Faculty of Science and Technology

Determining the Viscoelastic Response of Trabecular Bone

in collaboration with

Radboudumc
university medical center

Internship Report

Supervised By:

Dr. Jai Prakash
Professor
Engineered Therapeutics
University of Twente
j.prakash@utwente.nl

Dr. Dennis Janssen
Assistant Professor
Orthopedic Research Laboratory
Radboud UMC
dennis.janssen@radboudumc.nl

Thomas Gersie
PhD Candidate
Orthopedic Research Laboratory
Radboud UMC
gersie.thomas@radboudumc.nl

Submitted By:

Sainivedhitha Arunajatesan
(s2662841)
Masters, Biomedical Engineering
s.arunajatesan@student.utwente.nl

Duration of Internship:
28-Feb-2022 to 10-June-2022

Abstract

The time independent mechanical properties of trabecular bone have been extensively studied including the relation between bone mineral density and Young's modulus. The time dependent properties (or) viscoelasticity of bone have not gained much attention. The purpose of this report is to determine the stress response and relate it to the bone mineral density and strain. The bone mineral density was calculated for human samples while the stress response and further correlations were made in bovine samples due to time constraints. Human distal femoral and proximal tibial samples were collected from 6 donors and the bone mineral density were determined. Simultaneously, stress relaxation experiments with a holding time of 30 minutes were performed on 16 cylindrical bovine femoral samples. The minimum duration data of the reference samples were fitted using the Schapery method and the Modified Superposition Method and validated with other samples. Among these two methods, Modified Superposition Method showed a better fit (error $\approx$ 46.43%) than Schapery Method (error $\approx$ 66.82%). The parameters of these methods were used to determine if there is a correlation between bone mineral density and strain together with stress response. While comparing the parameters of the Schapery and Modified Superposition Method equations with the bone mineral density, there was no correlation between the parameters of Modified Superposition Method and the non-linear parameters of the Schapery method meaning that the equation of Modified Superposition Method cannot be represented in terms of bone mineral density. On the other hand, the linear parameters of Schapery method related directly to the bone mineral density of the samples. Even if Modified Superposition Method essentially had a fairly better fit for the stress relaxation data, it has no parameters in its equation that relates to bone mineral density. Therefore, it was concluded that bone mineral density can be related to strain and stress with linear parameters of the Schapery model. The Schapery formula with bone mineral density, strain and stress had an error of 88.2%. Due to a high error rate, the correlation between bone mineral density and strain coupled with stress response using Schapery model was not ideal.

Contents

List of Figures

List of Tables

1 Introduction

The trabecular bone is an open porous composite cellular solid material from an engineering perspective. The vertebral bodies of the axial skeleton and the ends of long bones of appendicular skeleton include trabecular bone. It is also present in the hollow long bones near the tip of the medullary chambers. The skeleton as a whole benefits from the intricate, porous spatial organization of the trabecular bone which helps to maximize strength [1]. A large substrate is available for cellular interaction with bone mineral material provided by the high mineral surface area associated with the arrangement of the trabecular bone components. It creates a trabecular network of connecting rod- and plate-shaped structures that varies in density depending on the skeletal location.The proportions of the trabecular elements and their orientation in space, as well as trabecular architecture are all very diverse [2]. The mechanical properties of this cellular material depend on its heterogeneous microstructure, which varies with age, disease and anatomical site being considered [3]. The time-independent mechanical properties of the trabecular bone such as Young's modulus is generally related to its density and computer models of bone typically exploit these relationships. The density of the trabecular bone is usually determined with the help of Computed Tomography (CT) images [4, 5]. Studies have found that the trabecular bone has time-dependent mechanical properties [6].

Viscoelasticity is the time-dependent elastic behavior of materials. Viscoelastic behavior combines elastic and viscous behavior, with an immediate elastic strain followed by a viscous, time-dependent strain as a function of applied stress. Therefore, the response to an external stimulus is delayed, which results in a loss of energy inside the material [7]. There are certain experiments performed to determine the viscoelastic behavior of trabecular bones. Tensile tests [8, 9] test the force required to break the specimen and the extent to which the specimen elongates to that breaking point. Stress relaxation tests [5, 9, 10, 11] measures time-dependent varying force due to applied constant strain over time. Creep recovery tests [4, 12] measures the time-varying strain due to applied constant load over time. Dynamic Mechanical Analysis [13] measures the lag between sinusoidal stress and strain over a frequency range. Fatigue tests [10] determines the variation in Young's modulus at a constant strain rate and different cyclic loads. Various studies which determined viscoelastic behavior of the bones claim that the behavior linear [10, 12]. But in reality, the viscoelasticity is not linear and is dependent on the applied stress or strain level [11]. So, a material model cannot be developed with the assumption of linear viscoelasticity. At this point, the study on the non-linear behavior of trabecular bone comes into account.

A study by Manda.et.al (2016) attempted to explain the non-linear viscoelastic properties of the bovine bone sample using Multiple Load Creep Recovery tests [4] but crucially fails to explain the effect of bone mineral density (BMD). Therefore, it is known that the effect of strain on stress response is non-linear. But the effect of BMD on stress response is still unclear. To determine this relationship, an existing model which relates strain to stress should be used to fit the stress relaxation data.

There are several models that were used to fit the stress relaxation data. The comparison of Quasi-Linear Viscoelasticity (QLV), Nonlinear Superposition and Schapery model on tendons and ligaments concluded that Schapery's nonlinear viscoelastic model correctly predicts recovery and reloading behavior while fitting a single relaxation curve and predicts the strain dependent relaxation behavior as well [14]. Many studies have used Schapery model to fit the stress relaxation data of polymers [15, 16, 17] and a few studies have used Modified Superposition Method to fit their stress relaxation data of ligaments and human dentin [18, 19, 20]. Provenzano et. al., decided to determine whether the Schapery theory or the modified superposition method could adequately model the strain-dependent stress-relaxation behavior of ligaments and found that both works the same [21]. So, it was decided to compare the Schapery model and Modified Superposition Method to fit the stress relaxation data. Both the Schapery method and Modified Superposition Method directly relates stress to applied stain level using Young's modulus of Tangent modulus. So, the variables of the models related to BMD were replaced with the relationship equations.

The aim of this study is to perform stress relaxation experiments on each sample for 24 hours, fit the results of a minimum duration to two existing models and determine if these models could be used to simulate nonlinear (or strain dependent) stress relaxation of bovine trabecular bone and if these can relate the bone mineral density to this stress either linearly or non-linearly.

The remainder of the report is organised as follows: First, the scenario that was presented before starting the internship was described in Section 2, followed by the problem statement in Section 3. Section 4 is about the planning for the entire assignment followed by methodology and results in Sections 5 and 6. The discussion and conclusion of the assignment is described in Sections 7 and 8 respectively.

2 Background

The Orthopedics Research Laboratory (ORL) at the Radboud University Medical Centre constructed an Finite Element (FE) model to reduce the chance of failure of a total knee implant. This model is able to predict the primary fixation of a prosthesis by calculating the relative displacement at the bone-implant interface, also known as micromotions. However, it was shown that this FE model underestimated the micromotions in comparison to experimental measurements [22]. It is believed to be due to the absence of viscoelastic properties in the FE model. The solution is that a model must incorporate the viscoelastic behavior. So, the viscoelastic behavior of the trabecular bone from the femur and tibia need to be described numerically to make a model. Previous studies relate the viscoelastic behavior to strain level but not to BMD. The goal of this internship is to perform the material assignments on human bone sample to determine bone mineral density and characterize the viscoelastic behavior of bone specimens, through stress relaxation tests performed in the trabecular bone samples and to correlate the stress response to the bone mineral density and the initial strain levels. Due to time constraints, in this internship, the material assignments were carried out on human bone sample and stress relaxation tests were carried out on bovine femur samples.

The following section describes the problem statement presented throughout this internship report and the goals that were defined to address problem statement.

3 Problem Statement and goals

The primary objective of the study was **to identify the correlation between the stress relaxation response and the initial strain level coupled with bone mineral density in bovine femoral trabecular bone.**

To achieve this the following question need to be answered:

1. What is the BMD of each sample?
2. Which of the MSM or Schapery predict the viscoelastic behavior better?
3. How does BMD influence the viscoelastic behavior?
4. How does either MSM or Schapery incorporated with BMD predict the viscoelastic behavior?

The subsequent sections describe how the activities during the course of the internship were planned and the procedure in which the experiments were carried out in order to achieve the results towards the aforementioned problem statement.

4 Planning

The plan for the total duration of 14 weeks during the internship was scheduled as shown in Figure 1.

As stated before, due to time constraints, the entire assignment was carried out under two phases:

- **Phase I (Human Samples)**: Sample collection, Material assignments

- **Phase II (Bovine Samples)**: Stress Relaxation Experiments, Data Fitting

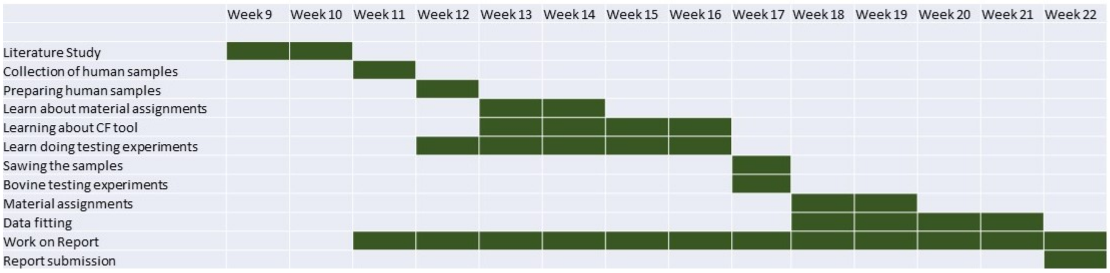


Figure 1: Internship Planning

5 Methodology

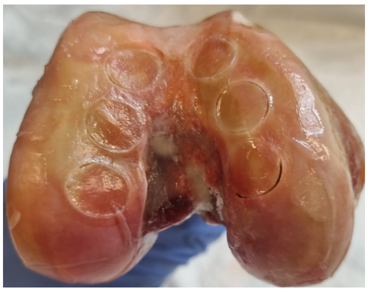
5.1 Sample Preparation

Human Femur and tibia were collected from 6 donors - 1 male and 5 female. The donors were classified into two groups based on age - 3 with mean age of 54 (53,57,53) and 3 with mean age of 80 (90,73,76). Cylindrical samples were harvested from the distal end of the femur and the proximal tibia as they form the knee joint. A 10 mm inner diameter cylindrical saw was used to acquire the trabecular bone core each of 30mm in length. A minimum of six samples were drilled from each tibia parallel to the shaft. In the femur, two samples at 80° from posterior condyles and six samples parallel to the femoral shaft. The femur and tibia after drilling as shown in Figure 2 were wrapped with saline cloth and stored at $-20°C$.

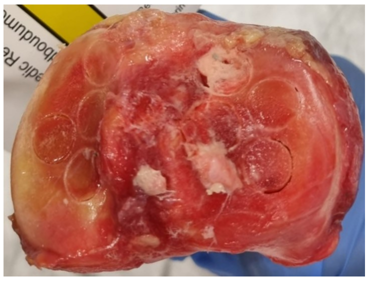
After a few days, all the 12 drilled samples were scanned under a high resolution CT (Siemens Biograph mCT) at the Radboud UMC. These CT images (Figure 3) were used to determine bone mineral density as described in Section 5.2. The excess bones other than the sample were sawn off. The femur and tibia without the samples is shown is Figure 4. The samples with a size of around 21mm were selected for further procedures to get an aspect ratio of 1:2.

The bones were sliced from the selected samples to remove all the cortical bone on top or marrow on the bottom, wrapped with saline cloth and stored at $-20°C$. The sample height, diameter and the height of the sliced portion from the top were measured using a digital caliper and noted.

(a) Femur in the drill



(b) Femur after drilling



(c) Tibia after drilling

Figure 2: Sample drilling

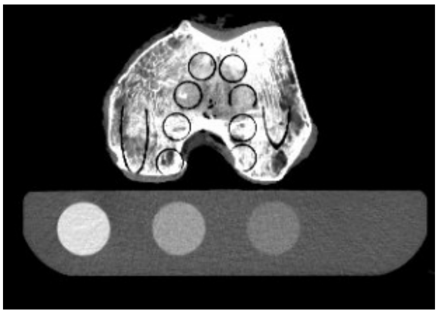
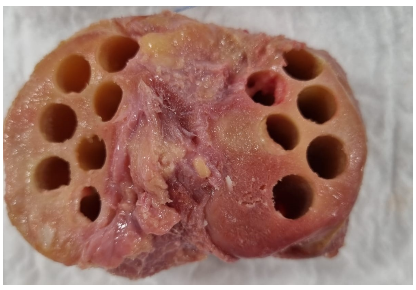


Figure 3: High Resolution CT image

(a) Femur after sawing



(b) Tibia after sawing

Figure 4: Sample sawing

The successive section explains the methods with which the further procedures were performed on the human sample and the bovine sample.

5.2 Material Assignments

The first step of the material assignment was to illustrate all the data (such as the sample height after sawing, the height of the sample cut from the top, diameter of each sample) obtained during the sample collection. A cylindrical Standard Triangle Language (.stl) file was created for each sample with the diameter same as the sample using Tinkercad. The STL file and the CT scans were loaded in a software called 3D Slicer (Slicer 4.11.20210226) and the cylinders were transformed to the locations that matches the exact projection of each sample. The height of the sample cut from the top was also adjusted while positioning each cylinder. Then the diameter of the cylinders were slightly adjusted so that it fits perfectly in the CT scan. These transformed cylinders were hardened to make sure it's position remains unchanged. The hardened cylinders as stl files were loaded in Hypermesh (HyperMesh Version : 2021.2.1 - HWDesktop) in which the 2D cylinders were filled with mesh and made 3D. The mesh created using Hypermesh is shown in Figure 5. The outputs of Hypermesh were saved as job files of the latest version. But our in-house software accepts a job file of version 2017.1. So, Marc Mentat (Marc Mentat 2021.4) is used to convert the job file of current version to job file of 2017.1 version and makes it suitable for use in Matlab. Finally, a custom written Matlab script (Matlab R2021b, Mathworks, MA, USA) with the job (.dat) file and the DICOM images (CT scans) of the sample as input was used to determine the BMD of the human trabecular bone of the femur and the tibia [23].

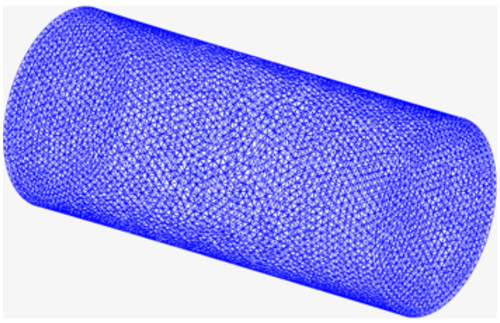


Figure 5: A sample of Hypermesh output

5.3 Stress Relaxation Tests

For this assignment, stress relaxation tests were done in the bovine femur samples. There were 16 bovine samples in total and on each of the samples 24 hours stress relaxation experiments were performed. As the first step, the sample to be tested was cemented to the end caps using Autoplast Cold Curing Denture Base Material. The end caps were used to ensure that the bone does not get damaged during the application of stress. The sample after cementing is shown in Figure 6.

Figure 6: Bovine sample after cementing

Prior to pre-conditioning, the sample embedded in the endcaps was placed in a water basin filled with physiological saline, with a temperature of $37.0 \pm 0.5\ °C$ for 30 minutes. Then, each sample was preconditioned by applying 0.1% apparent strain for ten cycles [24] and was allowed to recover for 30 minutes again. The experiments were conducted using an MTS machine (MTS Systems Corporation, Eden Priairie, Minnesota, USA) with a measuring frequency of 10 Hz. The entire experimental setup is shown in Figure 7. It is known that trabecular exhibits nonlinearity and residual strain buildup below the yield strain of 0.8 % [3, 25]. In order to quantify the viscoelastic response, static stresses of 0.2, 0.4, 0.6 and 0.8% were used. With a strain rate of 0.01 s^{-1}, the stresses were initially applied and then re-moved. Each strain level investigated four cylindrical trabecular bone samples, yielding a total of 16 specimens for the three studies. A holding period of 24 hours was applied to the static strain in order to measure the length of the stress relaxation response.

Digital imaging contrast (DIC) was used to measure the axial displacement applied on the bone samples. The dots in the end caps, shown in Figure 6, were marked to visualize if any displacement has occurred during the experiment. Images of the uniaxial compression test were continuously captured and deformations of the samples were calculated based on a custom-written Matlab script.

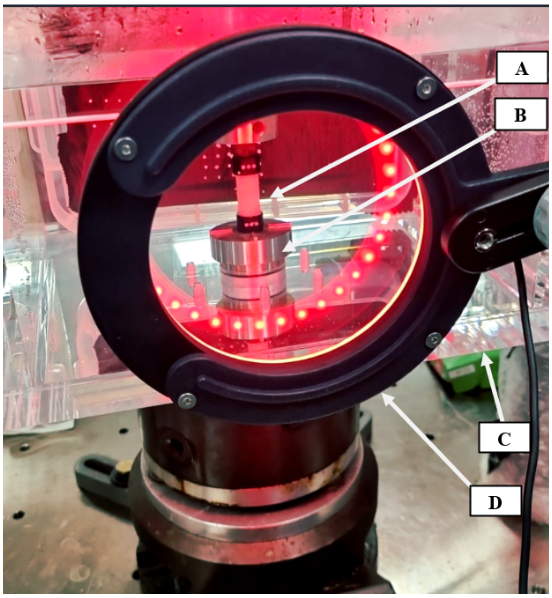


Figure 7: Sample inside MTS during experiment (A)Bovine femur sample; (B)MTS; (C)water bath with saline solution; (D)DIC

5.4 Data fitting

5.4.1 Force Versus Time

The fitting of stress relaxation data with the Schapery and MSM models was done using Matlab. The output text files of the 24 hours experiments from the MTS were taken as the input along with the images obtained using DIC. The DIC images were used to determine the applied strain (ϵ) on each sample. The data such as force, time and axial displacement were read from the MTS files. The data for a minimum duration of 30 minutes was chosen for further findings and illustrations. The force and time shown in Figure 8 were used to determine the time and the position of the force between the region of interest (i.e., between minimum force and maximum force).

Using the diameter of the sample which is determined during sample preparation and the strain applied on each sample which is determined using the DIC images, the stress (σ) and the stress relaxation modulus (E) for each sample were calculated using the equations 1 and 2 where Area is the area of the circle with the diameter of the sample.

$$\sigma = \frac{Force}{Area} \tag{1}$$

$$E = \frac{\sigma}{\epsilon} \tag{2}$$

The force position and time between the region of interest for the first 30 minutes of the experiment was determined. The respective σ(t) and E(t) for each sample was

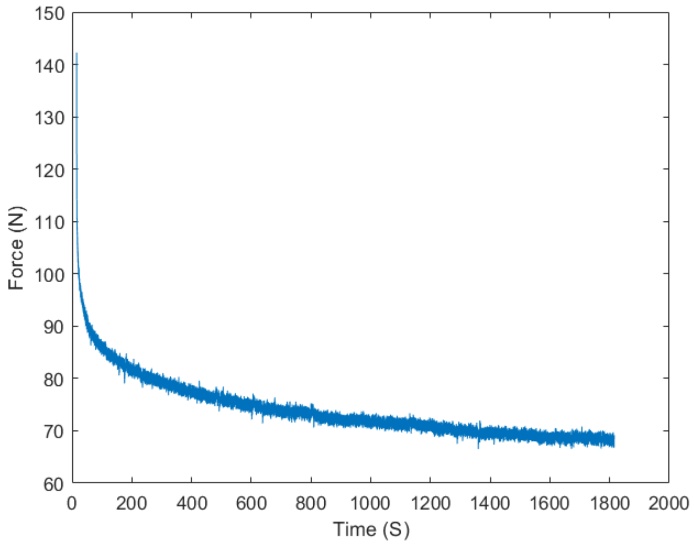


Figure 8: Force Versus Time graph for one bovine sample

calculated. The E(t) Vs time of this experimental data was plotted and fitted with power law to verify as shown in Figure 9. The power law is that a relative change in one quantity results in a proportional relative change in another and the equation of power law looks like $y = ax^b$.

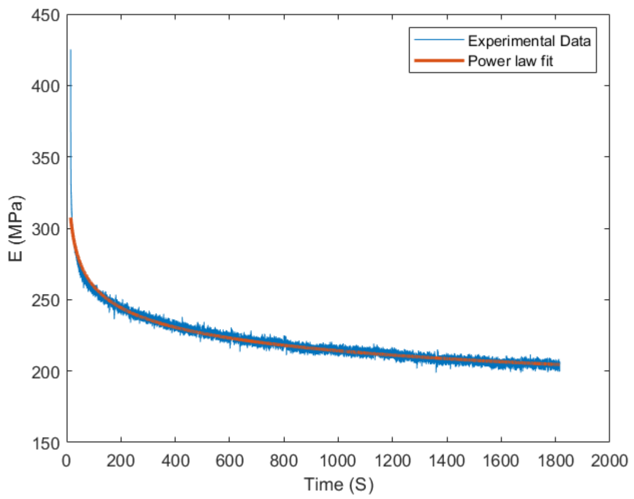


Figure 9: E Versus Time graph with fit for one bovine sample

5.4.2 Models for data fitting

As stated in the section 1, the stress relaxation data was fitted to the Schapery model and MSM model to determine which method has a better fit. Another reason for carrying out this comparison was to identify the correlation of BMD and stress response which will be discussed later in section 7. A detailed explanation on these methods is given in Sections 5.4.3 and 5.4.4.

5.4.3 Schapery Method

Schapery's nonlinear viscoelastic theory can be derived using principles of irreversible thermodynamics [16, 26, 27]. When strain is treated as the independent state variable and the case of uniaxial loading is considered, Schapery's theory reduces to a single integral expression as given in Equation 3.

$$\sigma(\epsilon,t) = h_e(\epsilon)E_e\epsilon + h_1(\epsilon)C\int_0^t \Delta(E)(\rho(t)-\rho^{'}(\tau))\frac{dh_2(\epsilon)\epsilon}{d\tau}d\tau \tag{3}$$

In this formula E_e is the equilibrium modulus which was taken as final value of the young's modulus (E) and $\Delta(E)$ is called the transient modulus. The reduced time ρ and $\rho^{'}$ is defined as

$$\rho = \int_0^t \frac{dt^{'}}{a_e(\epsilon(t^{'})}d\tau \tag{4}$$

$$\rho = \int_0^\tau \frac{dt^{'}}{a_e(\epsilon(t^{'})}d\tau \tag{5}$$

The terms h_e, h_1, h_2, and a_e are strain-dependent material properties. Theoretically, all these values were assumed to be 1 for a linear behaviour which would reduce the equation to a Boltzmann equation [21]. In reality, bovine trabecular bone is non-linear [4]. If the experiments were performed under isothermal conditions, the values of h1 and a_e are assumed to be 1. It is applicable for our data as we have performed our experiments under constant room temperature and the transient modulus is equated to power law as in Equation 6.

$$\Delta E(\rho) = C\rho^n \tag{6}$$

Substituting all these terms to Equation 3, the Schapery equation becomes,

$$\sigma(\epsilon,t) = h_e(\epsilon)E_e\epsilon + h_1(\epsilon)C\int_0^t (\rho-\rho^{'})^n\frac{dh_2(\epsilon)\epsilon}{d\tau}d\tau \tag{7}$$

Applying Heaviside step function, the final equation of Schapery model is

$$\sigma(\epsilon,t) = h_e(\epsilon)E_e\epsilon_0 + h_2(\epsilon)C\epsilon_0 t^n \tag{8}$$

In the above equation, h_e, h_2, E_e can be found using curve fitting and ϵ_0 is the initial strain level from the experimental data. The curve fitting will be performed by fitting the above equation to the graph of E Vs time.

The general equations of h_2 and h_e can be written as a first order polynomials as Equation 9 and Equation 10 respectively.

$$h_2(\epsilon) = L * \epsilon + M \tag{9}$$

$$h_e(\epsilon) = O * \epsilon + P \tag{10}$$

The final equation of the Schapery fit with the functions can be written as Equation 11.

$$\sigma(\epsilon, t) = (O * \epsilon + P) * E_e * \epsilon + (L * \epsilon + M) * C * \epsilon * t^n \tag{11}$$

5.4.4 Modified Superposition Method (MSM)

The single integral formula of MSM is given by Equation 12

$$\sigma(\epsilon, t) = \int_0^t E(t - \tau, \epsilon(\tau)) \frac{d\epsilon(\tau)}{d\tau} d\tau \tag{12}$$

The form of the relaxation function will be chosen as a non-separable strain-dependent power law:

$$E(\epsilon, t) = A(\epsilon) t^{B(\epsilon)} \tag{13}$$

Applying Heaviside step function, the final equation of Schapery model is

$$\sigma(\epsilon, t) = A(\epsilon) \epsilon_0 t^{B(\epsilon)} \tag{14}$$

The general equations of B(ϵ) and A(ϵ) can be expressed as first order polynomials as Equations 15 and 16.

$$B(\epsilon) = D * \epsilon + F; \tag{15}$$

$$A(\epsilon) = H * \epsilon + J; \tag{16}$$

By substituting these two equations in Equation 14, the final equation of MSM becomes,

$$\sigma(\epsilon, t) = (H\epsilon + J) * \epsilon * t^{(D\epsilon + F)} \tag{17}$$

6 Parameter Estimation and Results

6.1 Schapery Method

In order to get a correlation between BMD and stress response, and a random distribution of BMDs over the strain levels, the samples were classified based on their BMD into 4 groups with 4 samples in each group in the order of increasing BMDs (i.e., lowest BMDs in group 1 and highest BMDs in group 4). All the 4 samples in each BMD group had different strain levels and different BMDs. Each group was assumed to have a common reference BMD (average of all 4 BMDs), say BMD_r. The Stress relaxation modulus (E) Versus Time graphs for each BMD group were plotted with the data obtained using the MTS output and DIC images. The Curve Fitter toolbox in Matlab was used to identify the parameters of the Schapery formula (Equation 8) as described below.

The lowest strains of each BMD group (0.2%, 0.19%, 0.18% and 0.16%) were taken as the initial strain levels (ϵ_0) for fitting. The bovine trabecular bone was assumed to behave linearly at these initial strain levels. In other words, the E vs time curve of these values were assumed to respond in a linear way. The curve of E vs time for the strain of 0.2% (initial strain of BMD group 1) was plotted in the curve fitter with the Equation 8 as fit. The values of h_e and h_2 were assumed to be 1 based on previous work by Provenzano et al., as given in the Section 5.4.3 and the values of E_e, C and n were determined from the Curve fitter. Fitting Equation 8 using the above values of E_e, C and n to the stress relaxation data of the other strain levels in BMD group 1, h_e and h_2 for other samples were calculated.

The h-strain curves of each BMD groups were fitted to the Equation 9 and Equation 10 with weights as shown in the Figure 10 to determine the variables of those equations. The weights were applied to the first values (values of initial strain level) to make it approximately 1.

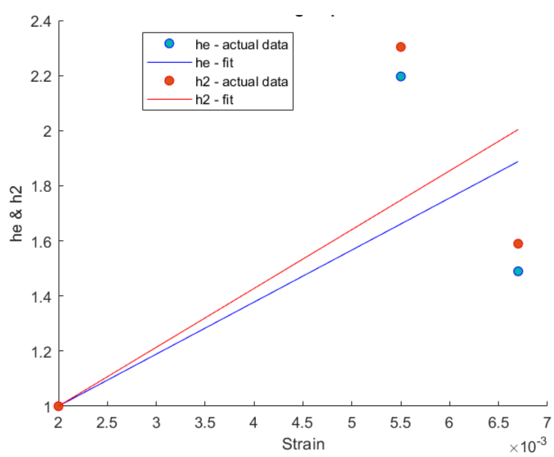


Figure 10: h_e and h_2 of BMD group 1 as a function of strain

By substituting the known values in Equation 11, the final equation of BMD group 1 can be obtained.

The same procedure was followed for the other BMD groups and the respective final equations were formed. The values of Ee, C and n of each BMD group are listed in table 1 and the figures 11, 12 and 13 show the plots of h_e & h_2 for BMD groups 2, 3 and 4 respectively. The values of the variables in equations 9 & 10 can be found in the table 2.

BMD Group	Ee (MPa)	C	n
1	1215	-913.2	0.01439
2	958.4	-547.9	0.0304
3	1729	-384.5	0.08508
4	636.1	-149.3	0.1123

Table 1: Value of Ee, C & n of each BMD group

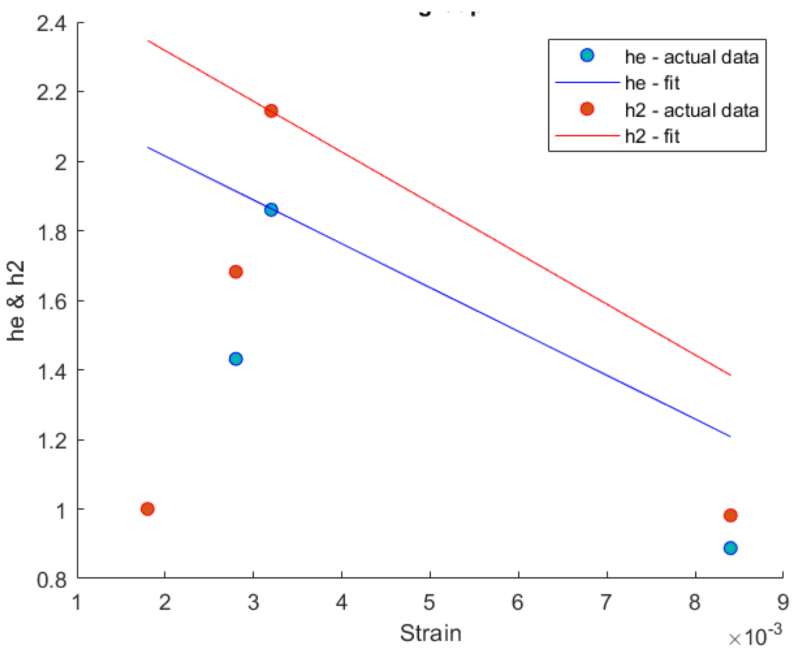


Figure 11: h_e and h_2 of BMD group 2 as a function of strain

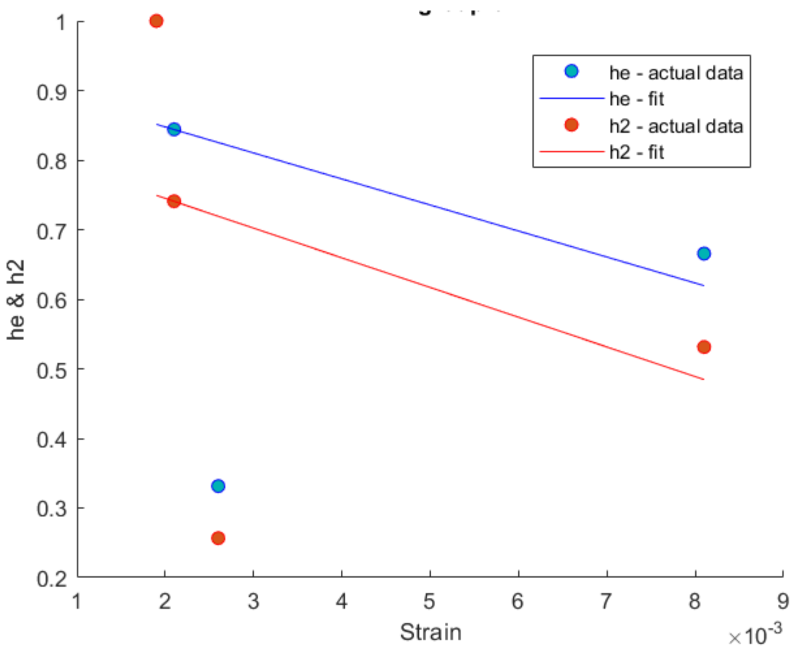


Figure 12: h_e and h_2 of BMD group 3 as a function of strain

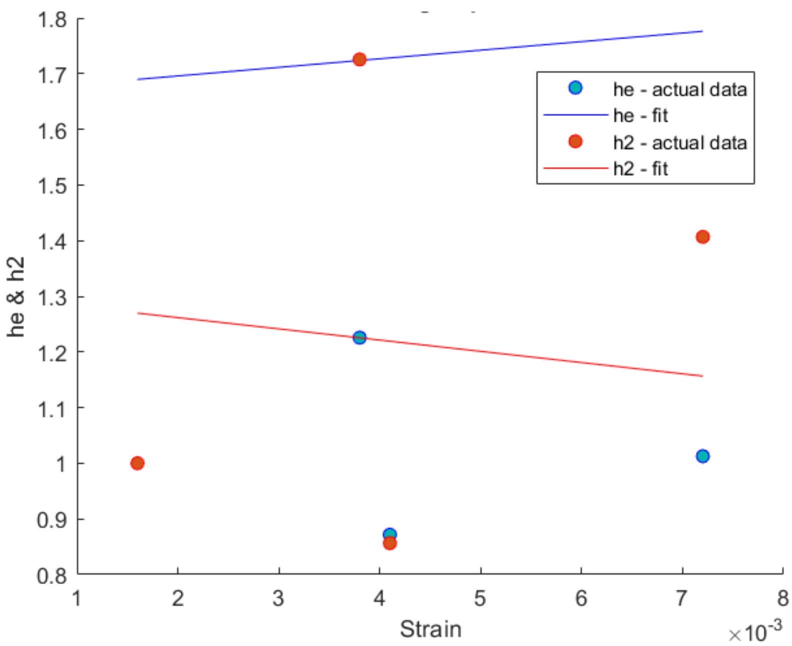


Figure 13: h_e and h_2 of BMD group 4 as a function of strain

BMD Group	L	M	O	P
1	213.6	0.5730	188.9	0.6223
2	-145.8	2.609	-126	2.2661
3	-42.7	0.8303	-37.4	0.9224
4	-20.19	1.302	15.43	1.6650

Table 2: Value of variables of $h_2(\epsilon)$ and $h_e(\epsilon)$ of each BMD group

The Equation 11 with the values of Table 1 and Table 2 is plotted with experimental data in Figures 14,15, 16 and 17.

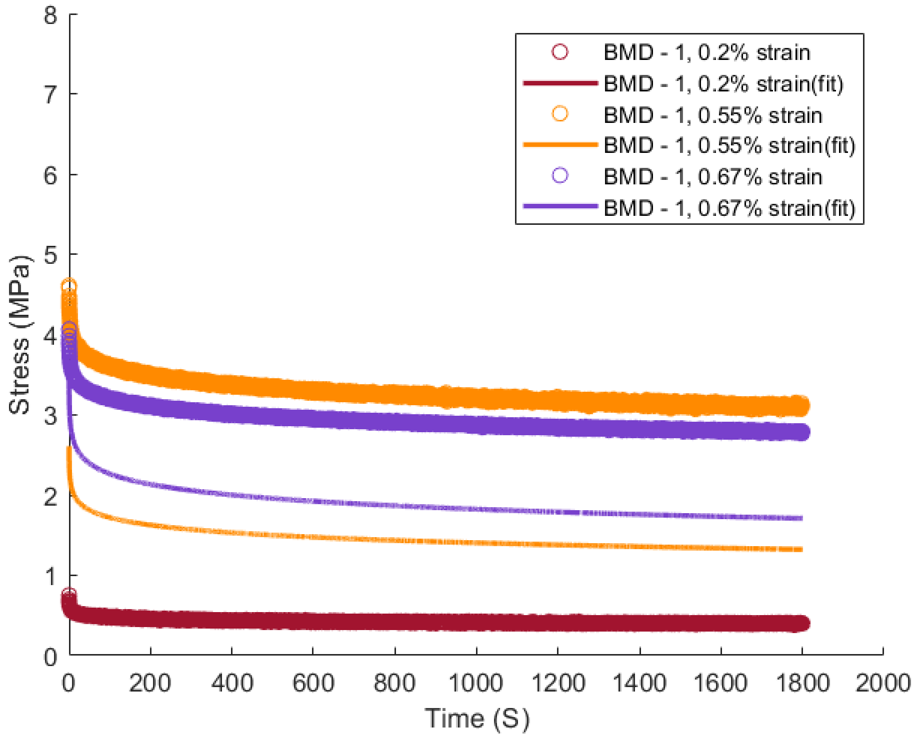


Figure 14: Experimental data with 0.2% Schapery model - BMD Group 1

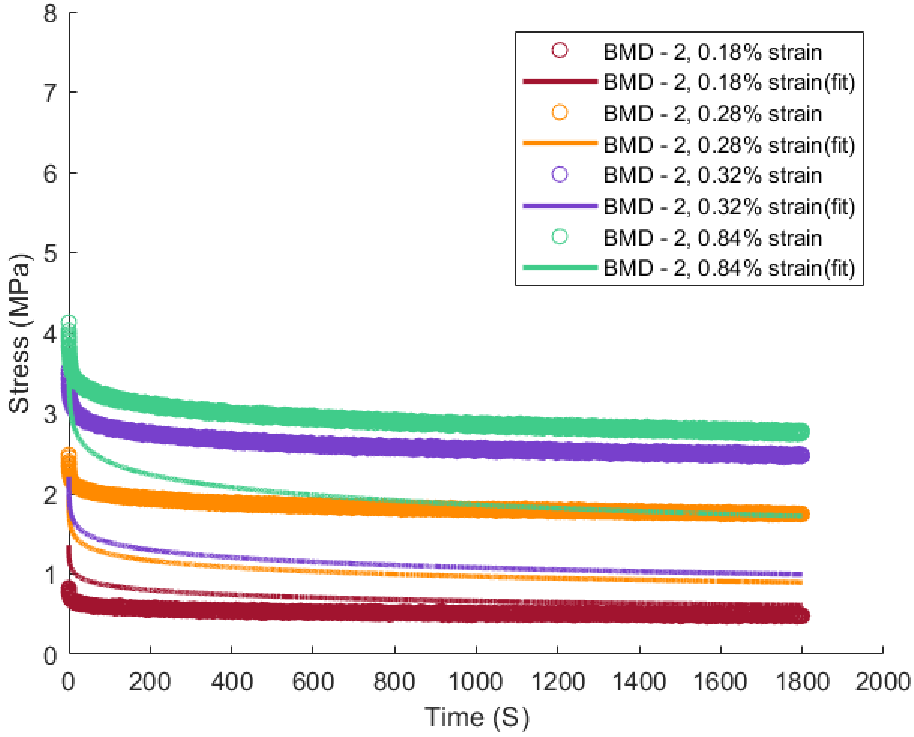


Figure 15: Experimental data with 0.18% Schapery model - BMD Group 2

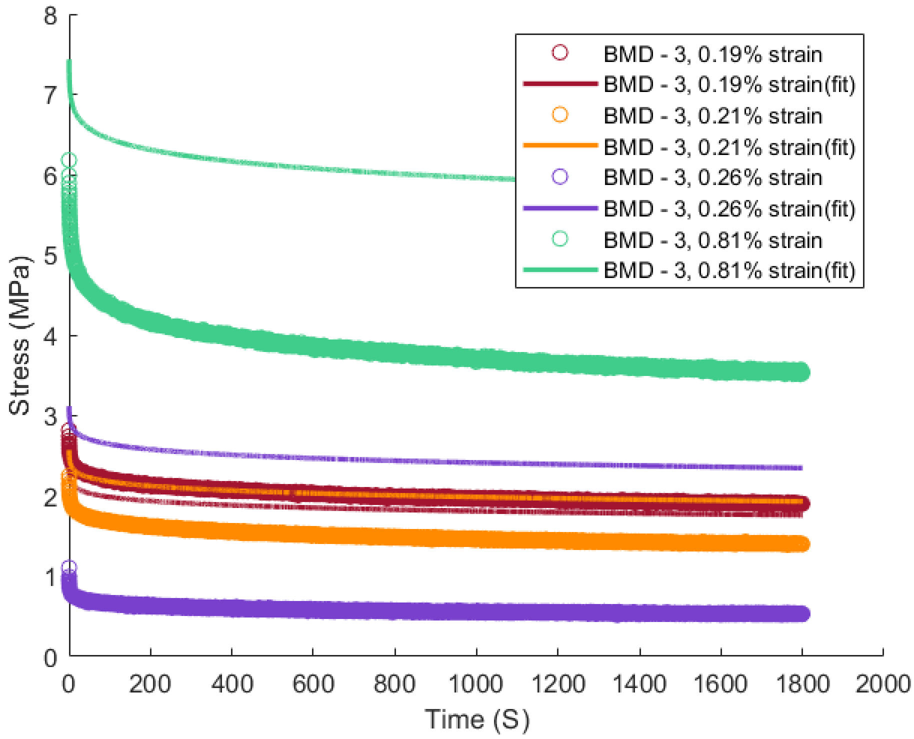


Figure 16: Experimental data with 0.19% Schapery model - BMD Group 3

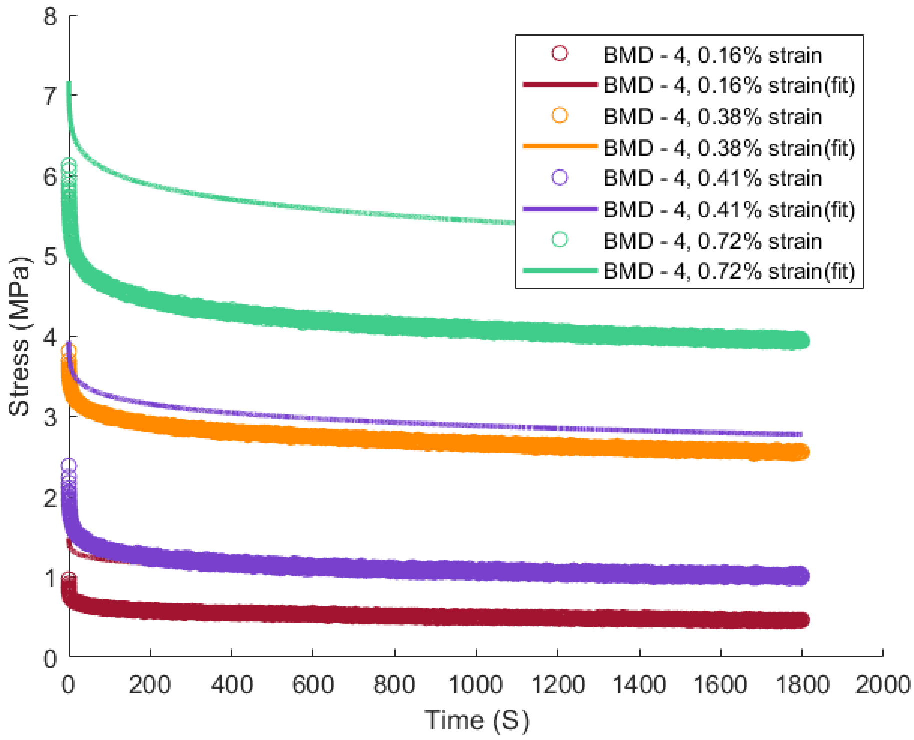


Figure 17: Experimental data with 0.16% Schapery model - BMD Group 4

The above procedure provided the equation of stress with respect to strain. Moreover, the aim of this study was to quantify the correlation between BMD and stress.

The variables of Equation 8 other than the strain and time were related to reference BMDs (BMD_r) to determine if they had any correlation. The graphs of C, n, h_e and h_2 versus BMD was shown in figures 18, 19 and 20.

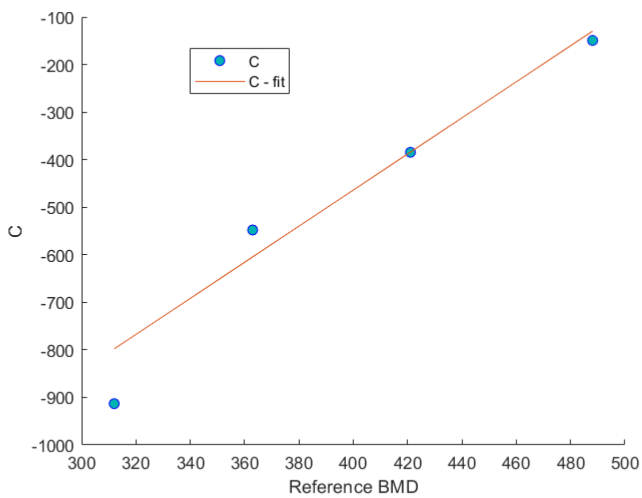


Figure 18: Variable C as a function of BMD

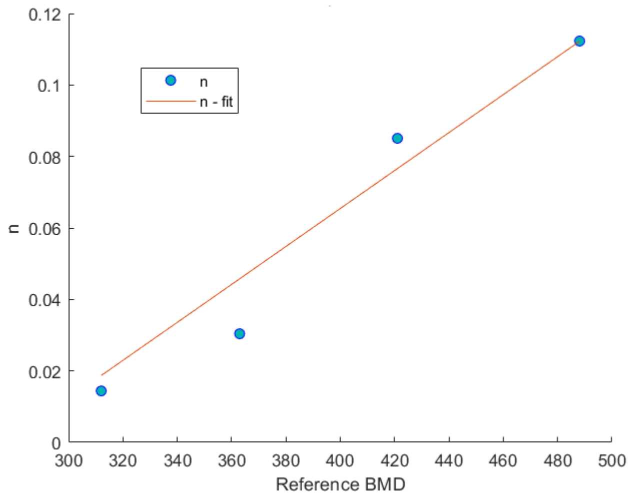


Figure 19: Variable n as a function of BMD

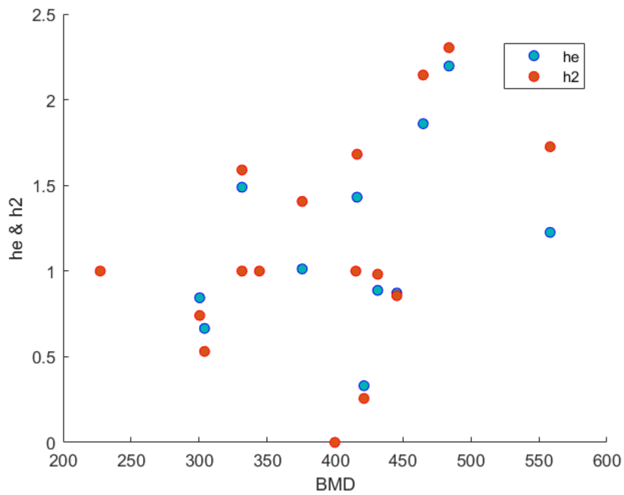


Figure 20: Overall h_e and h_2 as a function of BMD

The variables C and n show a correlation with BMD and their functions were given by equations 18 and 19.

$$f_c = 3.795 * BMD_r - 1982 \tag{18}$$

$$f_n = 0.0005311 * BMD_r - 0.147 \tag{19}$$

6.2 Modified Superposition Method (MSM)

For the same BMD groups, MSM model was implemented to determine the correlation. The initial strain levels and the reference BMDs were similar to the Shapery method. The Stress Versus Time graphs for each BMD group were plotted with the data obtained using the MTS output and DIC images. The logarithm of stress and time were calculated for further procedures in this method. The Curve Fitter toolbox in Matlab was used to identify the parameters of the Modified Superposition Method (Equation 14) as described below.

For each BMD group, the slope of the log-log stress-time graph of every sample was found and the slope-strain graph was fitted with Equation 15 ($B(\epsilon)$). Then the initial relaxation modulus - strain graph was fitted Equation 16 ($A(\epsilon)$).

The values of the variables H, J, D and F of Equation 17 are given in the table 3

BMD Group	D	F	H	J
1	3.558	-0.070	67699	286.07
2	0.887	-0.052	-36024	882.73
3	-1.968	-0.055	-53228	1131.49
4	6.283	-0.098	37066	605.66

Table 3: Value of variables of D, F, H and J of each BMD group

The Equation 17 with the values of Table 3 is plotted with the experimental data as shown in figures 21, 22, 23 and 24.

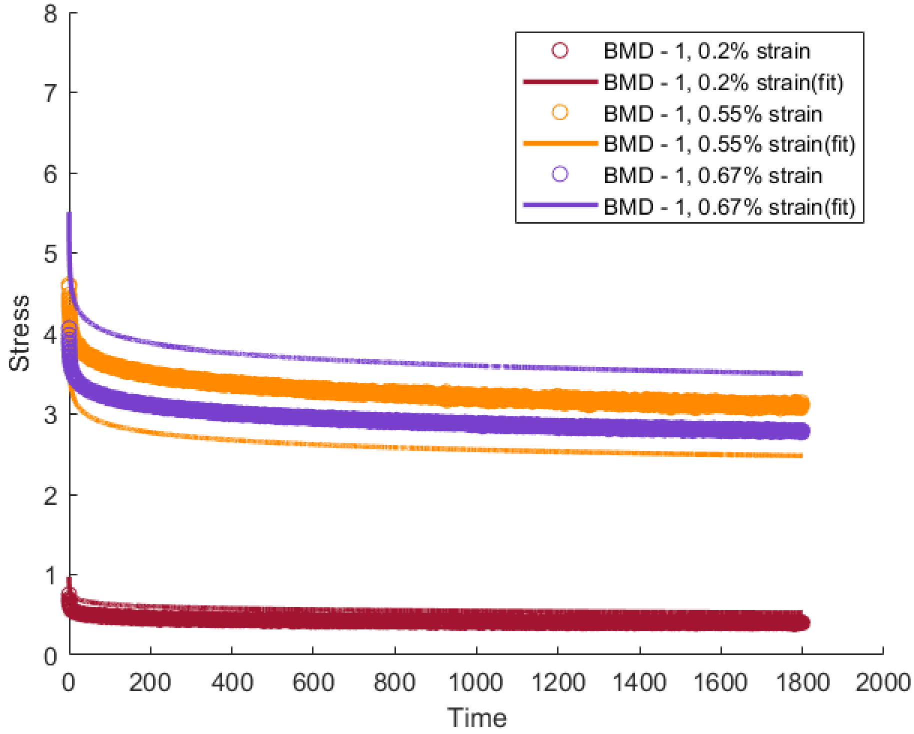


Figure 21: Experimental data with 0.2% MSM model - BMD Group 1

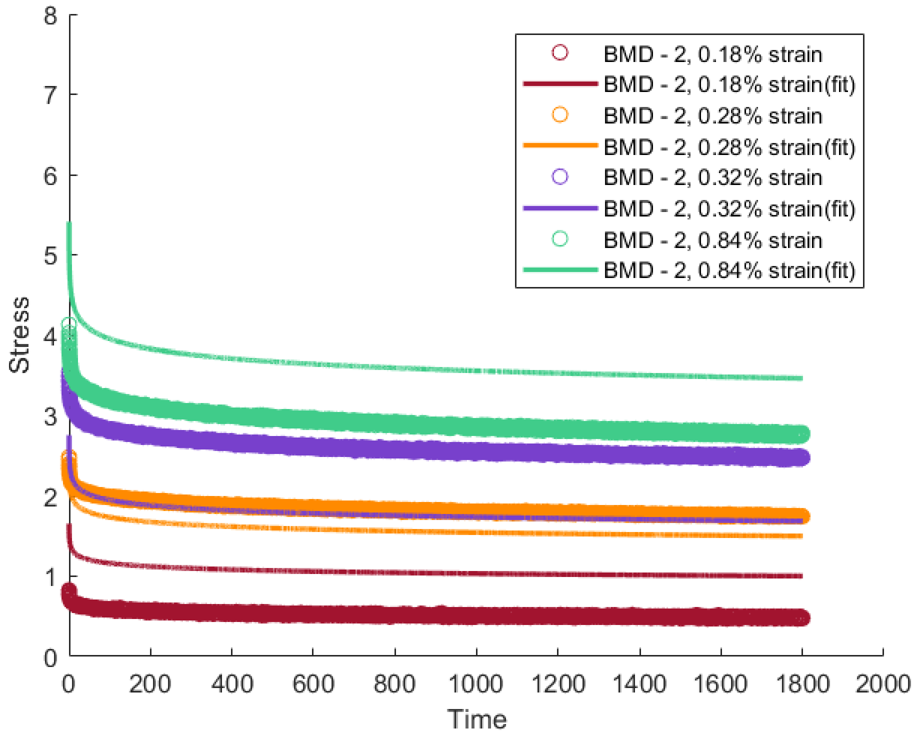


Figure 22: Experimental data with 0.18% MSM model - BMD Group 2

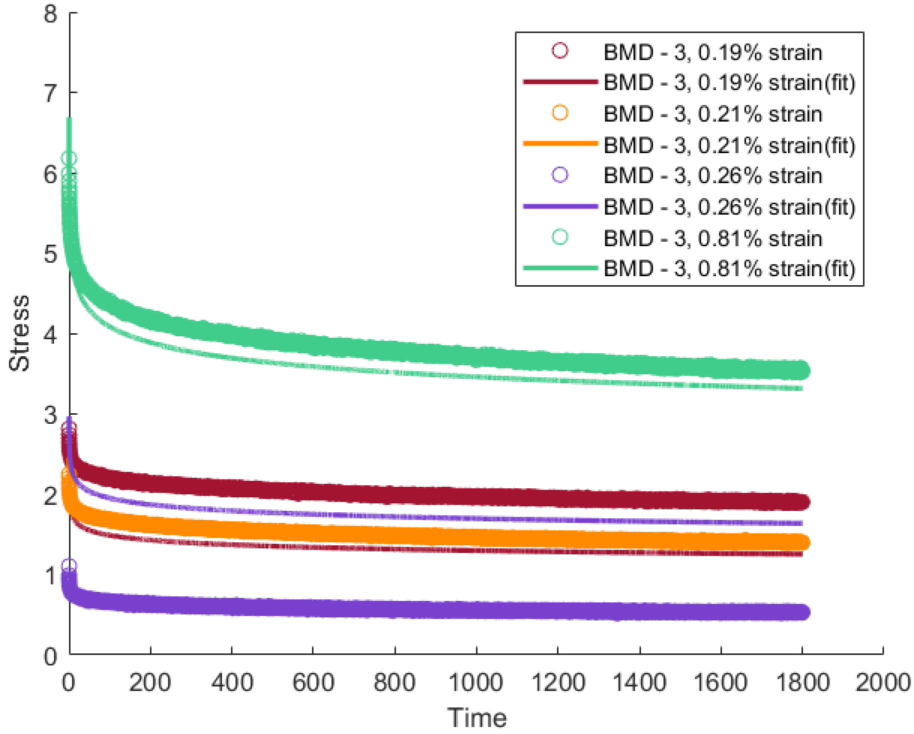


Figure 23: Experimental data with 0.19% MSM model - BMD Group 3

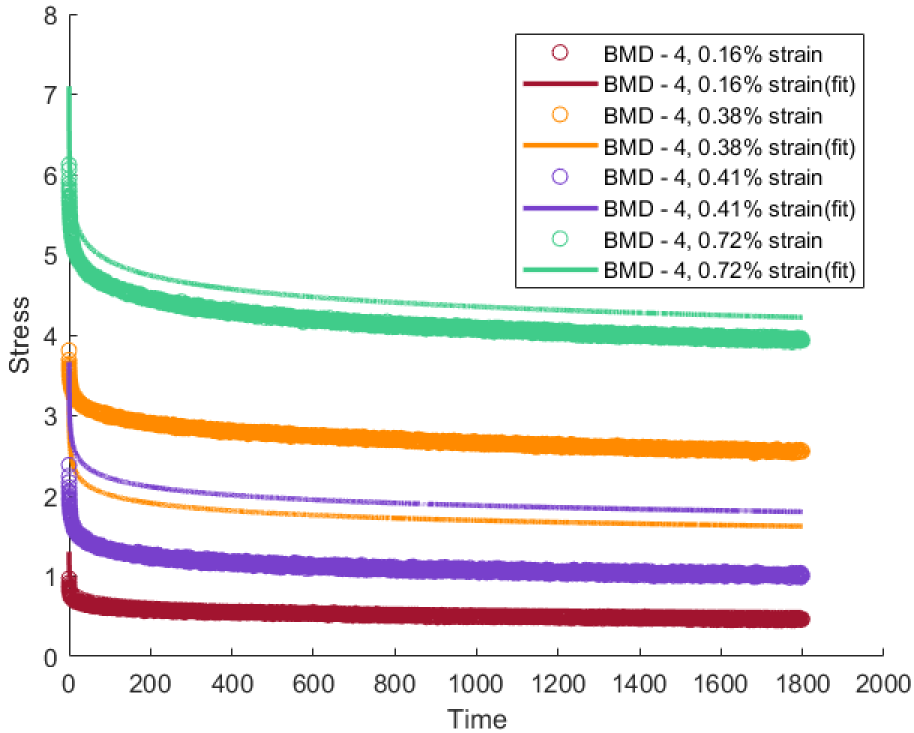


Figure 24: Experimental data with 0.16% MSM model - BMD Group 4

The above procedure provided the equation of stress with respect to strain. Moreover, the aim of this study was to quantify the correlation between BMD and stress.

The variables of Equation 17 other than the strain and time were related to reference BMDs to determine if they had any correlation. The functions of D, F, H and J versus BMD were found to be equations 20, 21, 22 and 23 and the graphs were plotted as shown in figures 25, 26, 27 and 28 to determine the correlation and the influence (linear/non-linear) of BMD on these variables.

$$f_D = 0.01165 * BMD_r - 2.422 \tag{20}$$

$$f_F = -0.0001626 * BMD_r - 0.004826 \tag{21}$$

$$f_H = -140 * BMD_r + 59330 \tag{22}$$

$$f_J = 1.788 * BMD_r + 18.48 \tag{23}$$

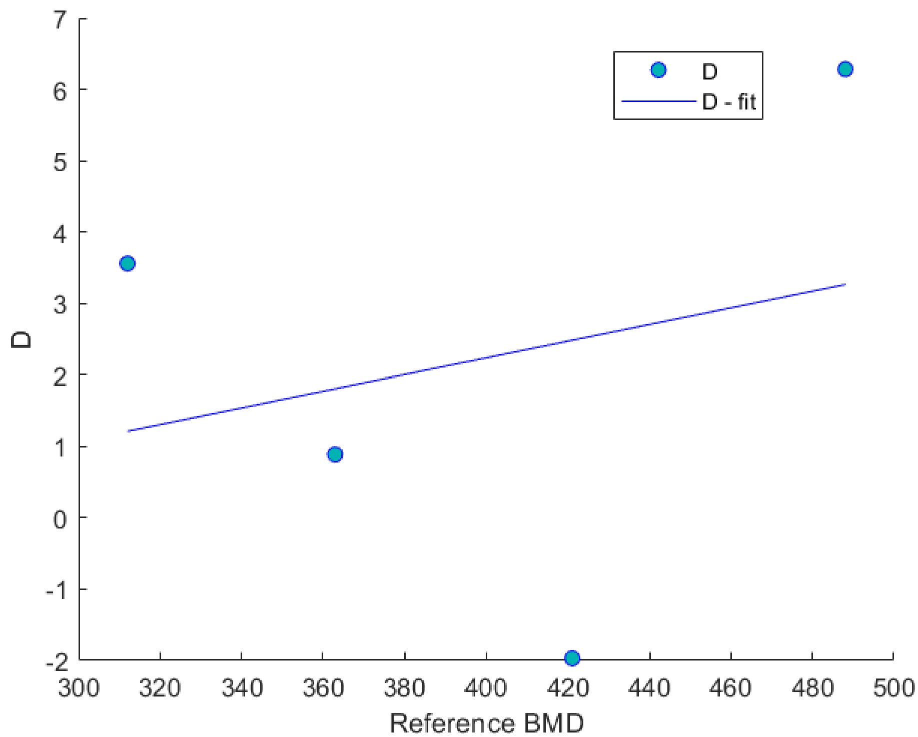


Figure 25: Variable D as a function of BMD

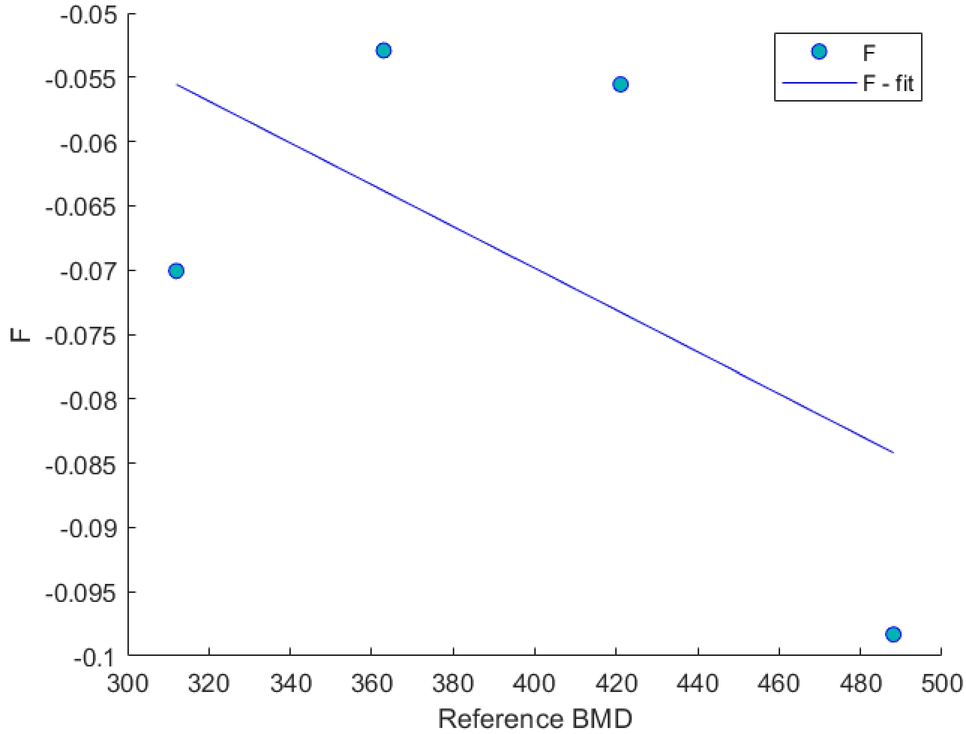


Figure 26: Variable F as a function of BMD

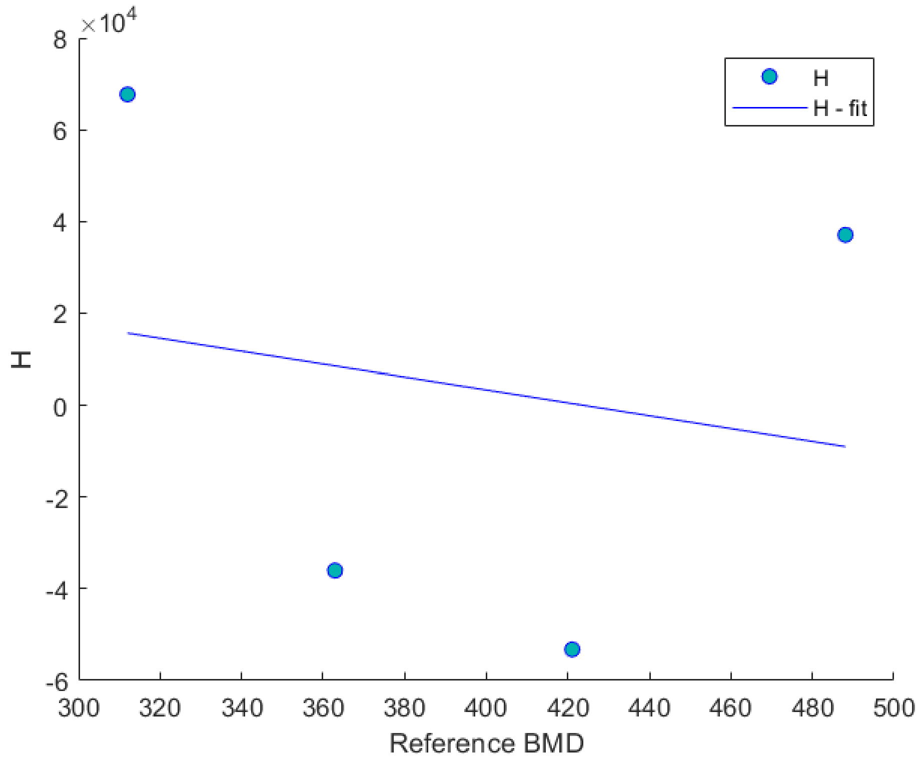


Figure 27: Variable H as a function of BMD

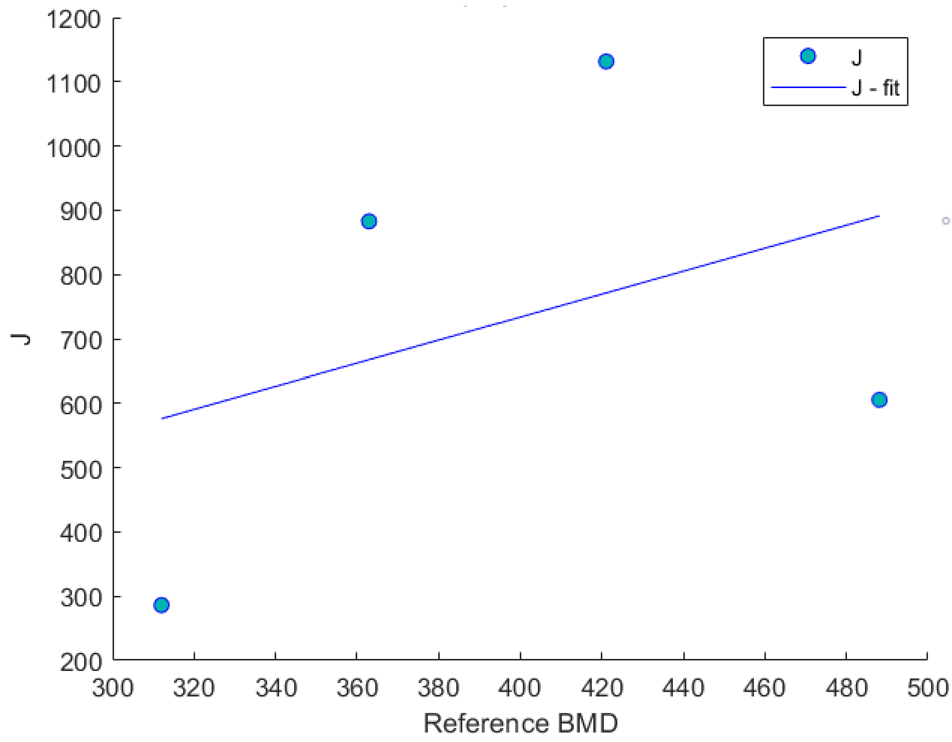


Figure 28: Variable J as a function of BMD

6.3 Comparison between Schapery & MSM models

While comparing the Schapery model and MSM model, MSM was easy to compute with less number of variables. Even if Schapery model related BMD, strain and stress, the mean of root mean square error (Coefficients of determinants or R^2) of Schapery method was 0.9451 while in MSM, it was 0.5807. In simple terms, there was about 66.82% variation in the fit of the final data point and the experimental data in Schapery method and it was 46.43% in MSM. This shows that MSM has a better fit with this stress relaxation data. The higher root mean square error and higher error percentage were assumed to be due to the lack of sufficient bovine samples. The linear parameters of Schapery method had a correlation with BMD while the non-linear parameters of the Schapery method and the parameters of MSM had no correlation with BMD. This might also be due to the lack of sufficient samples. The Schapery formula with bone mineral density, strain and stress had an error of 88.2%.

7 Discussion

Several studies have been performed to understand the time dependent mechanical properties of the trabecular bone [11, 24]. The studies that study the time independent properties of the trabecular bone relate the properties like Young's modulus to the density of the samples in order to develop a material model [1], but the relationship between time dependent viscoelastic properties with the density and strain has not been investigated to the finest of our information. A study related the viscoelastic properties of trabecular bone to the strain but failed to relate it to the density [4].

In this study, we conducted 24 hour stress relaxation experiments on 16 bovine trabecular femur samples and the behavior was quantified using non-linear viscoelastic theory based on the Schapery Method and Modified Superposition Method. Both the Schapery method and Modified Superposition Method directly relates stress to applied stain level using Young's modulus (time dependent properties) of a bone. But the goal of this assignment is to relate the time dependent and independent properties to strain level. The attempt to relate the variables of these methods to BMD in order to identify correlation resulted in the following discussions:

While comparing the Schapery model and MSM model, MSM was easy to compute with less number of variables. The mean of R^2 of Schapery method was 0.9451 while in MSM, it was 0.5807. The lower the root mean square error, the better the model is able to fit the data. This shows that MSM has a better fit with this stress relaxation data than the Schapery method.

In Schapery method, the graphs of the variables h_e and h_2 seems to have no correlation with BMD. This statement is supported by the overall graph of h_e vs BMD and h_2 vs BMD (Figure 20). The same holds true for the variables of Modified Superposition Method (Figure 25, 26, 27, 28). But there is a linear relationship between the variables C and n of the Schapery model with the BMD (Figure 18,19). It means with an increase in density, there is an increase in the value of C and n. This clearly defines that there is a positive correlation between the stress relaxation response and strain level coupled with BMD in relation to the linear variables. For non-linear variables, no correlation was observed.

After fitting the relationship equations of C and n to the final Schapery formula, the Schapery fit had an error of 88.2%. Due to a high error rate, the correlation between bone mineral density and strain coupled with stress response using Schapery model was not ideal.

There were a few challenges encountered during the course of the assignment. The lack of sufficient publications in determining the viscoelastic properties of trabecular bone made it difficult to conclude preconditioning and further procedures. The availability of less number of bovine samples intricated the process of data fitting. This reflected in higher room mean square error in the results and it made the process of concluding the correlation between the non-linear parameters of the Schapery model and BMD difficult. But the minimum number of samples required to perform a correlation is still unkown. The concept of relating the BMD to the stress was quite complex as the BMD varied for every 16 sample. Initially, the applied strain levels were thought to be 0.2%,0.4%,0.6% and 0.8%. But, the DIC images confirmed that the strains were different for each sample. So, the concept of fitting the data of same strain level and different BMDs was not possible. This led to the assumption of the samples belonging to the same BMD group have a common (average) BMD. After grouping the BMDs, one sample in BMD group 1 varied much than the other samples in the same group. So the particular sample was omitted for fitting the curves of Schapery model and Modified Superposition Method.

Further studies related to this research might include the determination of the influence of non-linear parameters of the Schapery model on BMD again by including more number of samples. This might have a greater influence on the result that the correlation is possible with linear parameters in the model.

The problem statement and the goals of the assignment as in Chapter 3 were answered during the course of the internship. In bovine femoral trabecular bone, the stress relaxation response and the initial strain level coupled with bone mineral density had a correlation between each other in relation to the linear variables. The BMD of each sample was found using softwares Slicer, Hypermesh, Mentat and Matlab. A custom-written script was used in Matlab to find the density of each sample. After performing the stress relaxation experiments and data fitting, MSM model predicted the viscoelastic behavior better than Schapery method. The Schapery method incorporated with the density did not predict the viscoelastic behavior better while MSM method had no correlation with density making the method not possible to incorporate with density. The entire assignment was performed completely within the duration of the internship.

8 Conclusion

This research was designed to study the correlation between stress response and strain together with density in bovine trabecular bone using stress relaxation experiments. First, the density was found and then 24 hour stress relaxation experiments were performed. Due to time constraints, in this assignment, the density was found for human femoral and tibial samples which were harvested during the assignment and the experiment was performed on already harvested 16 bovine femur samples in which the material assignments were also performed previously.

The stress relaxation data of the reference samples of each BMD group were used to fit the two existing models: Schapery model and Modified Superposition Model. The

MSM produced better fit than Schapery model. MSM had less number of parameters to compute too. But, both of these methods related stress to strain and not to BMD. An attempt was taken to relate the parameters of these models to BMD. At this moment, MSM method did not show any correlation between BMD, strain and stress response due to the absence of linear parameters. The linear parameters of the Schapery model indicate the change of elastic modulus changes over time and had a better correlation with BMD. This brings the conclusion that BMD and strain level can be related to the stress response with a linear parameter in the model. The Schapery formula with bone mineral density, strain and stress had an error of 88.20%. Due to a high error rate, the correlation between bone mineral density and strain coupled with stress response using Schapery model was not ideal. With the data provided, this report helps one to understand the correlation between the viscoelastic properties and strain level coupled with density and develop a model with both time-dependent and independent properties of trabecular bone.

References

[1] J. D. Currey, *Bones: structure and mechanics.* Princeton university press, 2006.

[2] I. H. Parkinson and N. L. Fazzalari, "Characterisation of trabecular bone structure," in *Skeletal aging and osteoporosis.* Springer, 2013, pp. 31–51.

[3] T. M. Keaveny, E. F. Morgan, G. L. Niebur, and O. C. Yeh, "Biomechanics of trabecular bone," *Annual Review of Biomedical Engineering*, vol. 3, no. 1, pp. 307–333, 2001, pMID: 11447066. [Online]. Available: https://doi.org/10.1146/annurev.bioeng.3.1.307

[4] K. Manda, R. J. Wallace, S. Xie, F. Levrero-Florencio, and P. Pankaj, "Nonlinear viscoelastic characterization of bovine trabecular bone," *Biomechanics and Modeling in Mechanobiology*, vol. 16, no. 1, pp. 173–189, Jul. 2016. [Online]. Available: https://doi.org/10.1007/s10237-016-0809-y

[5] C. Sandino, D. Mcerlain, J. Schipilow, and S. Boyd, "The poro-viscoelastic properties of trabecular bone: A micro computed tomography-based finite element study," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 44, 04 2015.

[6] H. Zilch, A. Rohlmann, G. Bergmann, and R. Kölbel, "Material properties of femoral cancellous bone in axial loading," *Archives of Orthopaedic and Traumatic Surgery*, vol. 97, no. 4, pp. 257–262, Dec. 1980. [Online]. Available: https://doi.org/10.1007/bf00380706

[7] *Biomaterials for Bone Regeneration.* Elsevier, 2014. [Online]. Available: https://doi.org/10.1016/c2013-0-16370-2

[8] A. M. Birzle and W. A. Wall, "A viscoelastic nonlinear compressible material model of lung parenchyma – experiments and numerical identification," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 94, pp. 164–175, Jun. 2019. [Online]. Available: https://doi.org/10.1016/j.jmbbm.2019.02.024

[9] D. D. Greef, J. Aernouts, J. Aerts, J. T. Cheng, R. Horwitz, J. J. Rosowski, and J. J. Dirckx, "Viscoelastic properties of the human tympanic membrane studied with stroboscopic holography and finite element modeling," *Hearing Research*, vol. 312, pp. 69–80, Jun. 2014. [Online]. Available: https://doi.org/10.1016/j.heares.2014.03.002

[10] E. Ramirez, J. Angeles, O. Ruiz Cervantes, V. Jacobo, and A. Ortiz Prado, "Viscoelastic characterization of bovine trabecular bone samples," 05 2015.

[11] V. Quaglini, V. L. Russa, and S. Corneo, "Nonlinear stress relaxation of trabecular bone," *Mechanics Research Communications*, vol. 36, no. 3, pp. 275–283, Apr. 2009. [Online]. Available: https://doi.org/10.1016/j.mechrescom.2008.10.012

[12] K. Manda, S. Xie, R. Wallace, F. Levrero, and P. Pankaj, "Linear viscoelasticity - bone volume fraction relationships of bovine trabecular bone," *Biomechanics and Modeling in Mechanobiology*, vol. 15, 12 2016.

[13] J. Yamashita, B. Furman, H. Rawls, X. Wang, and C. Agrawal, "The use of dynamic mechanical analysis to assess the viscoelastic properties of human cortical bone," *Journal of biomedical materials research*, vol. 58, pp. 47–53, 02 2001.

[14] S. Duenwald-Kuehl, R. Phd, and R. Lakes, "Stress relaxation and recovery in tendon and ligament: Experiment and modeling," *Biorheology*, vol. 47, pp. 1–14, 01 2010.

[15] D. Gamby and L. Blugeon, "On the characterization by schapery's model of non-linear viscoelastic materials," *Polymer Testing*, vol. 7, no. 2, pp. 137–147, 1987. [Online]. Available: https://doi.org/10.1016/0142-9418(87)90007-9

[16] R. Schapery, "On the characterization of nonlinear viscoelastic materials," *Polymer Engineering & Science*, vol. 9, pp. 295 – 310, 07 1969.

[17] L. Pupure, J. Vārna, and R. Joffe, "Applications and limitations of non-linear viscoelastic model for simulation of behaviour of polymer composites," 2015.

[18] C. Blandford, "Nonlinear visco-elastic models applied to human medial collateral ligaments," *CMBES Proceedings*, vol. 40, 2017.

[19] P. Provenzano, R. Lakes, T. Keenan *et al.*, "Nonlinear ligament viscoelasticity," *Annals of biomedical engineering*, vol. 29, no. 10, pp. 908–914, 2001.

[20] F. Safshekan, "Nonlinear viscoelastic properties of human dentin under uniaxial tension," *Dental materials: official publication of the Academy of Dental Materials*, vol. 37, 12 2020.

[21] P. P. Provenzano, R. S. Lakes, D. T. Corr, and R. Vanderby, "Application of nonlinear viscoelastic models to describe ligament behavior," *Biomechanics and Modeling in Mechanobiology*, vol. 1, no. 1, pp. 45–57, Jun. 2002. [Online]. Available: https://doi.org/10.1007/s10237-002-0004-1

[22] S. Berahmani, D. Janssen, D. Wolfson, M. de Waal Malefijt, C. K. Fitzpatrick, P. J. Rullkoetter, and N. Verdonschot, "FE analysis of the effects of simplifications in experimental testing on micromotions of uncemented femoral knee implants,"

Journal of Orthopaedic Research, vol. 34, no. 5, pp. 812–819, Dec. 2015. [Online]. Available: https://doi.org/10.1002/jor.23074

[23] J. H. Keyak and Y. Falkinstein, “Comparison of in situ and in vitro ct scan-based finite element model predictions of proximal femoral fracture load,” *Medical Engineering Physics*, vol. 25, no. 9, pp. 781–787, 2003. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S135045330300081X

[24] S. M. Bowman, T. M. Keaveny, L. J. Gibson, W. C. Hayes, and T. A. McMahon, “Compressive creep behavior of bovine trabecular bone,” *Journal of Biomechanics*, vol. 27, no. 3, pp. 301–310, Mar. 1994. [Online]. Available: https://doi.org/10.1016/0021-9290(94)90006-x

[25] E. F. Morgan, J. J. Lee, and T. M. Keaveny, “Sensitivity of multiple damage parameters to compressive overload in cortical bone,” *Journal of Biomechanical Engineering*, vol. 127, no. 4, p. 557, 2005. [Online]. Available: https://doi.org/10.1115/1.1933916

[26] Y. C. Lou and R. A. Schapery, “Viscoelastic characterization of a nonlinear fiber-reinforced plastic,” *Journal of Composite Materials*, vol. 5, no. 2, pp. 208–234, 1971.

[27] V. G. Karnaukhov, “Nonlinear theory of thermoviscoelasticity for a generalized thermo-rheologically simple material,” *Soviet Applied Mechanics*, vol. 14, no. 1, pp. 11–18, Jan. 1978. [Online]. Available: https://doi.org/10.1007/bf00887538

Appendix A

Obtaining the ROI & parameters of ROI from Stress Relaxation Data

```
%% Run ReadDataMTS first
clc

[dataStruct] =  ReadDataMTS_new();

%% obtain ROI for time Experiment

MaxForce = [];
MaxForceI = [];
EndTimeExperiment = [];
MinimumDifference = [];
MinForceI = [];
MinForce = [];
PercentageStressRelaxation = [];
ForcePosROI = [];
dataStruct_ROI =  struct('TimeROI', [],'ForcePosROI',[], 'StressSample',[],'StressRelMod',[],'StressRelMod_fit1',[]);
TimeROI = [];

for k = 1:size(dataStruct,2)
    if k==4 || k==9 || k==12 || k==13 || k==16
    [MaxForce(k),MaxForceI(k)] = min(dataStruct(k).kNForce);
    DurationExperiment = 86400; % aanpassen
    EndTimeExperiment(k) = dataStruct(k).Time(MaxForceI(k))+DurationExperiment;
    [MinimumDifference(k),MinForceI(k)]=min(abs(dataStruct(k).Time-EndTimeExperiment(k)));
    MinForce(k) = dataStruct(k).Force(MinForceI(k));
    PercentageStressRelaxation(k) = 100 - (MinForce(k) / MaxForce(k) * 100); %handige value

    dataStruct_ROI(k).TimeROI = dataStruct(k).Time(MaxForceI(k):MinForceI(k));
    dataStruct_ROI(k).ForcePosROI = dataStruct(k).kNForce(MaxForceI(k):MinForceI(k))*-1;

    NumberOfEndRemovers = 0;

while dataStruct_ROI(k).ForcePosROI(end) - dataStruct_ROI(k).ForcePosROI(end-1) > 5 || dataStruct_ROI(k).ForcePosROI(end) - dataStruct_ROI(k).ForcePosROI(end-1) < -5
    dataStruct_ROI(k).ForcePosROI(end) = [];
```

```
    dataStruct_ROI(k).TimeROI(end) = [];
    NumberOfEndRemovers = NumberOfEndRemovers + 1;
end
    else
    [MaxForce(k),MaxForceI(k)] = min(dataStruct(k).Force);
    DurationExperiment = 86400; % aanpassen
    EndTimeExperiment(k) = dataStruct(k).Time(MaxForceI(k))+
   DurationExperiment;
    [MinimumDifference(k),MinForceI(k)]=min(abs(dataStruct(k).Time-
   EndTimeExperiment(k)));
    MinForce(k) = dataStruct(k).Force(MinForceI(k));
    PercentageStressRelaxation(k) = 100 - (MinForce(k) / MaxForce(k) *
   100); %handige value

    dataStruct_ROI(k).TimeROI = dataStruct(k).Time(MaxForceI(k):
   MinForceI(k));
    dataStruct_ROI(k).ForcePosROI = dataStruct(k).Force(MaxForceI(k):
   MinForceI(k))*-1;

    NumberOfEndRemovers = 0;

while dataStruct_ROI(k).ForcePosROI(end) - dataStruct_ROI(k).
   ForcePosROI(end-1) > 5 || dataStruct_ROI(k).ForcePosROI(end) -
   dataStruct_ROI(k).ForcePosROI(end-1) < -5
    dataStruct_ROI(k).ForcePosROI(end) = [];
    dataStruct_ROI(k).TimeROI(end) = [];
    NumberOfEndRemovers = NumberOfEndRemovers + 1;
end
    end
end

%% Assigning the applied strains

PercentageAppliedStrain = [];
AppliedStrain = [];
StressSample = [];
StressRelMod = [];

PercentageAppliedStrain = [0.26 0.18 0.20 0.55 0.20 0.19 0.32 0.67 0.84
     0.38 0.21 0.41 0.72 0.16 0.28 0.81];
DiameterSample = [12.8 12.8 12.8 12.8 12.8 12.8 12.8 12.8 12.8 12.85
   12.85 12.8 12.85 12.85 12.85 12.85]; %aanpassen

for k=1:size(dataStruct,2)
    AppliedStrain(k) = PercentageAppliedStrain(k)/100;
    AreaSample(k) = (pi*(DiameterSample(k))^2) / 4;
```

```
end

%% Obtain Stress and Relaxation Modulus for 24 hr data

for k=1:size(dataStruct,2)
    dataStruct_ROI(k).StressSample = dataStruct_ROI(k).ForcePosROI /
    AreaSample(k);
    dataStruct_ROI(k).StressRelMod = dataStruct_ROI(k).StressSample/
    AppliedStrain(k);

%% Stress Relaxation Modulus Fit

A = [];
B = [];
C = [];

[fitresult, gof] = CreatFit1_Powerlaw(dataStruct_ROI(k).TimeROI,
    dataStruct_ROI(k).StressRelMod); %nu nog met powerlaw, y=a*x^b+c
R2 = gof.rsquare;
Coefficients = coeffvalues(fitresult);
A(k) = Coefficients(1);
B(k) = Coefficients(2);
C(k) = Coefficients(3);

dataStruct_ROI(k).StressRelMod_fit1 = A(k)*dataStruct_ROI(k).TimeROI.^B
    (k)+C(k);
end

%% obtain ROI for 0.5 hour

dataStruct_ROI_half_hr =  struct('TimeROI_half_hr',[], '
    ForcePosROI_half_hr',[],'StressSample_half_hr',[],'
    StressRelMod_half_hr',[],'StressRelMod_fit1_half_hr',[],'
    PercentageStressRelaxation_24hr_half_hr',[]);
EndTimeExperiment_half_hr = [];
MinimumDifference_half_hr = [];
MinForceI_half_hr = [];
MinForce_half_hr = [];
TimeROI_half_hr = [];
ForcePosROI_half_hr = [];
StressRelMod_half_hr = [];
StressSample_half_hr = [];
StressRelMod_fit1_half_hr = [];
PercentageStressRelaxation_24hr_half_hr = [];

DurationExperiment = 1800; % aanpassen
```

```
for k=1:size(dataStruct,2)
    if k==4 || k==9 || k==13 || k==16 || k==12
    EndTimeExperiment_half_hr(k) = dataStruct(k).Time(MaxForceI(k))+DurationExperiment;
    [MinimumDifference_half_hr(k),MinForceI_half_hr(k)]=min(abs(dataStruct(k).Time-EndTimeExperiment_half_hr(k)));
    MinForce_half_hr(k) = dataStruct(k).kNForce(MinForceI_half_hr(k));
    PercentageStressRelaxation_half_hr(k) = 100 - (MinForce_half_hr(k) / MaxForce(k) * 100); %handige value

    dataStruct_ROI_half_hr(k).TimeROI_half_hr = dataStruct(k).Time(MaxForceI(k):MinForceI_half_hr(k));
    dataStruct_ROI_half_hr(k).ForcePosROI_half_hr = dataStruct(k).kNForce(MaxForceI(k):MinForceI_half_hr(k))*-1;

    NumberOfEndRemovers_1hr = 0;

    while dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) - dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end-1) > 5 || dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) - dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end-1) < -5
          dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) = [];
          dataStruct_ROI_half_hr(k).TimeROI_half_hr(end) = [];
          NumberOfEndRemovers_1hr = NumberOfEndRemovers_1hr + 1;
    end

    else
    EndTimeExperiment_half_hr(k) = dataStruct(k).Time(MaxForceI(k))+DurationExperiment;
    [MinimumDifference_half_hr(k),MinForceI_half_hr(k)]=min(abs(dataStruct(k).Time-EndTimeExperiment_half_hr(k)));
    MinForce_half_hr(k) = dataStruct(k).Force(MinForceI_half_hr(k));
    PercentageStressRelaxation_half_hr(k) = 100 - (MinForce_half_hr(k) / MaxForce(k) * 100); %handige value

    dataStruct_ROI_half_hr(k).TimeROI_half_hr = dataStruct(k).Time(MaxForceI(k):MinForceI_half_hr(k));
    dataStruct_ROI_half_hr(k).ForcePosROI_half_hr = dataStruct(k).Force(MaxForceI(k):MinForceI_half_hr(k))*-1;

    NumberOfEndRemovers_1hr = 0;

    while dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) - dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end-1) > 5 || dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) -
```

```
    dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end-1) < -5
            dataStruct_ROI_half_hr(k).ForcePosROI_half_hr(end) = [];
            dataStruct_ROI_half_hr(k).TimeROI_half_hr(end) = [];
            NumberOfEndRemovers_1hr = NumberOfEndRemovers_1hr + 1;
     end
     end
end

%% obtain Stress & Relaxation modulus for 0.5 hour

for k=1:size(dataStruct,2)

dataStruct_ROI_half_hr(k).StressSample_half_hr = dataStruct_ROI_half_hr
    (k).ForcePosROI_half_hr ./ AreaSample(k);
dataStruct_ROI_half_hr(k).StressRelMod_half_hr = dataStruct_ROI_half_hr
    (k).StressSample_half_hr / AppliedStrain(k);

A_half_hr = [];
B_half_hr = [];
C_half_hr = [];

%% Stress Relaxation Modulus Fit for 0.5 hour data

[fitresult_half_hr, gof_half_hr] = CreatFit1_Powerlaw(
    dataStruct_ROI_half_hr(k).TimeROI_half_hr, dataStruct_ROI_half_hr(k)
    .StressRelMod_half_hr); %nu nog met powerlaw, y=a*x^b+c
R2_half_hr = gof_half_hr.rsquare;
Coefficients_half_hr = coeffvalues(fitresult_half_hr);
A_half_hr(k) = Coefficients_half_hr(1);
B_half_hr(k) = Coefficients_half_hr(2);
C_half_hr(k) = Coefficients_half_hr(3);


dataStruct_ROI_half_hr(k).StressRelMod_fit1_half_hr = A_half_hr(k)*
    dataStruct_ROI(k).TimeROI.^B_half_hr+C_half_hr;
dataStruct_ROI_half_hr(k).PercentageStressRelaxation_24hr_half_hr = 100
     - ((-dataStruct_ROI_half_hr(k).StressRelMod_fit1_half_hr(end)*
    AppliedStrain(k)*AreaSample(k)) / MaxForce(k) * 100);

%% Calculate relevant values between fits

% Error Determination

y0 = dataStruct_ROI(k).StressRelMod_fit1;
y1 = dataStruct_ROI_half_hr(k).StressRelMod_fit1_half_hr;

```

```
dy = y0- y1 ; % error
abs_dy = abs(y0 - y1) ;   % absolute error
relerr = abs(y0 - y1)./ y0 ;  % relative error
pererr = abs(y0 - y1)./ y0*100 ;   % percentage error
mean_err_SR = mean(abs(y0 - y1)) ;    % mean absolute error
MSE_SR = mean((y0 - y1).^2) ;         % Mean square error
RMSE_SR = sqrt(mean((y0 - y1).^2)) ; % Root mean square error

mean_err_Force = mean_err_SR*AppliedStrain(k)*AreaSample(k);
MSE_SR_Force = MSE_SR*AppliedStrain(k)*AreaSample(k);
RMSE_SR_Force = RMSE_SR*AppliedStrain(k)*AreaSample(k);
end

%% Plot for Force versus Time - for ref

figure()
plot(dataStruct_ROI_half_hr(1).TimeROI_half_hr,dataStruct_ROI_half_hr
    (1).ForcePosROI_half_hr)
title('Force Versus Time')
xlabel('Time (S)')
ylabel('Force (N)')

%% Plot for E versus Time - for ref

figure()
plot(dataStruct_ROI(1).TimeROI, dataStruct_ROI(1).StressRelMod)
hold on
plot(dataStruct_ROI(1).TimeROI, dataStruct_ROI(1).StressRelMod_fit1,'
    LineWidth',2)
plot(dataStruct_ROI(1).TimeROI, dataStruct_ROI_half_hr(1).
    StressRelMod_fit1_half_hr,'LineWidth',2)
hold off
legend('Experimental Data', 'Fit 24 hours', 'Fit 0.5 hour')
xlabel('Time (S)')
ylabel('E (MPa)')
```

Appendix B

Schapery Method

```
% Run the MTSdatafitter_durationofstressrelaxation first
%% Log transformation & plot

BMD = [421.2866 331.5744 415.2768 483.8211 399.8866 227.2884 464.8261
    331.5744 431.4026 558.237 300.4444 445.4421 375.8612 344.386
    416.2005 304.0382];
BMD1 = [300.4444 304.0382 331.5744];
BMD2 = [331.7045 344.386 375.8612 399.8866];
BMD3 = [415.2768 416.2005 421.2866 431.4026];
BMD4 = [445.4421 464.8261 483.8211 558.237];

BMD_r(1) = mean(BMD1);
BMD_r(2) = mean(BMD2);
BMD_r(3) = mean(BMD3);
BMD_r(4) = mean(BMD4);

epsilon = [0.0026 0.0018 0.0020 0.0055 0.0020 0.0019 0.0032 0.0067
    0.0084 0.0038 0.0021 0.0041 0.0072 0.0016 0.0028 0.0081];
epsilon_r = [0.002 0.0018 0.0019 0.0016]; % reference strain level
epsilon1 = [0.002 0.0055 0.0067];
epsilon2 = [0.0032 0.0018 0.0084 0.0028];
epsilon3 = [0.0021 0.0026 0.0019 0.0081];
epsilon4 = [0.0038 0.0041 0.0072 0.0016];

dataStruct_final = struct('t',[],'E',[],'Stress',[],'log_t',[],'log_E'
    ,[],'log_stress',[],'E_schapery',[]);

for k = 1:size(dataStruct,2)
    dataStruct_final(k).t = dataStruct_ROI_half_hr(k).TimeROI_half_hr -
     dataStruct_ROI_half_hr(k).TimeROI_half_hr(1);
    dataStruct_final(k).t(1) = 0.1;
    dataStruct_final(k).Stress = dataStruct_ROI_half_hr(k).
    StressSample_half_hr;
    dataStruct_final(k).E = dataStruct_final(k).Stress./(epsilon(k));
    dataStruct_final(k).log_t = log10(dataStruct_final(k).t);
    dataStruct_final(k).log_E = log10(dataStruct_final(k).E);
    dataStruct_final(k).log_stress = log10(dataStruct_final(k).Stress);
end

%% Fit reference strain - BMD Group 1

E_1 = dataStruct_final(3).E;
```

```
Time_1 = dataStruct_final(3).t;
he(3) = 1; h2(3) = 1;
epsilon1 = [0.002 0.0055 0.0067];

G(1) = -913.2;
Ee(1) = 1215;
n(1) = 0.01439;


%% Fit for other strain levels - BMD Group 1

for k = 1:16
    if k == 8 || k == 4
            [fitresult_s1, gof_s1] = createFit_s1(dataStruct_final(k).t, dataStruct_final(k).E);
            Coeff_s1 = coeffvalues(fitresult_s1);
            he(k) = Coeff_s1(1);
            h2(k) = Coeff_s1(2);
    end
end

he_1 = [he(3) he(4) he(8)];
h2_1 = [h2(3) h2(4) h2(8)];

%% Fit reference strain - BMD Group 2

E_2 = dataStruct_final(2).E;
Time_2 = dataStruct_final(2).t;
he(2) = 1; h2(2) = 1;

G(2) = -547.9;
Ee(2) = 958.4;
n(2) = 0.0304;

%% Fit for other strain levels - BMD Group 2

for k = 1:16
    if k == 7 || k == 9 || k == 15
            [fitresult_s2, gof_s2] = createFit_s2(dataStruct_final(k).t, dataStruct_final(k).E);
            Coeff_s2 = coeffvalues(fitresult_s2);
            he(k) = Coeff_s2(1);
            h2(k) = Coeff_s2(2);
    end
end

```

```
he_2 = [he(7) he(2) he(9) he(15)];
h2_2 = [h2(7) h2(2) h2(9) h2(15)];

%% Fit reference strain - BMD Group 3

E_3 = dataStruct_final(6).E;
Time_3 = dataStruct_final(6).t;
he(6) = 1; h2(6) = 1;

G(3) = -384.5;
Ee(3) = 1729;
n(3) = 0.08508;

%% Fit for other strain levels - BMD Group 3

for k = 1:16
    if k == 11 || k == 1 || k == 16
            [fitresult_s3, gof_s3] = createFit_s3(dataStruct_final(k).t
    , dataStruct_final(k).E);
            Coeff_s3 = coeffvalues(fitresult_s3);
            he(k) = Coeff_s3(1);
            h2(k) = Coeff_s3(2);
    end
end

he_3 = [he(11) he(1) he(6) he(16)];
h2_3 = [h2(11) h2(1) h2(6) h2(16)];

%% Fit reference strain - BMD Group 4

E_4 = dataStruct_final(14).E;
Time_4 = dataStruct_final(14).t;
he(14) = 1; h2(14) = 1;


G(4) = -149.3;
Ee(4) = 636.1;
n(4) = 0.1123;

%% Fit for other strain levels - BMD Group 4

for k = 1:16
    if k == 10 || k == 12 || k == 13
            [fitresult_s4, gof_s4] = createFit_s4(dataStruct_final(k).t
    , dataStruct_final(k).E);
            Coeff_s4 = coeffvalues(fitresult_s4);
```

```
            he(k) = Coeff_s4(1);
            h2(k) = Coeff_s4(2);
    end
end

he_4 = [he(10) he(12) he(13) he(14)];
h2_4 = [h2(10) h2(12) h2(13) h2(14)];

%% Weights for he & h2 fit

w_1 = [1000 1 1];
w_2 = [1000 1 1 1];
w_3 = [1000 1 1 1];
w_4 = [1000 1 1 1];

%% Schapery fit & functions of he and h2

L = [213.6 -145.8 -42.7 -20.19];
M = [0.5730 2.609 0.8303 1.302];
O = [188.9 -126 -37.4 15.43];
P =[0.6223 2.2661 0.9224 1.665];

for k = 1:16
    if k == 8 || k == 4 || k ==3
        f_h2(k) = L(1)*epsilon(k)+M(1);
        f_he(k) = O(1)*epsilon(k)+P(1);
        dataStruct_final(k).E_schapery = epsilon(k)*(f_he(k)*Ee(1) +
   f_h2(k)*G(1)*(dataStruct_final(k).t.^(n(1))));
    elseif k == 7 || k == 2 || k == 9 || k ==15
        f_h2(k) = L(2)*epsilon(k)+M(2);
        f_he(k) = O(2)*epsilon(k)+P(2);
        dataStruct_final(k).E_schapery = epsilon(k)*(f_he(k)*Ee(2) +
   f_h2(k)*G(2)*(dataStruct_final(k).t.^(n(2))));
    elseif k == 11 || k == 1 || k == 6 || k ==16
        f_h2(k) = L(3)*epsilon(k)+M(3);
        f_he(k) = O(3)*epsilon(k)+P(3);
        dataStruct_final(k).E_schapery = epsilon(k)*(f_he(k)*Ee(3) +
   f_h2(k)*G(3)*(dataStruct_final(k).t.^(n(3))));
    else
        f_h2(k) = L(4)*epsilon(k)+M(4);
        f_he(k) = O(4)*epsilon(k)+P(4);
        dataStruct_final(k).E_schapery = epsilon(k)*(f_he(k)*Ee(4)+
   f_h2(k)*G(4)*(dataStruct_final(k).t.^(n(4))));
    end
end

```

```
f_he_1 = [f_he(3) f_he(4) f_he(8)];
f_h2_1 = [f_h2(3) f_h2(4) f_h2(8)];
f_he_2 = [f_he(7) f_he(2) f_he(9) f_he(15)];
f_h2_2 = [f_h2(7) f_h2(2) f_h2(9) f_h2(15)];
f_he_3 = [f_he(11) f_he(1) f_he(6) f_he(16)];
f_h2_3 = [f_h2(11) f_h2(1) f_h2(6) f_h2(16)];
f_he_4 = [f_he(10) f_he(12) f_he(13) f_he(14)];
f_h2_4 = [f_h2(10) f_h2(12) f_h2(13) f_h2(14)];


 %% Plot experimental data with Schapery Model - BMD group 1

figure()
hold on
scatter(dataStruct_final(3).t,dataStruct_final(3).Stress)
hold on
plot(dataStruct_final(3).t,dataStruct_final(3).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(4).t,dataStruct_final(4).Stress)
hold on
plot(dataStruct_final(4).t,dataStruct_final(4).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(8).t,dataStruct_final(8).Stress)
hold on
plot(dataStruct_final(8).t,dataStruct_final(8).E_schapery,'LineWidth'
    ,2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 1, 0.2% strain','BMD - 1, 0.2% strain(fit)','BMD - 1,
    0.55% strain','BMD - 1, 0.55% strain(fit)','BMD - 1, 0.67% strain','
    BMD - 1, 0.67% strain(fit)')
 ylabel('Stress (MPa)');
 xlabel('Time (S)');

newcolors =   [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)
```

```
%% Plot experimental data with Schapery Model - BMD group 2

figure()
scatter(dataStruct_final(2).t,dataStruct_final(2).Stress)
hold on
plot(dataStruct_final(2).t,dataStruct_final(2).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(15).t,dataStruct_final(15).Stress)
hold on
plot(dataStruct_final(15).t,dataStruct_final(15).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(7).t,dataStruct_final(7).Stress)
hold on
plot(dataStruct_final(7).t,dataStruct_final(7).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(9).t,dataStruct_final(9).Stress)
hold on
plot(dataStruct_final(9).t,dataStruct_final(9).E_schapery,'LineWidth'
    ,2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 2, 0.18% strain','BMD - 2, 0.18% strain(fit)','BMD - 2,
    0.28% strain','BMD - 2, 0.28% strain(fit)','BMD - 2, 0.32% strain','
    BMD - 2, 0.32% strain(fit)','BMD - 2, 0.84% strain','BMD - 2, 0.84%
    strain(fit)')
 ylabel('Stress (MPa)');
 xlabel('Time (S)');

  newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

%% Plot experimental data with Schapery Model - BMD group 3

```

```
figure()
scatter(dataStruct_final(6).t,dataStruct_final(6).Stress)
hold on
plot(dataStruct_final(6).t,dataStruct_final(6).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(11).t,dataStruct_final(11).Stress)
hold on
plot(dataStruct_final(11).t,dataStruct_final(11).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(1).t,dataStruct_final(1).Stress)
hold on
plot(dataStruct_final(1).t,dataStruct_final(1).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(16).t,dataStruct_final(16).Stress)
hold on
plot(dataStruct_final(16).t,dataStruct_final(16).E_schapery,'LineWidth'
    ,2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 3, 0.19% strain','BMD - 3, 0.19% strain(fit)','BMD - 3,
    0.21% strain','BMD - 3, 0.21% strain(fit)','BMD - 3, 0.26% strain','
    BMD - 3, 0.26% strain(fit)','BMD - 3, 0.81% strain','BMD - 3, 0.81%
    strain(fit)')
 ylabel('Stress (MPa)');
 xlabel('Time (S)');

   newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

 %% Plot experimental data with Schapery Model - BMD group 4

figure()
scatter(dataStruct_final(14).t,dataStruct_final(14).Stress)
hold on
```

```
plot(dataStruct_final(14).t,dataStruct_final(14).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(10).t,dataStruct_final(10).Stress)
hold on
plot(dataStruct_final(10).t,dataStruct_final(10).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(12).t,dataStruct_final(12).Stress)
hold on
plot(dataStruct_final(12).t,dataStruct_final(12).E_schapery,'LineWidth'
    ,2)
hold on
scatter(dataStruct_final(13).t,dataStruct_final(13).Stress)
hold on
plot(dataStruct_final(13).t,dataStruct_final(13).E_schapery,'LineWidth'
    ,2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 4, 0.16% strain','BMD - 4, 0.16% strain(fit)','BMD - 4,
    0.38% strain','BMD - 4, 0.38% strain(fit)','BMD - 4, 0.41% strain','
    BMD - 4, 0.41% strain(fit)','BMD - 4, 0.72% strain','BMD - 4, 0.72%
    strain(fit)')
 ylabel('Stress (MPa)');
 xlabel('Time (S)');

   newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

%% Calculate relevant values between fits

% Error Determination

error = struct('y00',[],'y11',[],'dy1',[],'abs_dy1',[],'relerr1',[],'
    pererr1',[]);
for k = 1:16

error(k).y00 = dataStruct_final(k).Stress;
```

```
error(k).y11 = dataStruct_final(k).E_schapery;
error(k).dy1 = error(k).y00- error(k).y11 ; % error
error(k).abs_dy1 = abs(error(k).dy1) ;   % absolute error
error(k).relerr1 = error(k).abs_dy1./ y00 ;  % relative error
error(k).pererr1 = error(k).relerr1*100 ;   % percentage error
mean_err_SR1(k) = mean(abs(error(k).y00 - error(k).y11)) ;    % mean
    absolute error
MSE_SR1(k) = mean((error(k).y00 - error(k).y11).^2) ;        % Mean
    square error
RMSE_SR1(k) = sqrt(mean((error(k).y00 - error(k).y11).^2)) ; % Root
    mean square error

mean_RMSE1 = mean(RMSE_SR1); % Mean Root Mean Square error
mean_pererr1(k) = mean(error(k).pererr1(end));
mean_perc = mean(mean_pererr1);
end
%% Correlation between he,h2 & strain

% plot he & h2 for BMD group 1

figure()
hold on
scatter(epsilon1, he_1,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7
    0.7]);
hold on
plot(epsilon1, f_he_1,'b');
hold on
scatter(epsilon1, h2_1,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold on
plot(epsilon1, f_h2_1,'r');
hold off
title('BMD group 1')
 legend('he - actual data','he - fit','h2 - actual data','h2 - fit')
 ylabel('he & h2');
 xlabel('Strain');

%% plot he & h2 for BMD group 2

figure()
hold on
scatter(epsilon2, he_2,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7
    0.7]);
hold on
plot(epsilon2, f_he_2,'b');
hold on
```

```
scatter(epsilon2, h2_2,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold on
plot(epsilon2, f_h2_2,'r');
hold off
title('BMD group 2')
 legend('he - actual data','he - fit','h2 - actual data','h2 - fit')
 ylabel('he & h2');
 xlabel('Strain');

 %% plot he & h2 for BMD group 3

figure()
hold on
scatter(epsilon3, he_3,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7
    0.7]);
hold on
plot(epsilon3, f_he_3,'b');
hold on
scatter(epsilon3, h2_3,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold on
plot(epsilon3, f_h2_3,'r');
hold off
title('BMD group 3')
 legend('he - actual data','he - fit','h2 - actual data','h2 - fit')
 ylabel('he & h2');
 xlabel('Strain');

 %% plot he & h2 for BMD group 4

figure()
hold on
scatter(epsilon4, he_4,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7
    0.7]);
hold on
plot(epsilon4, f_he_4,'b');
hold on
scatter(epsilon4, h2_4,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold on
plot(epsilon4, f_h2_4,'r');
hold off
title('BMD group 4')
 legend('he - actual data','he - fit','h2 - actual data','h2 - fit')
 ylabel('he & h2');
```

```
 xlabel('Strain');

%% Overall he & h2 to predict correlation

figure()
hold on
    scatter(epsilon,he,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7
    0.7]);
hold on
    scatter(epsilon,h2,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold off
legend('he','h2')
ylabel('he & h2');
xlabel('Strain')

%% Correlation between G, n & BMD

for k=1:4
f_g(k) = 3.795*BMD_r(k) - 1982;
f_n(k) = 0.0005311*BMD_r(k) - 0.147;
end

 %% plot G & n vs BMD

figure()
hold on
scatter(BMD_r, G,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_g);
hold off
 title('C vs BMD')
 legend('C','C - fit')
 ylabel('C');
 xlabel('Reference BMD');

 figure()
hold on
scatter(BMD_r, n,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_n);
hold off
 title('n vs BMD')
 legend('n','n - fit')
 ylabel('n');
 xlabel('Reference BMD');
```

```
%% Correlation between he,h2 & BMD

% Overall he & h2 to predict correlation

figure()
hold on
    scatter(BMD,he,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7])
    ;
    hold on
    scatter(BMD,h2,'MarkerEdgeColor','r','MarkerFaceColor',[0.8500
    0.3250 0.0980]);
hold off
legend('he','h2')
ylabel('he & h2');
xlabel('BMD')
```

Appendix C

Modified Superposition Method

```
% Run the MTSdatafitter_durationofstressrelaxation first

%% Log transformation & plot

BMD = [421.2866 331.5744 415.2768 483.8211 399.8866 227.2884 464.8261
    331.5744 431.4026 558.237 300.4444 445.4421 375.8612 344.386
    416.2005 304.0382];
BMD(5) = [];
BMD1 = [300.4444 304.0382 331.5744];
BMD2 = [331.7045 344.386 375.8612 399.8866];
BMD3 = [415.2768 416.2005 421.2866 431.4026];
BMD4 = [445.4421 464.8261 483.8211 558.237];

BMD_r(1) = mean(BMD1);
BMD_r(2) = mean(BMD2);
BMD_r(3) = mean(BMD3);
BMD_r(4) = mean(BMD4);

epsilon = [0.0026 0.0018 0.0020 0.0055 0.0020 0.0019 0.0032 0.0067
    0.0084 0.0038 0.0021 0.0041 0.0072 0.0016 0.0028 0.0081];
epsilon_r = [0.002 0.0018 0.0019 0.0016]; % reference strain level
epsilon1 = [0.002 0.0055 0.0067];
epsilon2 = [0.0032 0.0018 0.0084 0.0028];
epsilon3 = [0.0021 0.0026 0.0019 0.0081];
epsilon4 = [0.0038 0.0041 0.0072 0.0016];

dataStruct_final = struct('t',[],'E',[],'Stress',[],'log_t',[],'log_E'
    ,[],'log_stress',[],'E_MSM',[]);

for k = 1:size(dataStruct,2)
    dataStruct_final(k).t = dataStruct_ROI_half_hr(k).TimeROI_half_hr -
     dataStruct_ROI_half_hr(k).TimeROI_half_hr(1);
    dataStruct_final(k).t(1) = 0.1;
    dataStruct_final(k).Stress = dataStruct_ROI_half_hr(k).
    StressSample_half_hr;
    dataStruct_final(k).E = dataStruct_final(k).Stress./(epsilon(k));
    dataStruct_final(k).log_t = log10(dataStruct_final(k).t);
    dataStruct_final(k).log_E = log10(dataStruct_final(k).E);
    dataStruct_final(k).log_stress = log10(dataStruct_final(k).Stress);
end

%% Slope of Stress Vs time graph
```

```

coefficients_slope = struct('Coefficients_slope',[]);
for k=1:16
coefficients_slope(k).Coefficients_slope = polyfit(dataStruct_final(k).
    log_t, dataStruct_final(k).log_stress, 1);
slope(k) = coefficients_slope(k).Coefficients_slope(1);
end

%% BMD - group 1

figure
% hold on
%         scatter(dataStruct_final(5).log_t, dataStruct_final(5).
    log_stress)
hold on
       scatter(dataStruct_final(3).log_t, dataStruct_final(3).
    log_stress)
hold on
       scatter(dataStruct_final(8).log_t, dataStruct_final(8).
    log_stress)
hold on
       scatter(dataStruct_final(4).log_t, dataStruct_final(4).
    log_stress)
hold off
    legend('BMD - 1, 0.2% strain','BMD - 1, 0.67% strain','BMD - 1,
    0.55% strain')
    ylabel('log stress');
    xlabel('log time');

%% BMD group 1 - slope

slope1 = [slope(3) slope(4) slope(8)];

%% Polynomial Fit

[fitresult_1, gof_1] = createFit_poly(epsilon1, slope1);

Coefficients_poly1 = coeffvalues(fitresult_1);
D(1) = Coefficients_poly1(1);
F(1) = Coefficients_poly1(2);

Stress_fit1 = []; % B(epsilon)

 Stress_fit1(3) = D(1)*(epsilon1(1)) + F(1);
 Stress_fit1(4) = D(1)*(epsilon1(2)) + F(1);
 Stress_fit1(8) = D(1)*(epsilon1(3)) + F(1);
```

```

%% Initial Stress Relaxation Modulus

E1(1) = dataStruct_final(3).E(1);
E1(2) = dataStruct_final(4).E(1);
E1(3) = dataStruct_final(8).E(1);

%% Linear Fit

[fitresult1, gof1] = createFit_linear(epsilon1, E1);
Coefficients_linear1 = coeffvalues(fitresult1);
H(1) = Coefficients_linear1(1);
J(1) = Coefficients_linear1(2);

E1_fit = []; % A(epsilon)

E1_fit(3) = H(1)*(epsilon1(1)) + J(1);
E1_fit(4) = H(1)*(epsilon1(2)) + J(1);
E1_fit(8) = H(1)*(epsilon1(3)) + J(1);

%% BMD - group 2

figure
hold on
        scatter(dataStruct_final(7).log_t, dataStruct_final(7).
    log_stress)
hold on
        scatter(dataStruct_final(2).log_t, dataStruct_final(2).
    log_stress)
hold on
        scatter(dataStruct_final(9).log_t, dataStruct_final(9).
    log_stress)
hold on
        scatter(dataStruct_final(15).log_t, dataStruct_final(15).
    log_stress)
hold off
    legend('BMD - 2, 0.32% strain','BMD - 2, 0.18% strain','BMD - 2,
    0.84% strain','BMD - 2, 0.28% strain')
    ylabel('log stress');
    xlabel('log time');

    %% BMD group 2 - slope

slope2 = [slope(7) slope(2) slope(9) slope(15)];

%% Polynomial Fit
```

```
[fitresult_2, gof_2] = createFit_poly(epsilon2, slope2);

Coefficients_poly2 = coeffvalues(fitresult_2);
D(2) = Coefficients_poly2(1);
F(2) = Coefficients_poly2(2);

Stress_fit1(7) = D(2)*(epsilon2(1)) + F(2);
Stress_fit1(2) = D(2)*(epsilon2(2)) + F(2);
Stress_fit1(9) = D(2)*(epsilon2(3)) + F(2);
Stress_fit1(15) = D(2)*(epsilon2(4)) + F(2);
%% Initial Stress Relaxation Modulus

E2(1) = dataStruct_final(7).E(1);
E2(2) = dataStruct_final(2).E(1);
E2(3) = dataStruct_final(9).E(1);
E2(4) = dataStruct_final(15).E(1);

%% Linear Fit
[fitresult2, gof2] = createFit_linear(epsilon2, E2);
Coefficients_linear2 = coeffvalues(fitresult2);
H(2) = Coefficients_linear2(1);
J(2) = Coefficients_linear2(2);

E1_fit(7) = H(2)*(epsilon2(1)) + J(2);
E1_fit(2) = H(2)*(epsilon2(2)) + J(2);
E1_fit(9) = H(2)*(epsilon2(3)) + J(2);
E1_fit(15) = H(2)*(epsilon2(4)) + J(2);

%% BMD - group 3

figure
hold on
       scatter(dataStruct_final(11).log_t, dataStruct_final(11).
    log_stress)
hold on
       scatter(dataStruct_final(1).log_t, dataStruct_final(1).
    log_stress)
hold on
       scatter(dataStruct_final(6).log_t, dataStruct_final(6).
    log_stress)
hold on
       scatter(dataStruct_final(16).log_t, dataStruct_final(16).
    log_stress)
hold off
    legend('BMD - 3, 0.21% strain','BMD - 3, 0.26% strain','BMD - 3,
    0.19% strain','BMD - 3, 0.81% strain')
```

```
    ylabel('log stress');
    xlabel('log time');

%% BMD group 3 - slope

slope3 = [slope(11) slope(1) slope(6) slope(16)];

%% Polynomial Fit
[fitresult_3, gof_3] = createFit_poly(epsilon3, slope3);

Coefficients_poly3 = coeffvalues(fitresult_3);
D(3) = Coefficients_poly3(1);
F(3) = Coefficients_poly3(2);

Stress_fit1(11) = D(3)*(epsilon3(1)) + F(3);
Stress_fit1(1) = D(3)*(epsilon3(2)) + F(3);
Stress_fit1(6) = D(3)*(epsilon3(3)) + F(3);
Stress_fit1(16) = D(3)*(epsilon3(4)) + F(3);

%% Initial Stress Relaxation Modulus

E3(1) = dataStruct_final(11).E(1);
E3(2) = dataStruct_final(1).E(1);
E3(3) = dataStruct_final(6).E(1);
E3(4) = dataStruct_final(16).E(1);

%% Linear Fit
[fitresult3, gof3] = createFit_linear(epsilon3, E3);
Coefficients_linear3 = coeffvalues(fitresult3);
H(3) = Coefficients_linear3(1);
J(3) = Coefficients_linear3(2);

E1_fit(11) = H(3)*(epsilon3(1)) + J(3);
E1_fit(1) = H(3)*(epsilon3(2)) + J(3);
E1_fit(6) = H(3)*(epsilon3(3)) + J(3);
E1_fit(16) = H(3)*(epsilon3(4)) + J(3);

%% BMD - group 4

figure
hold on
        scatter(dataStruct_final(10).log_t, dataStruct_final(10).
    log_stress)
hold on
        scatter(dataStruct_final(12).log_t, dataStruct_final(12).
    log_stress)
```

```
hold on
    scatter(dataStruct_final(13).log_t, dataStruct_final(13).
  log_stress)
hold on
    scatter(dataStruct_final(14).log_t, dataStruct_final(14).
  log_stress)
hold off
    legend('BMD - 4, 0.38% strain','BMD - 4, 0.41% strain','BMD - 4,
  0.72% strain','BMD - 4, 0.16% strain')
    ylabel('log stress');
    xlabel('log time');

%% BMD group 4 - slope


slope4 = [slope(10) slope(12) slope(13) slope(14)];

%% Polynomial Fit
[fitresult_4, gof_4] = createFit_poly(epsilon4, slope4);

Coefficients_poly4 = coeffvalues(fitresult_4);
D(4) = Coefficients_poly4(1);
F(4) = Coefficients_poly4(2);

Stress_fit1(10) = D(4)*(epsilon4(1)) + F(4);
Stress_fit1(12) = D(4)*(epsilon4(2)) + F(4);
Stress_fit1(13) = D(4)*(epsilon4(3)) + F(4);
Stress_fit1(14) = D(4)*(epsilon4(4)) + F(4);

%% Initial Stress Relaxation Modulus

E4(1) = dataStruct_final(10).E(1);
E4(2) = dataStruct_final(12).E(1);
E4(3) = dataStruct_final(13).E(1);
E4(4) = dataStruct_final(14).E(1);

%% Linear Fit
[fitresult4, gof4] = createFit_linear(epsilon4, E4);
Coefficients_linear4 = coeffvalues(fitresult4);
H(4) = Coefficients_linear4(1);
J(4) = Coefficients_linear4(2);

E1_fit(10) = H(4)*(epsilon4(1)) + J(4);
E1_fit(12) = H(4)*(epsilon4(2)) + J(4);
E1_fit(13) = H(4)*(epsilon4(3)) + J(4);
E1_fit(14) = H(4)*(epsilon4(4)) + J(4);
```

```
%% Modified Superposition Method

for k = 1:16
    dataStruct_final(k).E_MSM = E1_fit(k)*epsilon(k)*((dataStruct_final
   (k).t).^Stress_fit1(k));
end

    %% BMD group 1 plot

figure()
scatter(dataStruct_final(3).t,dataStruct_final(3).Stress)
hold on
plot(dataStruct_final(3).t,dataStruct_final(3).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(4).t,dataStruct_final(4).Stress)
hold on
plot(dataStruct_final(4).t,dataStruct_final(4).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(8).t,dataStruct_final(8).Stress)
hold on
plot(dataStruct_final(8).t,dataStruct_final(8).E_MSM,'LineWidth',2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 1, 0.2% strain','BMD - 1, 0.2% strain(fit)','BMD - 1,
   0.55% strain','BMD - 1, 0.55% strain(fit)','BMD - 1, 0.67% strain','
   BMD - 1, 0.67% strain(fit)')
 ylabel('Stress');
 xlabel('Time');

   newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

 %% BMD group 2 plot

 figure()
scatter(dataStruct_final(2).t,dataStruct_final(2).Stress)
hold on
```

```
plot(dataStruct_final(2).t,dataStruct_final(2).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(15).t,dataStruct_final(15).Stress)
hold on
plot(dataStruct_final(15).t,dataStruct_final(15).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(7).t,dataStruct_final(7).Stress)
hold on
plot(dataStruct_final(7).t,dataStruct_final(7).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(9).t,dataStruct_final(9).Stress)
hold on
plot(dataStruct_final(9).t,dataStruct_final(9).E_MSM,'LineWidth',2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 2, 0.18% strain','BMD - 2, 0.18% strain(fit)','BMD - 2,
    0.28% strain','BMD - 2, 0.28% strain(fit)','BMD - 2, 0.32% strain','
    BMD - 2, 0.32% strain(fit)','BMD - 2, 0.84% strain','BMD - 2, 0.84%
    strain(fit)')
 ylabel('Stress');
 xlabel('Time');

   newcolors = [0.635 0.078 0.184
                0.635 0.078 0.184
                1.00 0.54 0.00
                1.00 0.54 0.00
                0.47 0.25 0.80
                0.47 0.25 0.80
                0.25 0.80 0.54
                0.25 0.80 0.54];
colororder(newcolors)

 %% BMD group 3 plot

 figure()
 scatter(dataStruct_final(6).t,dataStruct_final(6).Stress)
hold on
plot(dataStruct_final(6).t,dataStruct_final(6).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(11).t,dataStruct_final(11).Stress)
hold on
plot(dataStruct_final(11).t,dataStruct_final(11).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(1).t,dataStruct_final(1).Stress)
hold on
```

```
plot(dataStruct_final(1).t,dataStruct_final(1).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(16).t,dataStruct_final(16).Stress)
hold on
plot(dataStruct_final(16).t,dataStruct_final(16).E_MSM,'LineWidth',2)
hold off
ylim([0 8])
xlim([0 2000])
 legend('BMD - 3, 0.19% strain','BMD - 3, 0.19% strain(fit)','BMD - 3, 0.21% strain','BMD - 3, 0.21% strain(fit)','BMD - 3, 0.26% strain','BMD - 3, 0.26% strain(fit)','BMD - 3, 0.81% strain','BMD - 3, 0.81% strain(fit)')
 ylabel('Stress');
 xlabel('Time');

   newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

 %% BMD group 4 plot

 figure()
scatter(dataStruct_final(14).t,dataStruct_final(14).Stress)
hold on
plot(dataStruct_final(14).t,dataStruct_final(14).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(10).t,dataStruct_final(10).Stress)
hold on
plot(dataStruct_final(10).t,dataStruct_final(10).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(12).t,dataStruct_final(12).Stress)
hold on
plot(dataStruct_final(12).t,dataStruct_final(12).E_MSM,'LineWidth',2)
hold on
scatter(dataStruct_final(13).t,dataStruct_final(13).Stress)
hold on
plot(dataStruct_final(13).t,dataStruct_final(13).E_MSM,'LineWidth',2)
hold off
ylim([0 8])
xlim([0 2000])
```

```
 legend('BMD - 4, 0.16% strain','BMD - 4, 0.16% strain(fit)','BMD - 4, 0.38% strain','BMD - 4, 0.38% strain(fit)','BMD - 4, 0.41% strain','BMD - 4, 0.41% strain(fit)','BMD - 4, 0.72% strain','BMD - 4, 0.72% strain(fit)')
 ylabel('Stress');
 xlabel('Time');

   newcolors = [0.635 0.078 0.184
               0.635 0.078 0.184
               1.00 0.54 0.00
               1.00 0.54 0.00
               0.47 0.25 0.80
               0.47 0.25 0.80
               0.25 0.80 0.54
               0.25 0.80 0.54];
colororder(newcolors)

 %% Calculate relevant values between fits

% Error Determination

error = struct('y00',[],'y11',[],'dy1',[],'abs_dy1',[],'relerr1',[],'pererr1',[]);
for k = 1:16

error(k).y00 = dataStruct_final(k).Stress;
error(k).y11 = dataStruct_final(k).E_MSM;
error(k).dy1 = error(k).y00- error(k).y11 ; % error
error(k).abs_dy1 = abs(error(k).dy1) ;   % absolute error
error(k).relerr1 = error(k).abs_dy1./ y00 ;  % relative error
error(k).pererr1 = error(k).relerr1*100 ;   % percentage error
mean_err_SR1(k) = mean(abs(error(k).y00 - error(k).y11)) ;    % mean absolute error
MSE_SR1(k) = mean((error(k).y00 - error(k).y11).^2) ;         % Mean square error
RMSE_SR1(k) = sqrt(mean((error(k).y00 - error(k).y11).^2)) ; % Root mean square error

mean_RMSE1 = mean(RMSE_SR1); % Mean Root Mean Square error
mean_pererr1(k) = mean(error(k).pererr1(end));
mean_perc = mean(mean_pererr1);
end

 %% Correlation between H, J, D, F and BMD

for k = 1:4
```

```
    f_H(k) =  -140*BMD_r(k) + 59330;
    f_J(k) =  1.788*BMD_r(k) + 18.48;
    f_D(k) = 0.01165*BMD_r(k) - 2.422;
    f_F(k) = -0.0001626*BMD_r(k) - 0.004826;
end

%% plot H, J, D & F vs BMD

figure()
hold on
scatter(BMD_r, H,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_H,'b');
hold off
 title('H vs BMD')
 legend('H','H - fit')
 ylabel('H');
 xlabel('Reference BMD');

 figure()
hold on
scatter(BMD_r, J,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_J,'b');
hold off
 title('J vs BMD')
 legend('J','J - fit')
 ylabel('J');
 xlabel('Reference BMD');

 figure()
hold on
scatter(BMD_r, D,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_D,'b');
hold off
 title('D vs BMD')
 legend('D','D - fit')
 ylabel('D');
 xlabel('Reference BMD');

 figure()
hold on
scatter(BMD_r, F,'MarkerEdgeColor','b','MarkerFaceColor',[0 0.7 0.7]);
hold on
plot(BMD_r, f_F,'b');
```

```
hold off
 title('F vs BMD')
 legend('F','F - fit')
 ylabel('F');
 xlabel('Reference BMD');
```