A Non-linear Visco-elastic Model for Dynamic Finite Element Simulation of Bovine Cortical Bone

This work is submitted in partial fulfilment of the requirements of the degree of Master of Science in Engineering specialising in Mechanical Engineering

Presented by:
Caitlyn Blignaut

Supervisors:
Mr. Ernesto Ismail
Mr. Trevor Cloete

April 19, 2021

Blast Impact and Survivability Research Unit
Department of Mechanical Engineering
University of Cape Town

Keywords: Cortical, bone, visco-elastic, non-linear, modelling, finite element analysis, constitutive model, intermediate strain rate

Declaration

1. I know the meaning of plagiarism and declare that all the work in the document, save for that which is properly acknowledged, is my own.
2. This dissertation has been submitted to the Turnitin module (or equivalent similarity and originality checking software) and I confirm that my supervisor has seen my report and any concerns revealed by such have been resolved with my supervisor.
3. I have used the IEEE convention for citation and referencing. Each significant contribution to, and quotation in, this report from work(s) of other people has been attributed, and has been cited and referenced.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof.

Signature:... Signed by candidate
Caitlyn Blignaut

Date: April 19, 2021

Abstract

Department of Mechanical Engineering

University of Cape Town

A Non-linear Visco-elastic Model for Dynamic Finite Element Simulation of Bovine Cortical Bone

by Caitlyn Blignaut

Modelling and simulation of the human body during an impact situation such as a car accident, can lead to better designed safety features on vehicles. In order to achieve this, investigation into the material properties and the creation of a numerical model of cortical bone is needed.

One approach to creating a material model of cortical bone suitable for these situations is to describe the material model as visco-elastic, as reported by Shim *et al.* [1], Bekker *et al.* [2] and Cloete *et al.* [3]. The work by Shim *et al.* and Bekker *et al.* developed three-dimensional models, but do not accurately capture the transition in behaviour in the intermediate strain rate region, while Cloete *et al.* developed a phenomenological model which captures the intermediate strain rate behaviour in one dimension. This work aims to verify and extend these models.

The intermediate strain rate regime ($1\,s^{-1}$ to $100\,s^{-1}$) is of particular interest because it is a key characteristic of the behaviour of cortical bone and several studies have been conducted to gather experimental data in this region [3, 4, 5, 6]. The behaviour can be captured using non-linear visco-elastic models. This dissertation focuses on the development and implementation of a material model of cortical bone based on non-linear visco-elastic models to capture the intermediate strain rate regime behaviour. The material model was developed using uni-axial test results from cortical bone. The model by Cloete *et al.* has been improved and extended, and issues of local and global strain rate with regards to the viscosity have been clarified. A hereditary integral approach was taken in the analysis and implementation of discrete models and was found to be consistent with mathematical models. The model developed was extended to three dimensions in a manner similar to that of Shim *et al.* and Bekker *et al.* for implementation in commercial finite element software (LS-Dyna and Abaqus).

Acknowledgments

I would like to thank all the people who have helped and supported me over the last 4 years. Without you the completion of this dissertation would not have been possible.

To my supervisors, Ernesto Ismail and Trevor Cloete, thank you for your guidance and input into this project, and the many "1 hour" meetings. To Trevor, for his curious mind and never ending desire to get down to the "nitty-gritty" of a problem. To Ernesto, for his extensive knowledge and expertise on all things numerical and visco-elastic, and in particular for sacrificing many evenings editing my drafts and really being the driving force behind me submitting this year.

To Professor Nurick, who always took the time to find out how I was doing, and did much campaigning for funding on my behalf in the background. Special mention needs to be made to Genevieve Langdon who nominated me for a bursary without me knowing about it. Finding out while on holiday was one of the best surprises of my funding struggles. To Ernesto, again, for also providing me with funding and always advocating for me in this regard. Thank you all for being my guardian angels within the department.

To the University of Cape Town and in particular the UCT Department of Mechanical Engineering Research Fund and the UCT Research Block Grant for all funding assistance.

To all the BISRU and EEPEI students during my time on campus. The lunch table was always alive with hot debates and a place where any question could be asked. Thank you for the chats, the laughs and the camaraderie. It was unlike anything I am likely to experience again.

To my work colleagues who over the last 18 months have always been supportive and encouraging. Particular mention to my team lead Shaun Bosch, who has gone so far as to add completing my dissertation to my career goals. We can mark that as done now.

To my family, for your unwavering support from all corners of the world. To my mom, especially, thank you for your help and guidance at every step. Without you, I would not have been able to pursue this masters. The sacrifices you have made for me and my pursuit of higher education are far more than I can ever hope to repay.

To the most patient person I know, Thomas Düsterwald, you have kept me sane and given me strength. Your love and support have meant more than you know, and the many cups of tea, coffee and wine are always much appreciated. Thank you for everything.

To my other half, Claire Lawrence, you are my rock. Thank you for the endless hours of debugging simulations and listening to me try to explain things, I think you know as much about my

dissertation as I do. I have said it before, but I'll say it again, I want to be like you when I grow up!

To my surrogate mom, Desiree Lawrence, you have always supported me, usually financially, every step of the way. I would not have been able to register in my first year without your help, and would probably not have been able to get on a plane back to Cape Town several times if it wasn't for your generous spirit. I will forever be grateful to you for enabling me to pursue this masters. Thank you.

Contents

List of Figures

List of Tables

Nomenclature

$\ddot{\boldsymbol{x}}$	Nodal acceleration vector
$\dot{\boldsymbol{x}}$	Nodal velocity vector
$\boldsymbol{\mathcal{F}}$	Internal force vector
$\boldsymbol{C}$	Damping matrix
$\boldsymbol{f}$	Applied force vector
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{M}$	Diagonal Mass matrix
$\boldsymbol{x}$	Nodal displacement vector
$\dot{\lambda}$	Stretch Rate, difference in nodal velocities
$\dot{\varepsilon}$	Strain Rate
η	Dashpot coefficient
λ	Stretch, difference in nodal displacements
σ	Stress
ε	Strain
E	Elastic modulus
k	Elastic stiffness

Chapter 1

Introduction

1.1 Background and Context

Modelling and simulation of the behaviour of the human body during impact events, such as a car accident, can improve the design of vehicle safety features. The bio-mechanical reactions of the skeleton form a major component of the overall behaviour of the body during impact situations. Investigation into the material properties of cortical bone is a key step to making representative skeletal modelling a reality in the future. This will help create more accurate numerical models which will lead to better-designed safety features and ultimately safer cars and transport. Such numerical models could also be used in designing preventative measures for sporting and other injuries resulting from impact load situations.

Biological structures are highly dependent on the organism, and on the development and regeneration of that specific structure. A long-term goal of bio-mechanical modelling is to create patient-specific models. The complexity of these models could vary greatly, from simple scaling of properties based on factors such as bone density or calcium content, to those based on medical images and scans. The models considered in this work are centred around population-representative material parameters, but in several places these factors could be refined to patient specific ones in future work.

1.2 Aims

The main aim of this research is to create a material model that captures the behaviour of cortical bovine bone. Bovine bone was chosen as there is a large amount of experimental data in the

literature to validate against, and a consistent sample set can be obtained in terms of parameters such as age. It is also convenient to source (from a butchery) and does not have as many ethical considerations as human bone specimens. While there are differences in bone composition and density between human and bovine bone [7], there are also similarities in the microstructure that would allow the model to be adapted for modelling of human cortical bone [8]. The model should capture the rate dependent behaviour and therefore the visco-elastic properties, particularly in the intermediate strain rate (ISR) regime (i.e. $1\,s^{-1}$ to $100\,s^{-1}$) and should also be implementable in commercial finite element software. An extensive amount of work has already been carried out by the Blast Impact and Survivability Research Unit (BISRU) at UCT in obtaining experimental data on the material properties of cortical bone. This data will be used to validate the material model.

1.3 Objectives

In order to achieve the aims outlined above, the following objectives were set:

- Collate mechanical material property data from previous work done within BISRU,
- Model simple linear and non-linear one-dimensional models,
- Develop and implement three-dimensional visco-elastic models extended from the one-dimensional models.

1.4 Methodology

The development and deployment of the new material model proposed followed two distinct steps. The first was the development of the model and the second was the implementation in commercial software, such as LS Dyna.

There were three phases to the development of the material model. The first was to reproduce the results of relevant visco-elastic models that have been presented in the literature. This was done to obtain an understanding of the visco-elastic theory as applied to one-dimensional models of bone. The standard visco-elastic models do not capture all the behaviour that has been determined to be of importance. Thus, the second phase was to investigate the treatment of similar behaviours in other applications. The final phase was the development of a new material model that captures the visco-elastic properties of cortical bone.

The implementation of the material model developed consists of two parts. Firstly, a verification of the model which involves reproducing one-dimensional results. Then the validation of the material model in a three-dimensional axisymmetric setting using existing experimental data.

1.5 Plan of Development

This dissertation will begin with a general overview of the literature on cortical bone and the study of its behaviour as well as the various models that have been proposed. In order to be able to understand in depth the visco-elastic models explored in this dissertation, the classic visco-elastic theory is then reworked in one dimension. This theory is then extended to three dimensions before a detailed analysis of the visco-elastic models by Cloete *et al.* , Bekker *et al.* and Shim *et al.* is explored. The implementation of one-dimensional visco-elastic models and non-linearity is then outlined with the results of these models then presented with a discussion of the results.

The material model for cortical bone is then presented and the implementation of three-dimensional models in finite element software (LS Dyna and Abaqus) is outlined. The results of the three-dimensional models are then presented followed by a discussion of these results. Future work is then discussed and final conclusions of this dissertation are outlined.

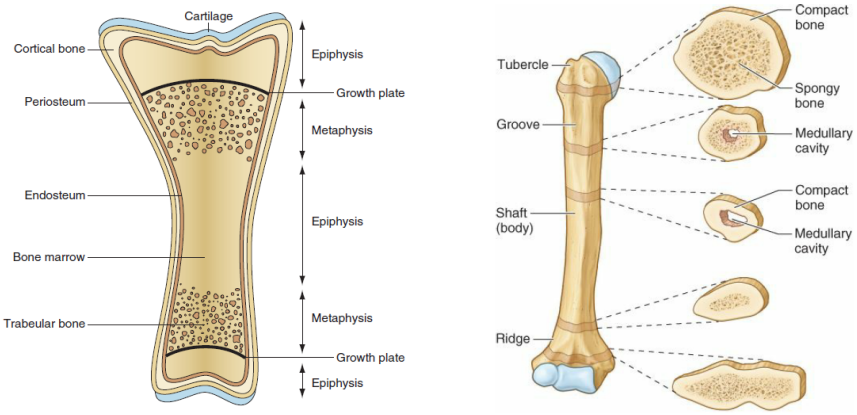
Chapter 2

High Level Literature Review

The physiology and behaviour of cortical bone has been extensively researched in the past. This chapter will outline the general structure of bones and the role that cortical bone has to play in the structure of bone, as well as the strain rate dependent behaviour of cortical bone. In addition to this, the modelling of bone in the literature will be explored and the visco-elastic bone models built upon by this dissertation will be introduced.

2.1 Physiology of Cortical Bone

Long bones consist of a hollow shaft (diaphysis) and the two ends, made up of the metaphyses and epiphyses [9, 10], shown in Figure 2.1a. Bone is a complex hierarchical biological material. There are two types of bone at the macroscopic level, cortical bone and cancellous (spongy) bone [11, 12]. All bones have a thin layer of compact cortical bone surrounding either trabecular bone or a marrow (medullary) cavity [11]. The diaphysis consists mainly of cortical bone, while the metaphyses and epiphyses contain trabecular bone enclosed by a thin layer of cortical bone [9, 10], shown in Figure 2.1b. Cortical bone is the hard outer part of the bone, usually found as a thin layer at the joints and is thickest at the mid-diaphysis of long bones [9], as this is where a long bone is most likely to buckle [11].



(a) The microstructure of long bones [10].

(b) The structure of long bone femur [11].

Figure 2.1: The structure of long bones.

Cortical bone is made up of approximately 90% mineralised tissue from which it gets most of its strength [10]. It is a dense solid and is made up of osteons, which are sheets of mineralised collagen fibre arranged in a cylindrical shape [12], also know as Haversian systems [9], shown in Figure 2.2. Cortical bone is normally formed in a lamellar pattern where collagen fibrils are laid down in alternating orientations [9]. Collagen is the main protein found in bone and the collagen fibres are densely packed which is the reason for high mechanical strength [10]. The direction of these collagen fibres is what determines the lamellar structure of the bone [10]. Lamellar collagen structures are found in a parallel orientation along the length of the bone, or in concentric orientations around blood vessels (Haversian channels) [10].

If the bone has undergone fracture and is in the process of healing, it takes on a woven structure called woven bone [10]. This will develop into lamellar bone over time [10]. Woven bone has the collagen fibrils laid down in a less organised way and is therefore weaker than lamellar bone [9].

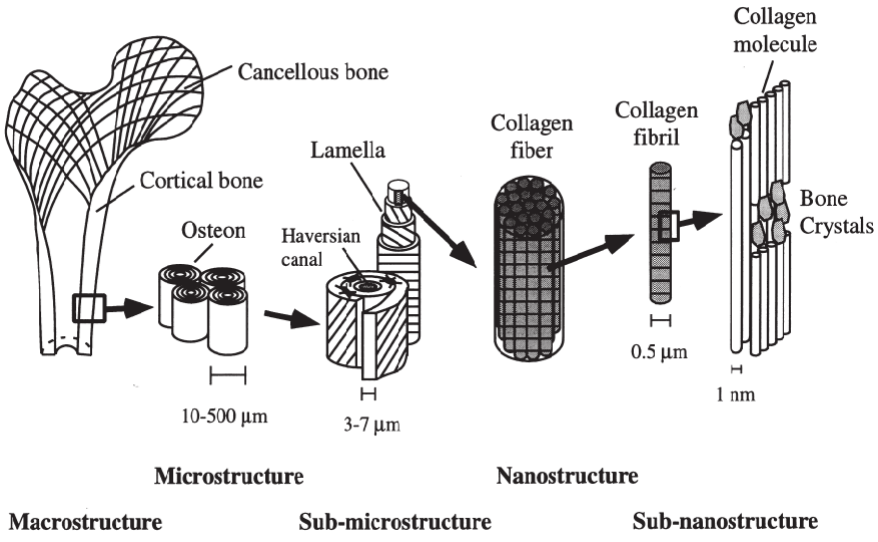


Figure 2.2: The hierarchical structure of cortical bone [12].

2.2 Effect of Strain Rate on Behaviour of Cortical Bone

Cortical bone has been found to exhibit significant strain rate dependent behaviour, particularly in the intermediate strain rate (ISR) regime (i.e. $1\,s^{-1}$ to $100\,s^{-1}$) [3, 4]. At low strain rates it exhibits elastic behaviour with a relatively low modulus, and at high strain rates it exhibits elastic behaviour with a relatively high modulus. Between these two elastic regions there is a transition regime, which is defined as the ISR regime.

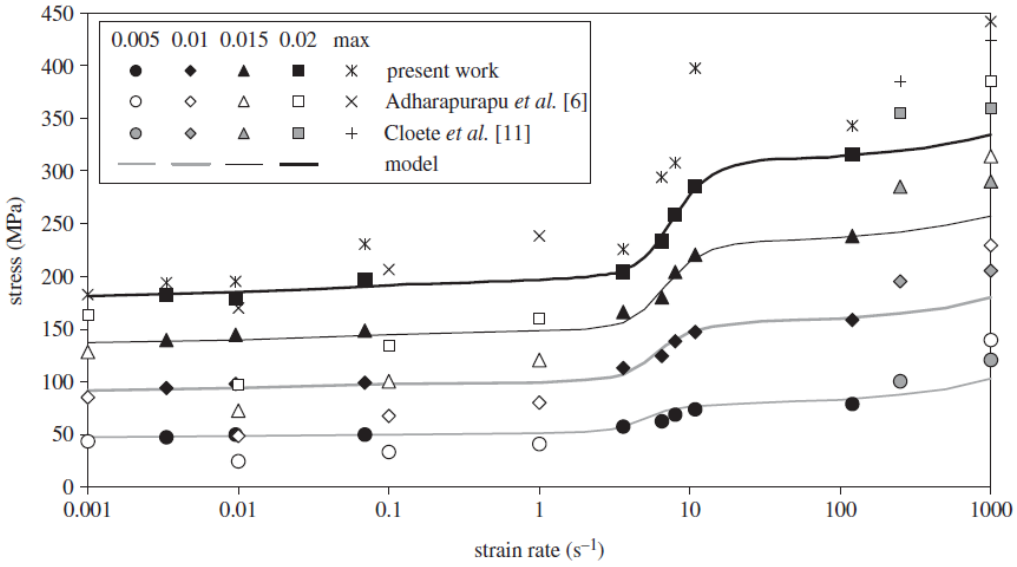


Figure 2.3: Strain rate dependent behaviour of cortical bovine bone [3].

There has been some experimental work reported in the literature to determine the behaviour of cortical bovine bone which has been collated by Cloete *et al.* [3] and is shown in Figure 2.3. This was generated from compressive tests that were conducted at constant strain rates and had a typical result such as that shown in Figure 2.4.

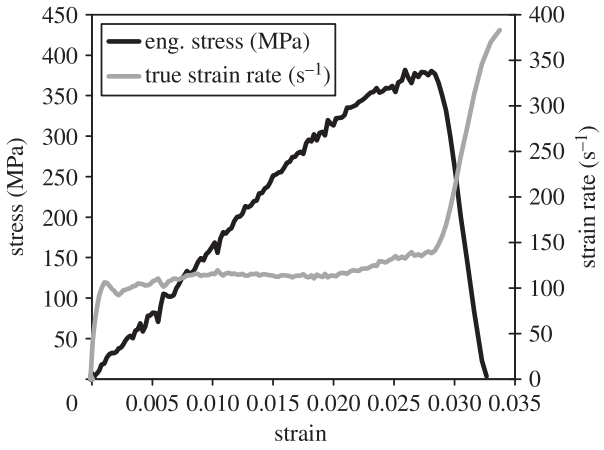


Figure 2.4: Stress - strain and strain rate - strain graphs of a constant strain rate test of cortical bovine bone conducted by Cloete *et al.* [3].

There has been significant research into the use of visco-elastic models to describe the experimentally determined material behaviour of cortical bone, as collated by Johnson *et al.* [6] and shown in Figure 2.5. However, this experimental data has typically not been found at constant strain rates. Cortical bone is a strain rate dependent material, and phenomena that appear in the data are difficult to classify if testing was not carried out at constant strain rates. The issue of constant strain rate is crucial in order to account for the strain rate effects expected from cortical bone. This will help with obtaining reliable experimental data against which to compare a visco-elastic material model.

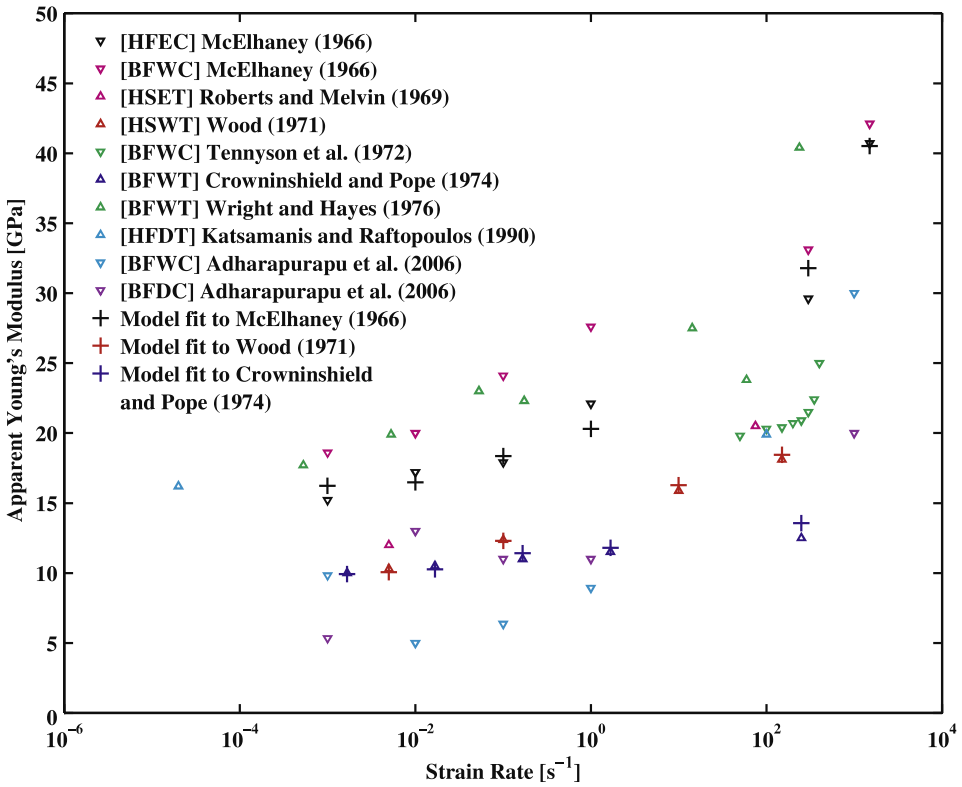


Figure 2.5: Collated data of Young's moduli for cortical bone at varying strain rates [6].

Cloete *et al.* [3] and Adharapurapu *et al.* [4] have obtained near constant strain rate testing which provides reliable data for cortical bovine bone for the purposes of this research. However, Adharapurapu *et al.* [4] have only achieved constant strain rate at high strain rates. Cloete *et al.* [3] have achieved constant strain rate testing in the ISR regime, as shown in Figure 2.4. Once the testing can be conducted at a constant strain rate, the full stress - strain rate curve can be obtained.

2.3 Modelling of Cortical Bone

Cortical bone has been observed to exhibit highly strain rate dependent behaviour [5, 6]. Visco-elastic models are used to model certain strain rate dependent behaviours and can produce both elastic and viscous behaviours similar to cortical bone. This makes visco-elastic models a suitable starting point for a material model of cortical bone. There are some differences in the composition and density of human and bovine cortical bone. Bovine bone is from a larger animal that bears weight differently to humans and grows at a faster rate [7]. In order to successfully transfer a material model from one type to the other, several factors such as age and the effect this has on the microstructure and composition will have to be considered. There are similarities, however, in the microstructural components between human and bovine cortical bone that would allow a material model for bovine cortical bone to be extended for human bone. This is mainly in the microstructural behaviour of osteons in cortical bone [8].

Several studies have been conducted into the development of a material model for bone using visco-elastic models. Tennyson *et al.* proposed a linear visco-elastic model which, under the assumption of constant stress and therefore a zero stress rate, was reduced to a Kelvin-Voigt model to capture the behaviour of cortical bone [13]. Tanabe and Kobayashi then proposed a non-linear Kelvin-Voigt model to capture the non-linearity seen in the results from their study on anisotropy [14].

Johnson *et al.* suggest that after collating all the experimental data from various studies in the literature, shown in Figure 2.5, there are two strain rate regimes of visco-elastic behaviour over the various strain rates [6]. Shim *et al.* introduced a power law equation which was fitted to their experimental results from cancellous bone specimens which would differ from those investigated in this dissertation. The parameters used in their proposed model were extracted from the best fit curves [1] and the model was then extended to three dimensions by using a function of the strain rate tensor in place of the one-dimensional strain rate terms.

Bekker *et al.* expanded on the model proposed by Shim *et al.* by raising the Kelvin-Voigt dashpot to a variable power [2]. In addition, the three-dimensional model proposed by Bekker *et al.* was implemented in a finite element analysis software and compared successfully to their experimental results. Work in both cancellous and cortical bone [1, 2] show different moduli at high and low strain rates with a transition region. Models that allow for this behaviour can likely be applied to both types. The models shown by Shim *et al.* and Bekker *et al.* allow for differing transitions between the high and low strain rate behaviours.

Cloete *et al.* [3] proposed a model similar to that of Bekker *et al.* [2] and Shim *et al.* [1] however in order to match the increased level of non-linearity seen in their experimental results, a non-linear dashpot was introduced into the Maxwell elements. This increased the gradient of the transition between the low and high strain rate responses to match that of the experimental results. The key limitations to the three-dimensional visco-elastic models put forth in the literature by Shim *et al.*

and Bekker *et al.* [1, 2], are that they are unable to consider the strain rate dependent behaviour of cortical bone in the ISR regime due to a lack of data. The work by Cloete *et al.* [3] has provided a more comprehensive view of the behaviour in the ISR regime, and more data points on which to build a material model. From these data the transition between the low strain rate and high strain rate regimes is observed to be sharper than a linear visco-elastic model can capture.

While the models proposed by Shim *et al.* and Bekker *et al.* [1, 2] are not able to capture the sharper transition response, the three-dimensional implementation approach proposed in these studies will be used going forward. Cloete *et al.* [3] presented a one-dimensional phenomenological visco-elastic model which captures the ISR regime strain rate dependent behaviour of cortical bone, observed in their experimental results. This provides a basis on which to build this research, and create a new material model that could describe the rate-sensitive response of cortical bovine bone.

Chapter 3

Review of Visco-elastic Theory

This chapter presents an overview of classic visco-elastic theory. The purpose is to lay a theoretical foundation for the discussion of the models presented by Cloete *et al.* , Bekker *et al.* and Shim *et al.* in Chapter 5. Visco-elasticity can be defined as the time-dependent behaviour of materials [15, 16] which exhibit both viscous and elastic characteristics [17]. Cortical bone has been shown to exhibit behaviour similar to spring-dashpot visco-elastic models. In order to better understand the link between observed phenomena and visco-elastic models of cortical bone, further investigation into the theory behind certain spring-dashpot models was required.

Constructed in one dimension, simple linear visco-elastic models can be represented as a combination of springs and dashpots, which appear in different arrangements to describe varying visco-elastic behaviour. Three of these arrangements will be presented in this chapter, namely the Kelvin-Voigt element, the Maxwell element and the Standard Linear Model.

3.1 Kelvin-Voigt Element (with Linear Dashpot)

The first visco-elastic element to be presented is the linear Kelvin-Voigt element, which consists of a spring, of stiffness k, and a linear dashpot, of viscosity η, in parallel [18], shown in Figure 3.1.

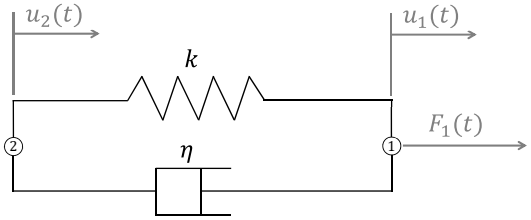


Figure 3.1: The arrangement of the Kelvin-Voigt element.

The stretch of the element λ is equal to the stretch of the spring λ_k which is equal to the stretch of the dashpot λ_η. The stretch of the element is the difference between the displacements of the ends of the element such that

$$\lambda = u_1 - u_2 \tag{3.1}$$

and the stretch rate of the element is the derivative of the stretch with respect to time

$$\dot{\lambda} = \frac{\partial \lambda_\eta}{\partial t} \tag{3.2}$$

$$= \dot{u}_1 - \dot{u}_2 \tag{3.3}$$

$$= \frac{\partial u_1}{\partial t} - \frac{\partial u_2}{\partial t}. \tag{3.4}$$

The force F, generated by the stretch of the element, is shared between the spring and the dashpot such that

$$F = F_k + F_\eta \ , \tag{3.5}$$

where F_k is the force generated by the stretch of the spring and F_η is the force generated by the stretch rate of the dashpot such that

$$F = k\lambda + \eta\dot{\lambda} \ , \tag{3.6}$$

where k is the elastic modulus and η is the dashpot coefficient.

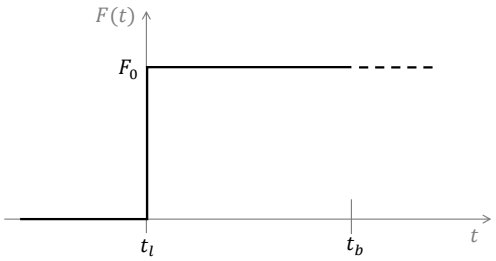
3.1.1 Kelvin-Voigt Element: Instantaneous Loading of a Force Step Function

To analyse the relaxation behaviour of a Kelvin-Voigt element, we consider the response of such an element to a suddenly applied load which is modelled as a step function, derived from the Heaviside function

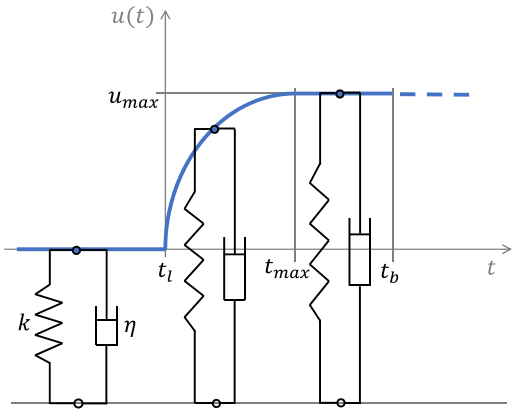
$$H(t-a) = \begin{cases} 0 & t < a \\ 1 & t \geq a \end{cases} \tag{3.7}$$

which has a magnitude of zero for values of t less than a and a value of one for values of t greater than or equal to a. The force step function $F(t)$ has a value of zero before the time of loading t_l, and a value of F_0 at or after time t_l, shown in Figure 3.2a, and can be written as

$$F(t) = F_0 H(t-t_l) = \begin{cases} 0 & t < t_l \\ F_0 & t \geq t_l \end{cases}. \tag{3.8}$$



(a) Step force function of magnitude F_0.



(b) Displacement-time graph response to the step force function (with the element diagram overlaid at varying stages to portray the physical behaviour of the element).

Figure 3.2: A step force function applied to a Kelvin-Voigt element and the displacement response over time.

The function $F(t)$ is applied to a Kelvin-Voigt element, producing the following responses, shown in Figure 3.2b

$t < t_l$: There is zero stretch in the element due to the zero force before time of loading $t = t_l$.

$t = t_l$: The element will start to move initially with a velocity of $\frac{F_0}{\eta}$ [18].

$t_l < t < t_b$: As the spring begins to stretch some of the force is transferred to the spring and the velocity is now $\frac{F_\eta}{\eta}$, where F_η is the force response of the dashpot and is decreasing. The spring and the dashpot will stretch by the same amount at a decreasing velocity while the force is being applied until a time $t = t_{max}$ when the spring is considered to have reached its maximum stretch $\lambda_{k_{max}}$, within some tolerance, such that

$$\lambda_{k_{max}} = \frac{F_0}{k} = u_{max}. \tag{3.9}$$

$t \geq t_b$: The velocity asymptotes to zero, which is when the spring is at its maximum length, and the element is at its maximum stretch λ_{max} as defined above.

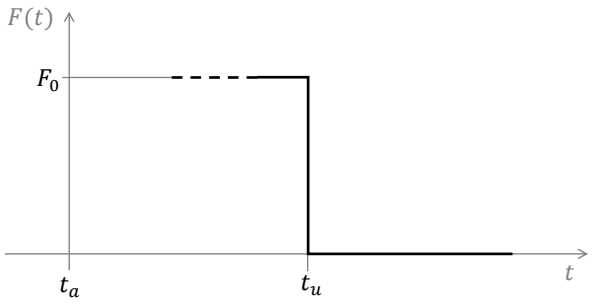
3.1.2 Kelvin-Voigt Element: Instantaneous Unloading of a Force Step Function

At time t_u, the load is then removed. The unloading force step function is derived from the negative of the Heaviside function

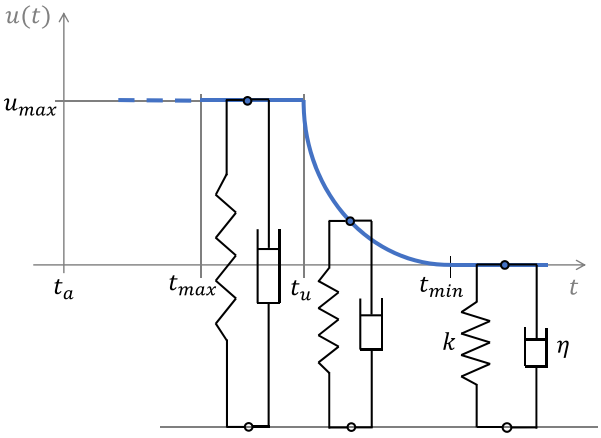
$$H(a-t) = \begin{cases} 1 & t < a \\ 0 & t \geq a \end{cases} \tag{3.10}$$

This function has a magnitude of one for values of t less than a and a value of zero for values of t greater than or equal to a. The force step function $F(t)$ has a value of F_0 before the time of unloading t_u, and a value of zero after time t_u, shown in Figure 3.3a, and can be written as

$$F(t) = F_0 H(t_u - t) = \begin{cases} F_0 & t < t_u \\ 0 & t \geq t_u \end{cases}. \tag{3.11}$$



(a) Instantaneous unloading of force function of magnitude F_0.



(b) Displacement-time graph response to the unloading force function.

Figure 3.3: Instantaneous unloading of a step force function to a Kelvin-Voigt element and the displacement response over time.

The unloading force step function, $F(t)$, is applied to the element, and the element produces the following response, shown in Figure 3.3b

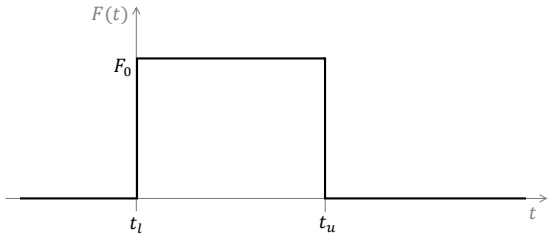
$t = t_u$: After having reached maximum stretch, the force F_0 is now removed. However, the Kelvin-Voigt element does not instantaneously recover. This is because the dashpot will resist motion of recovery [18].

$t_u < t < t_{min}$: The force of the spring is at a maximum at $t = t_u$ (due to the maximum stretch). This force acts on the dashpot as the spring contracts until the spring has returned to its original dimensions.

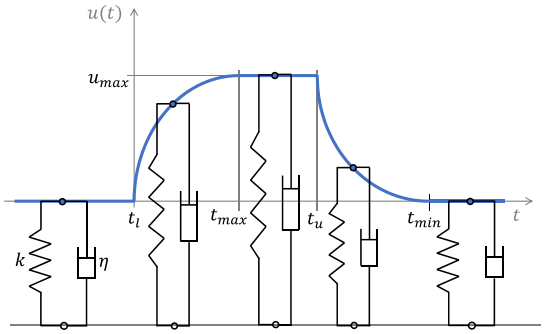
$t \geq t_{min}$: The element has returned to the original dimensions, and remains in this state since no external forces are being applied to it.

3.1.3 Kelvin-Voigt Element: Instantaneous Loading and Unloading of a Force Step Function

An overview illustration of the configuration of the Kelvin-Voigt element at each stage of loading and unloading, as well as the displacement-time graph response of the Kelvin-Voigt element to the square force pulse is shown in Figure 3.4.



(a) Square force pulse of magnitude F_0 applied for a time period of $t_u - t_l$.



(b) Displacement-time graph response to the square force pulse.

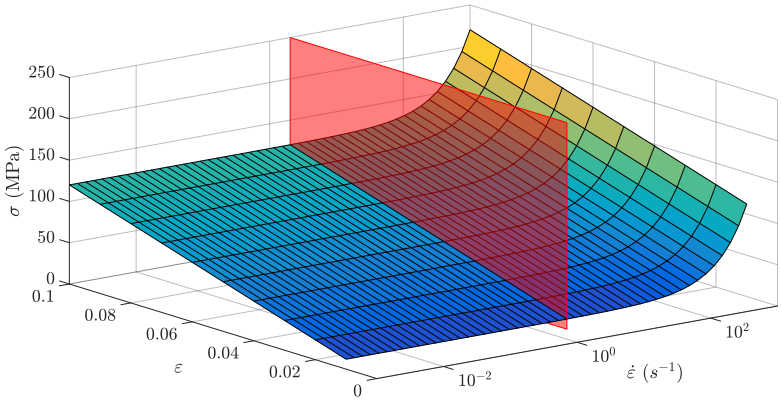
Figure 3.4: A square force pulse applied to a Kelvin-Voigt element and the displacement response over time.

3.1.4 Kelvin-Voigt Element: Instantaneous Prescribed Displacement

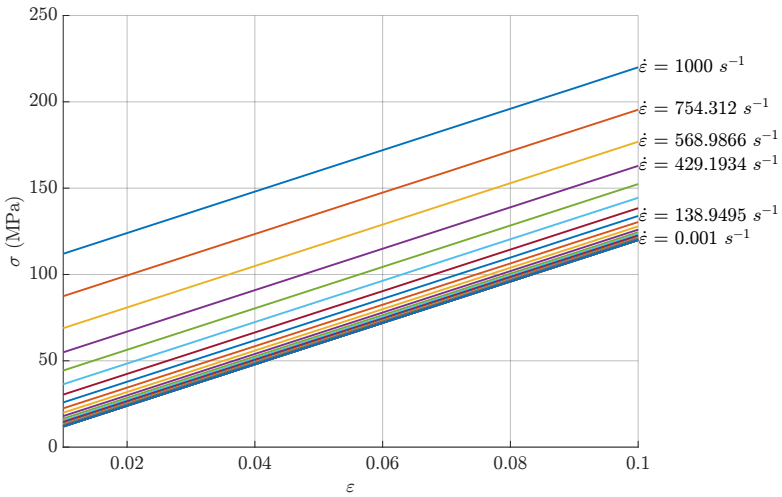
To investigate the force relaxation behaviour of a Kelvin-Voigt element, an instantaneous prescribed displacement would need to be applied. This implies the element moves to that displaced position at an infinite rate of change of displacement (or velocity). Since the force response of the dashpot is proportional to the velocity, an infinite force would have to be applied to the dashpot in order for this to happen. The displacement of the Kelvin-Voigt element is controlled by the characteristics of the dashpot and therefore it is impossible, in theory, to give the Kelvin-Voigt element an instantaneous prescribed displacement.

3.1.5 Kelvin-Voigt Element: Constant Strain Rate

The behaviour of a Kelvin-Voigt element when subjected to a range of constant strain rates was investigated by fixing node two in Figure 3.1, such that $u_2 = 0$, and moving node one at a constant velocity. Each constant strain rate test is shown in Figure 3.5b.



(a) Surface cut along stress - strain lines.

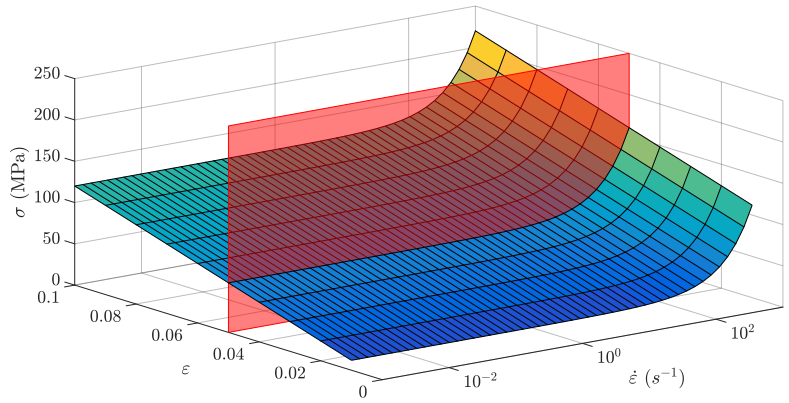


(b) Stress - strain graphs.

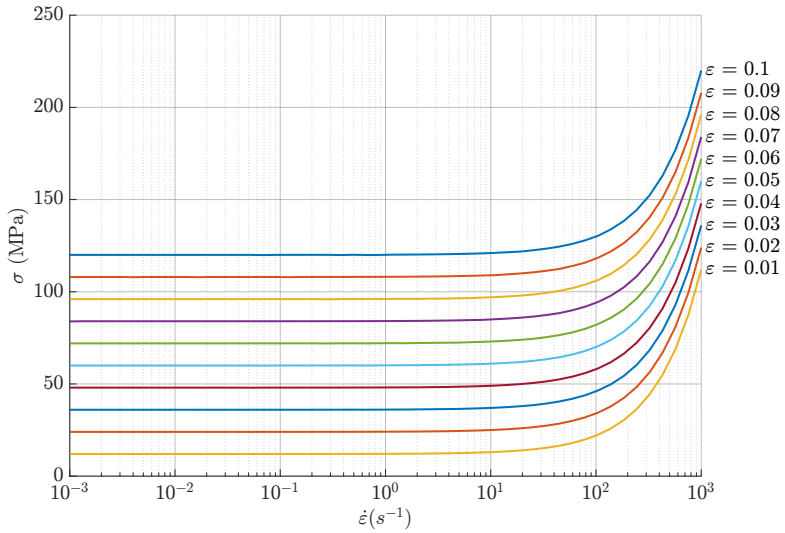
Figure 3.5: The response of a Kelvin-Voigt element showing the stress - strain graphs at varying strain rates.

These curves can be represented as a surface on a three-dimensional plot where the depth axis represents the strain rate the test was performed at. This surface plot is shown in Figure 3.5a. The lines of Figure 3.5b can be visualised as slices of the surface in Figure 3.5a along the planes of constant strain rate. By projecting the surface in Figure 3.5a onto the stress - strain rate plane, a two-dimensional plot showing the stress resulting from a certain strain at a given strain rate is obtained. The curves of the stress - strain rate result, shown in Figure 3.6b are obtained by slicing the surface in Figure 3.6a along the planes of constant strain.

These stress - strain rate curves describe the strain rate dependent behaviour of a Kelvin-Voigt element. At low strain rates, the effect of the Kelvin-Voigt spring dominates the response while the dashpot will not provide resistance to movement of node one. At higher strain rates, the dashpot will start to provide an ever increasing resistance to movement of node one. This will result in a monotonically increasing stress response of the Kelvin-Voigt element as the strain rate increases.



(a) Surface cut along stress - strain rate lines.



(b) Stress - strain rate graphs.

Figure 3.6: The response of a Kelvin-Voigt element showing the stress - strain rate graphs at varying strains.

3.2 Maxwell Element (with Linear Dashpot)

The second visco-elastic element that is fundamental in modelling rate dependent behaviour is the linear Maxwell element, which consists of a spring of stiffness k and a dashpot with coefficient η in series, shown in Figure 3.7.

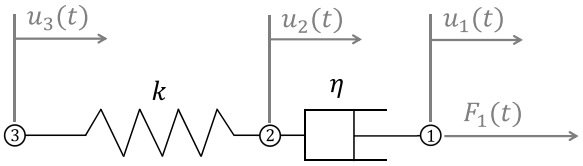


Figure 3.7: The arrangement of the Maxwell element.

The force of the element F is equal to the force caused by deformation (stretch) of the spring F_k which is equal to the force as a result of the rate of change of deformation (stretch rate) of the dashpot F_η such that

$$\begin{aligned} F &= F_k = k\lambda_k \\ &= F_\eta = \eta\dot{\lambda}_\eta. \end{aligned} \tag{3.12}$$

The stretch of the element, λ, is shared between the spring and the dashpot such that

$$\lambda = \lambda_k + \lambda_\eta \tag{3.13}$$

where λ_k is the stretch of the spring

$$\lambda_k = u_2 - u_3 \tag{3.14}$$

and λ_η is the stretch of the dashpot

$$\lambda_\eta = u_1 - u_2. \tag{3.15}$$

3.2.1 Maxwell Element: Instantaneous Loading of a Force Step Function

To analyse the relaxation behaviour of a Maxwell element, the step force function $F(t)$ defined in equation (3.8) is applied to a Maxwell element. This produces the following responses [18], shown in Figure 3.8b

$t < t_l$: There is zero stretch of the element as no force is applied before time $t = t_l$.

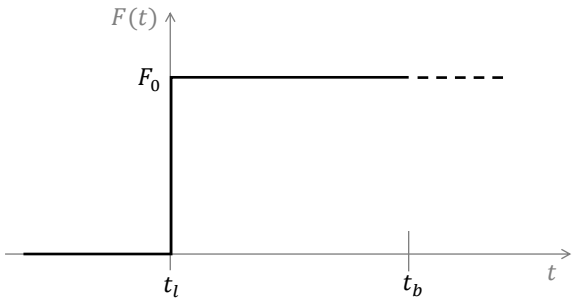
$t = t_l$: The spring will stretch by an amount λ_k, which is equal to the initial total displacement of the element u_0

$$\text{where} \qquad \lambda_k = \frac{F_0}{k} = u_0 \quad .$$

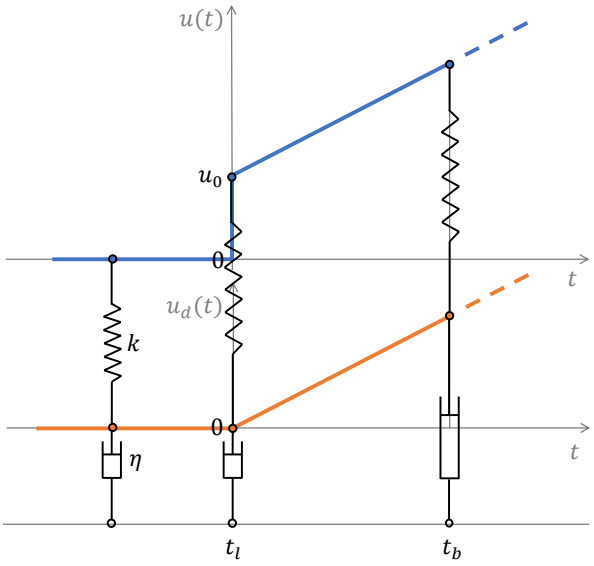
This is due to the dashpot not being able to stretch initially due to the instantaneous application of the force.

$t_l < t < t_b$: The spring will maintain its initial stretch, λ_k, however the dashpot will now stretch at a constant rate due to the constant force being applied to the element.

$t > t_b$: The spring has stretched to its maximum, and the dashpot continues to stretch.



(a) Step force function of magnitude F_0.



(b) Displacement-time graph response to the step force function.

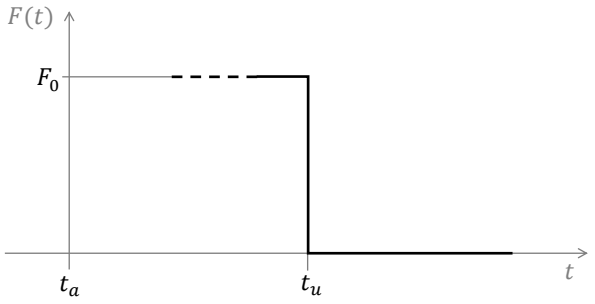
Figure 3.8: A step force function applied to a Maxwell element and the displacement response over time.

3.2.2 Maxwell Element: Instantaneous Unloading of the Constant Force

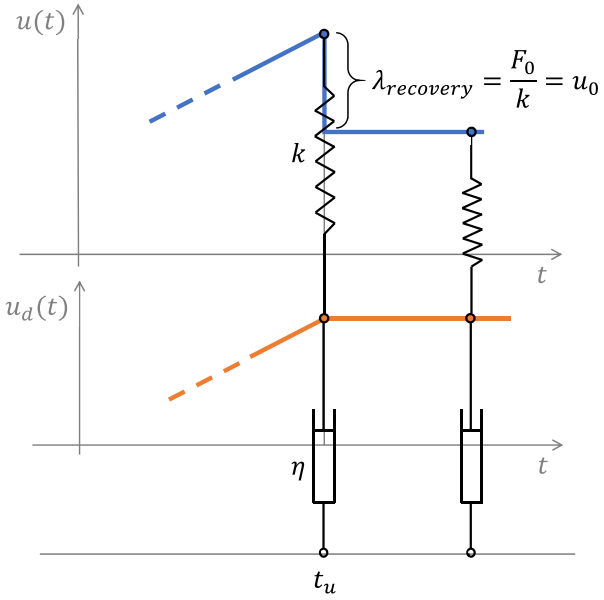
When the force is instantaneously removed by applying the step force function $F(t)$, defined in equation (3.11) and shown in Figure 3.9a, to the previously loaded Maxwell element at time $t = t_u$, this produces the following response, shown in Figure 3.9b

$t = t_u$: The spring immediately contracts by $\lambda_{recovery}$, the same amount it stretched by when the force was applied at $t = t_l$ which was denoted u_0. The spring component of the Maxwell element displays elastic behaviour as it recovers to its original dimensions. The dashpot, however, will not recover and will remain in its stretched state [18].

$t > t_1$: There is no force being applied to the element, and the spring has returned to its original dimensions. However, the dashpot remains at its displaced position due to the initial force, F_0, that was applied.



(a) Instantaneous unloading of force function of magnitude F_0.

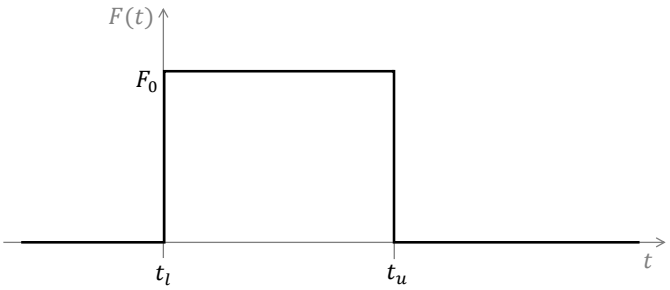


(b) Displacement-time graph response to the step force function (with the element diagram overlaid at varying stages to portray the physical behaviour of the element).

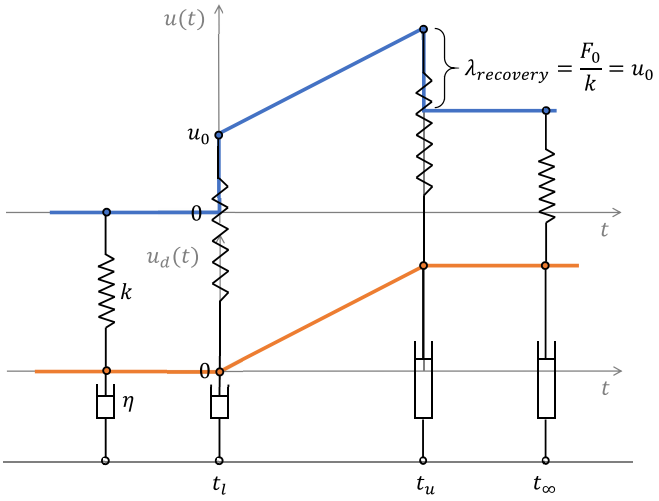
Figure 3.9: Unloading of an instantaneous step force to a Maxwell element and the displacement response over time.

3.2.3 Maxwell Element: Instantaneous Loading and Unloading of a Force Step Function

An overview illustration of the configuration of the Maxwell element at each stage of loading and unloading, and the displacement-time graph response of the Maxwell element to the step force is shown in Figure 3.10.



(a) Instantaneous step force of magnitude F_0 applied for a time period of $t_u - t_l$.



(b) Displacement-time graph response to the instantaneous step up, step down force.

Figure 3.10: A square force pulse applied to a Maxwell element and the displacement response over time.

3.2.4 Maxwell Element: Instantaneous Prescribed Displacement

To investigate the force relaxation behaviour of a Maxwell element, it is given an instantaneous prescribed displacement, u_0. This produces the following responses, shown in Figure 3.11

$t = t_l$: The prescribed displacement of the Maxwell element, u_0, is taken up by the spring (the dashpot cannot accommodate an instantaneous displacement).

The spring is at its maximum stretch at $t = t_l$ and therefore has the maximum force at this point.

$t_l < t < t_{eq}$: The force of the spring (created by the prescribed displacement) starts to act on the dashpot and the spring starts to contract. The force in the spring is always equal to the force of the dashpot. Since the spring starts to contract from its maximum displaced position (and therefore maximum force) at $t = t_l$, the force of the spring decreases until it reaches zero.

The force of the dashpot is proportional to the rate of change of displacement with the slope of the displacement-time graph at a maximum at $t = t_l$ and steadily decreases until it reaches zero at some equilibrium time, t_{eq}. This occurs when the spring has returned to its original dimensions and no longer exerts a force on the dashpot.

The change in magnitude of the stretch of the dashpot in the positive direction (extension), is equal to the change in magnitude of the stretch of the spring in the negative direction (contraction).

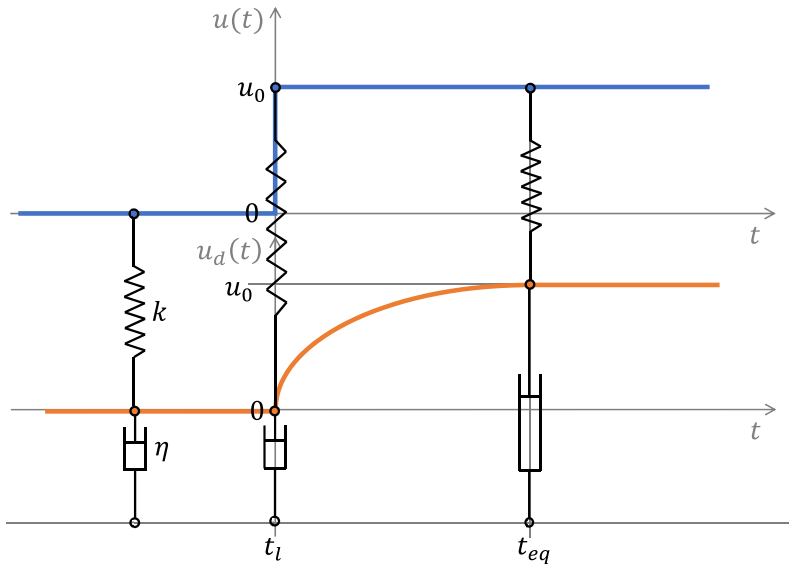
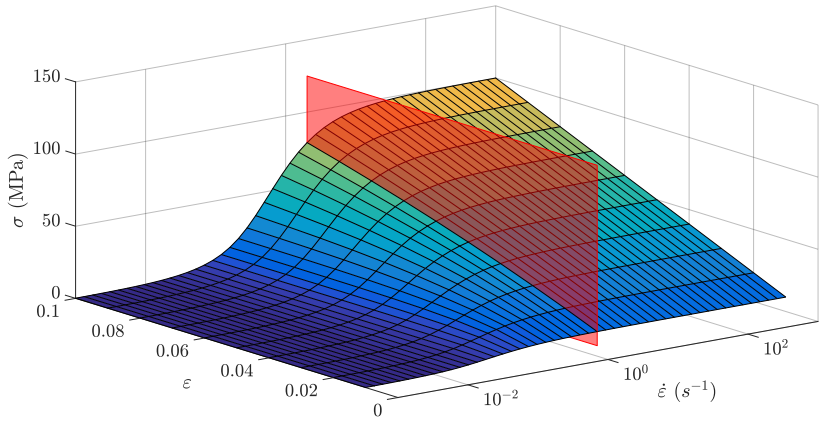


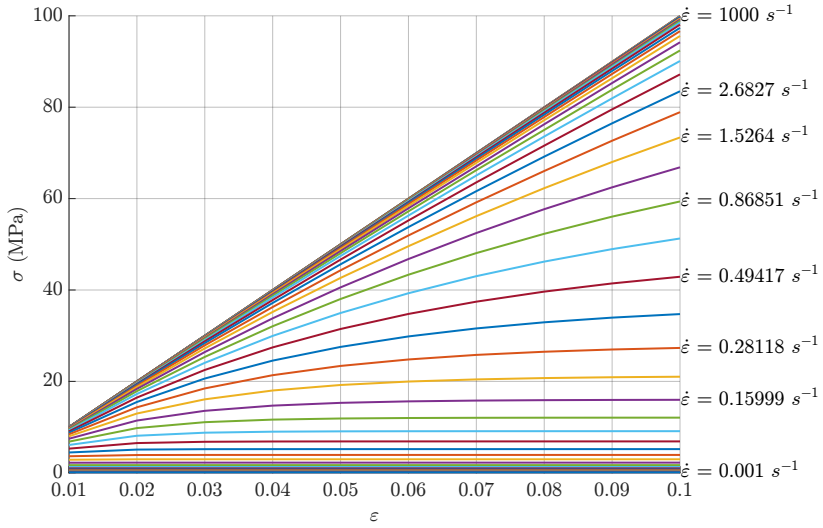
Figure 3.11: Applying a prescribed displacement and the displacement history of the mid point of the Maxwell element.

3.2.5 Maxwell Element: Constant Strain Rate

Examining the behaviour of the Maxwell element at different strain rates requires the element in Figure 3.7, to be fixed at one node, such that $u_2 = 0$, while the other node is moved at a constant velocity. The stress - strain results from each strain rate test are plotted for varying strain rates on a surface, shown in Figure 3.12a.



(a) Surface cut along stress - strain lines.

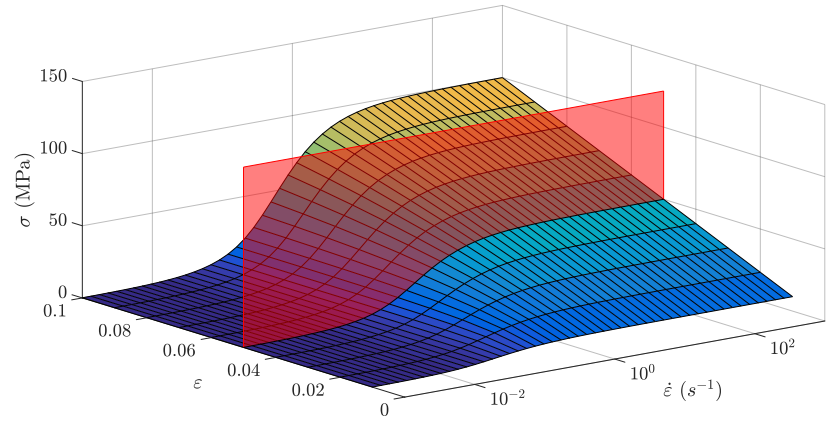


(b) Stress - strain graphs.

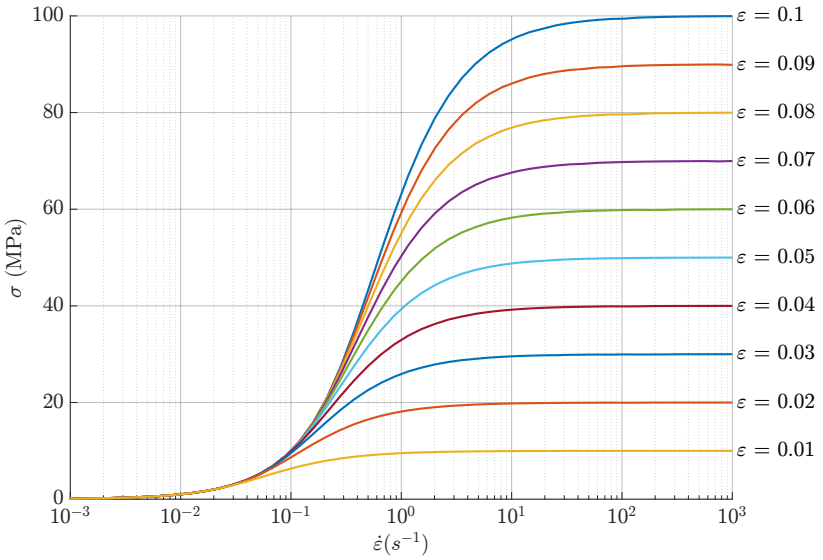
Figure 3.12: The response of a linear Maxwell element showing the stress - strain graphs at varying strain rates.

If the surface is sliced perpendicular to the stress - strain plane, the stress - strain results can be obtained at the varying strain rates, see Figure 3.12b. This gives the stress - strain graphs in Figure 3.12a. The strain rate dependent behaviour of the Maxwell element is best described by the stress - strain rate curves. These can be obtained by slicing the surface perpendicular to the stress - strain rate plane, shown in Figure 3.13a. The stress - strain curves are then shown in Figure 3.13b.

These stress - strain rate curves show that at low strain rates the Maxwell element will not provide resistance to motion, as the dashpot will not provide any resistance and therefore the spring will not be activated. At higher strain rates the dashpot provides an increasing resistance to motion and starts to lock up. The spring is therefore activated and at high strain rates the Maxwell element will exhibit a stress response, shown in Figure 3.13b, according to the stiffness of the Maxwell spring.



(a) Surface cut along stress - strain rate lines.



(b) Stress - strain rate graphs.

Figure 3.13: The response of a linear Maxwell element showing the stress - strain rate graphs at varying strains.

3.2.6 Hereditary Integral Formulation

For the purpose of implementing a Maxwell element formulation in three dimensions and because we cannot directly relate the force of the spring to the displacement across the element, an equivalent formulation of the Maxwell element was found such that tracking the internal node is not necessary. This formulation is referred to as the hereditary integral [19, 20]. The way that this hereditary integral is formulated starts with the differential equation of a Maxwell element, which is derived as follows, starting with the definition of the displacement of the Maxwell element

$$u = u_k + u_\eta, \tag{3.16}$$

taking the derivative with respect to time gives

$$\frac{\partial u}{\partial t} = \frac{\partial u_k}{\partial t} + \frac{\partial u_\eta}{\partial t}. \tag{3.17}$$

The terms $\frac{\partial u_k}{\partial t}$ and $\frac{\partial u_\eta}{\partial t}$ can be derived from the definition of the force across a spring due to the displacement and the force across a dashpot due to the rate of displacement, which are equal to the force across the element. The force acting on the spring is proportional to the stretch of that spring such that

$$F_k = F = ku_k. \tag{3.18}$$

This statement can be differentiated with respect to time to give an expression for the velocity as

$$\frac{\partial u_k}{\partial t} = \frac{1}{k}\frac{\partial F}{\partial t}. \tag{3.19}$$

The force acting on the dashpot is

$$F_\eta = F = \eta\frac{\partial u_\eta}{\partial t} \tag{3.20}$$

which means the velocity of the dashpot is

$$\frac{\partial u_\eta}{\partial t} = \frac{F}{\eta}. \tag{3.21}$$

Substituting equations (3.19) and (3.21) into (3.17) gives

$$\frac{\partial u}{\partial t} = \frac{1}{k}\frac{\partial F}{\partial t} + \frac{F}{\eta} \tag{3.22}$$

or equivalently

$$\begin{aligned} \frac{\partial F}{\partial t} + \frac{k}{\eta}F &= k\frac{\partial u}{\partial t} \\ \frac{dF}{dt} + \frac{k}{\eta}F &= k\dot{u} \end{aligned} \tag{3.23}$$

where $\dot{u} = \frac{du}{dt}$.

This takes the form of a first order differential equation [18]

$$\frac{dy}{dx} + p(x)y = r(x). \tag{3.24}$$

Finding a non-homogeneous solution to the first order differential equation gives a form of the hereditary integral that can be applied to a Maxwell element. Solving the non-homogeneous solution requires substitution, therefore let

$$h = \int p(x)dx \tag{3.25}$$

and let its derivative be

$$h' = \frac{dh}{dx} = p(x) \ . \tag{3.26}$$

We seek a solution to the differential equation (3.24) by considering the derivative of ye^h

$$\begin{aligned} \frac{d}{dx}ye^h &= y'e^h + y(e^h)' \\ &= y'e^h + ye^h h' \\ &= e^h \left[y' + yh'\right] \\ &= e^h \left[y' + p(x)y\right] . \end{aligned} \tag{3.27}$$

This is equal to the left hand side of the first order differential equation (3.24), multiplied by e^h.

Setting

$$\frac{d}{dx}ye^h = e^h r(x) \tag{3.28}$$

gives

$$ye^h = \int e^h r(x)dx + c \tag{3.29}$$

and thus

$$\begin{aligned} y &= e^{-h}\left[\int e^h r(x)dx + c\right] \\ &= e^{-\int p(x)dx}\left[\int e^{\int p(x)dx} r(x)dx + c\right] . \end{aligned} \tag{3.30}$$

Applying this to equation (3.23), which is the differential equation describing the Maxwell element

$$\frac{dF}{dt} + \frac{k}{\eta}F = k\dot{u} \tag{3.31}$$

where equation (3.31) is equivalent to equation (3.24) such that F is y, $\frac{k}{\eta}$ is $p(x)$ and $k\dot{u}$ is $r(x)$.

We can now use the solution to equation (3.30) to obtain an expression for the force as a function of time

$$F(t) = e^{-\int \frac{k}{\eta}dt}\left[\int e^{\int \frac{k}{\eta}dt} k\dot{u}(t)dt + c\right] . \tag{3.32}$$

To find the force acting on the Maxwell element at time t the limits of integration are between 0 and t, and a dummy variable τ is introduced to evaluate the inner integral with respect to τ

$$\begin{aligned} F &= e^{-\int_0^t \frac{k}{\eta} dt} \int_0^t e^{\int_0^\tau \frac{k}{\eta} d\tau} k\dot{u}(\tau) d\tau \\ &= e^{-\frac{k}{\eta}t} \int_0^t e^{\frac{k}{\eta}\tau} k\dot{u}(\tau) d\tau. \end{aligned} \tag{3.33}$$

Thus

$$F(t) = \int_0^t e^{\frac{k}{\eta}(\tau - t)} k\dot{u}(\tau) d\tau. \tag{3.34}$$

which is known as the hereditary integral.

3.3 Standard Linear Model

The final visco-elastic model presented in this chapter is the Standard Linear Model, which consists of a linear Maxwell element in parallel with a spring, shown in Figure 3.14. In order to investigate the strain rate dependent behaviour of this element, node 3 is fixed and node 1 is moved at a constant velocity. This test is carried out at varying strain rates.

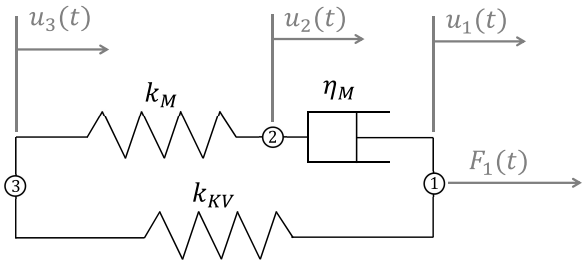
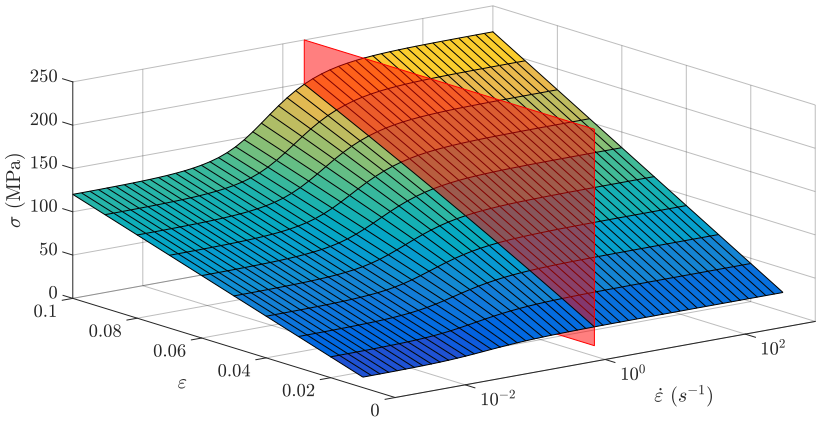
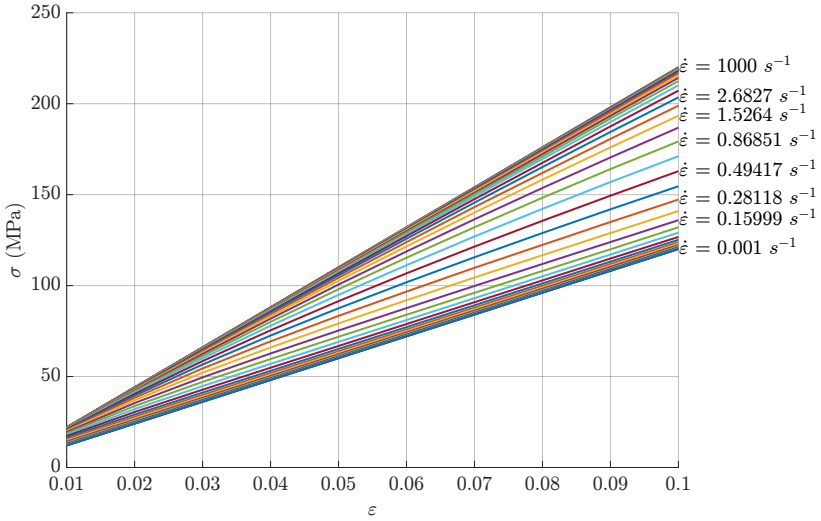


Figure 3.14: The arrangement of the Standard Linear Model element.

The visualisation of the stress of the element at varying strains and strain rates would be the surface shown in Figure 3.15a. Slicing this surface along the stress - strain plane would give the stress - strain results for the Standard Linear Model as shown in Figure 3.15b.



(a) Surface cut along stress - strain lines.

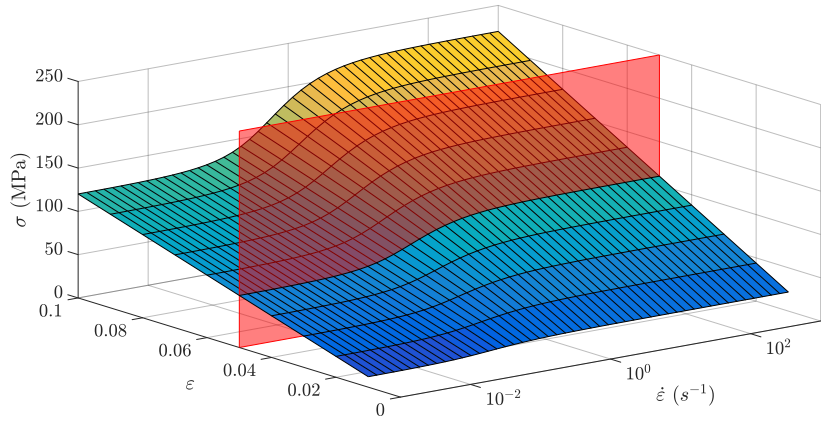


(b) Stress - strain graphs.

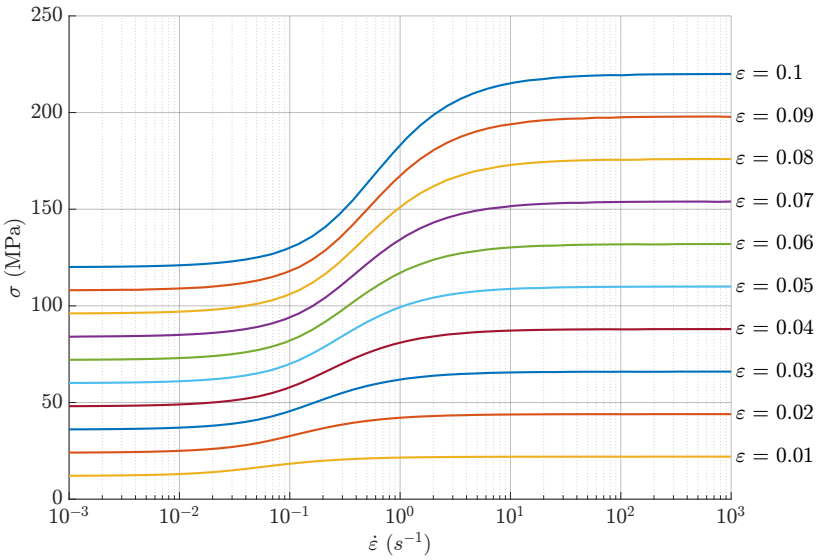
Figure 3.15: The response of a Standard Linear Model element showing the stress - strain graphs at varying strain rates.

If the stress - strain - strain rate surface was sliced along the stress - strain rate plane, as in Figure 3.16a, the stress - strain rate curves in Figure 3.16b would result. These stress - strain rate curves describe the strain rate dependent behaviour of this element.

At low strain rates, the Standard Linear model gives an elastic response driven by the value of the Kelvin-Voigt spring stiffness, k_{KV}. At high strain rates, this element gives a elastic response driven by the sum of the Kelvin Voigt and Maxwell spring stiffness, k_{KV} and k_M. At intermediate strain rates, there is a transition response between the two elastic responses. The strain rate at which this transition begins is determined by the Maxwell dashpot, η_M. This describes the characteristic shape of the behaviour that is seen from visco-elastic materials such as bone, shown in Figure 3.16b.



(a) Surface cut along stress - strain rate lines.



(b) Stress - strain rate graphs.

Figure 3.16: The response of a Standard Linear Model element showing the stress - strain rate graphs at varying strains.

Chapter 4

Extending Fundamental Visco-elastic Models to Three Dimensions

In order to implement the one-dimensional models outlined in Chapter 3 in finite element analysis software, the previous models were extended to three dimensions. First, the development of the linear elastic model was implemented in three dimensions. This implementation was then used to develop a solution to a three-dimensional dashpot, following the approach by Shim *et al.* [1], which will be discussed in Chapter 5.1. Lastly, using the hereditary integral formulation for one dimension, a linear Maxwell solution was developed in three dimensions, for use in Finite Element Analysis (FEA) software.

4.1 Spring

The stress equation for a linear spring in one dimension is denoted as the elastic stress

$$\sigma_E(t) = E\varepsilon(t) \tag{4.1}$$

where E is the elastic modulus.

This can be written in three dimensions with second order tensors for the stress $\boldsymbol{\sigma}$, and strain $\boldsymbol{\varepsilon}$, with the elastic modulus becoming a fourth order tensor $\mathfrak{C}$ [21], as

$$\boldsymbol{\sigma}_E(t) = \mathfrak{C} : \boldsymbol{\varepsilon}(t). \tag{4.2}$$

Using Voigt notation, and assuming isotropic behaviour and infinitesimal strain [22], equation (4.2)

can be written in matrix form as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_E = \frac{E}{(1-2\nu)(1+\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ 2\varepsilon_{23} \\ 2\varepsilon_{13} \\ 2\varepsilon_{12} \end{bmatrix}. \tag{4.3}$$

Multiplying out and grouping gives

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_E = \begin{bmatrix} \frac{E}{1+\nu}\varepsilon_{11} + \frac{E\nu}{(1+\nu)(1-2\nu)}(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33}) \\ \frac{E}{1+\nu}\varepsilon_{22} + \frac{E\nu}{(1+\nu)(1-2\nu)}(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33}) \\ \frac{E}{1+\nu}\varepsilon_{33} + \frac{E\nu}{(1+\nu)(1-2\nu)}(\varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33}) \\ \frac{E}{2(1+\nu)}2\varepsilon_{23} \\ \frac{E}{2(1+\nu)}2\varepsilon_{13} \\ \frac{E}{2(1+\nu)}2\varepsilon_{12} \end{bmatrix}. \tag{4.4}$$

Common factors are then removed to yield

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_E = \left(\begin{bmatrix} \frac{E}{1+\nu}\varepsilon_{11} \\ \frac{E}{1+\nu}\varepsilon_{22} \\ \frac{E}{1+\nu}\varepsilon_{33} \\ \frac{E}{2(1+\nu)}2\varepsilon_{23} \\ \frac{E}{2(1+\nu)}2\varepsilon_{13} \\ \frac{E}{2(1+\nu)}2\varepsilon_{12} \end{bmatrix} + \frac{E\nu}{(1+\nu)(1-2\nu)} \begin{bmatrix} \varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33} \\ \varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33} \\ \varepsilon_{11}+\varepsilon_{22}+\varepsilon_{33} \\ 0 \\ 0 \\ 0 \end{bmatrix} \right), \tag{4.5}$$

and rewriting in tensor form gives

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}_E = \frac{E}{1+\nu} \left(\frac{\nu}{1-2\nu} \begin{bmatrix} tr(\boldsymbol{\varepsilon}) & 0 & 0 \\ 0 & tr(\boldsymbol{\varepsilon}) & 0 \\ 0 & 0 & tr(\boldsymbol{\varepsilon}) \end{bmatrix} + \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix} \right), \tag{4.6}$$

where

$$\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_{11} & \varepsilon_{12} & \varepsilon_{13} \\ \varepsilon_{12} & \varepsilon_{22} & \varepsilon_{23} \\ \varepsilon_{13} & \varepsilon_{23} & \varepsilon_{33} \end{bmatrix}. \tag{4.7}$$

The three-dimensional stress equation can now be written in a form similar to the one-dimensional formulation (4.1)

$$\boldsymbol{\sigma}_E(t) = E\boldsymbol{\varepsilon}_{3D}(t) \tag{4.8}$$

where

$$\boldsymbol{\varepsilon}_{3D}(t) = \frac{1}{1+\nu}\left(\frac{\nu}{1-2\nu}tr(\boldsymbol{\varepsilon}(t))\boldsymbol{I} + \boldsymbol{\varepsilon}(t)\right). \tag{4.9}$$

This can be written in terms of constants, λ_1 and λ_2, as

$$\boldsymbol{\varepsilon}_{3D}(t) = \lambda_1 tr(\boldsymbol{\varepsilon}(t))\boldsymbol{I} + \lambda_2\boldsymbol{\varepsilon}(t) \tag{4.10}$$

where

$$\lambda_1 = \frac{\nu}{(1+\nu)(1-2\nu)} \tag{4.11}$$

and

$$\lambda_2 = \frac{1}{1+\nu}\,. \tag{4.12}$$

4.2 Dashpot

A similar approach can be taken with the dashpot, again under the assumption of isotropy and infinitesimal strain. The stress equation for a linear dashpot in one dimension is denoted as the viscous stress

$$\sigma_V(t) = \eta\dot{\varepsilon}(t). \tag{4.13}$$

Similar to the approach taken to extend the linear elastic stress equation (4.1), to three dimensions, equation (4.8), the viscous stress can be extended to three dimensions by replacing the strain rate term with a three-dimensional strain rate tensor

$$\boldsymbol{\sigma}_V(t) = \eta\dot{\boldsymbol{\varepsilon}}_{3D}(t) \tag{4.14}$$

where $\dot{\boldsymbol{\varepsilon}}_{3D}$ is defined by Shim *et al.* [1] as

$$\dot{\boldsymbol{\varepsilon}}(t)_{3D} = \lambda_1 tr(\dot{\boldsymbol{\varepsilon}}(t))\boldsymbol{I} + \lambda_2\dot{\boldsymbol{\varepsilon}}(t) \tag{4.15}$$

and

$$\dot{\boldsymbol{\varepsilon}}(t) = \begin{bmatrix} \dot{\varepsilon}_{11} & \dot{\varepsilon}_{12} & \dot{\varepsilon}_{13} \\ \dot{\varepsilon}_{12} & \dot{\varepsilon}_{22} & \dot{\varepsilon}_{23} \\ \dot{\varepsilon}_{13} & \dot{\varepsilon}_{23} & \dot{\varepsilon}_{33} \end{bmatrix}(t). \tag{4.16}$$

Substituting into equation 4.14 gives

$$\boldsymbol{\sigma}_V(t) = \eta\left(\lambda_1 tr(\dot{\boldsymbol{\varepsilon}}(t))\boldsymbol{I} + \lambda_2\dot{\boldsymbol{\varepsilon}}(t)\right) \tag{4.17}$$

and can be written as

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}_V = \eta \left(\lambda_1 \begin{bmatrix} tr(\dot{\boldsymbol{\varepsilon}}) & 0 & 0 \\ 0 & tr(\dot{\boldsymbol{\varepsilon}}) & 0 \\ 0 & 0 & tr(\dot{\boldsymbol{\varepsilon}}) \end{bmatrix} + \lambda_2 \begin{bmatrix} \dot{\varepsilon}_{11} & \dot{\varepsilon}_{12} & \dot{\varepsilon}_{13} \\ \dot{\varepsilon}_{12} & \dot{\varepsilon}_{22} & \dot{\varepsilon}_{23} \\ \dot{\varepsilon}_{13} & \dot{\varepsilon}_{23} & \dot{\varepsilon}_{33} \end{bmatrix} \right). \tag{4.18}$$

We seek an analogous expression of the stress across the dashpot in three dimensions, which we assume will have the form of equation (4.14). This can be written in Voigt notation as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_V = \begin{bmatrix} \eta \left(\lambda_1 (\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2 \dot{\varepsilon}_{11} \right) \\ \eta \left(\lambda_1 (\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2 \dot{\varepsilon}_{22} \right) \\ \eta \left(\lambda_1 (\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2 \dot{\varepsilon}_{33} \right) \\ \eta \frac{\lambda_2}{2} 2\dot{\varepsilon}_{23} \\ \eta \frac{\lambda_2}{2} 2\dot{\varepsilon}_{13} \\ \eta \frac{\lambda_2}{2} 2\dot{\varepsilon}_{12} \end{bmatrix} \tag{4.19}$$

or as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_V = \frac{\eta}{(1-2\nu)(1+\nu)} \begin{bmatrix} (1-\nu) & \nu & \nu & 0 & 0 & 0 \\ \nu & (1-\nu) & \nu & 0 & 0 & 0 \\ \nu & \nu & (1-\nu) & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \begin{bmatrix} \dot{\varepsilon}_{11} \\ \dot{\varepsilon}_{22} \\ \dot{\varepsilon}_{33} \\ 2\dot{\varepsilon}_{23} \\ 2\dot{\varepsilon}_{13} \\ 2\dot{\varepsilon}_{12} \end{bmatrix}. \tag{4.20}$$

4.3 Linear Maxwell Element

For the linear Maxwell element, the stress equation is formulated from the hereditary integral in a time-stepping fashion to produce what is denoted as the Maxwell stress equation in one dimension

$$\sigma_M(t) = \int_0^t e^{\frac{E}{\eta}(\tau - t)} E \dot{\varepsilon}(\tau) d\tau. \tag{4.21}$$

Shim *et al.* [1] extend (4.21) to three dimensions, by replacing the strain rate term with the three dimensional strain rate tensor (equation 4.15), which gives

$$\boldsymbol{\sigma}_M(t) = \int_0^t e^{\frac{E}{\eta}(\tau - t)} E \dot{\boldsymbol{\varepsilon}}_{3D}(\tau) d\tau \tag{4.22}$$

where $\dot{\boldsymbol{\varepsilon}}_{3D}(\tau)$ is defined in equation (4.15) and therefore the stress is

$$\boldsymbol{\sigma}_M(t) = \int_0^t e^{\frac{E}{\eta}(\tau-t)} E\left(\lambda_1 tr(\dot{\boldsymbol{\varepsilon}}(\tau))\boldsymbol{I} + \lambda_2 \dot{\boldsymbol{\varepsilon}}(\tau)\right) d\tau. \tag{4.23}$$

Writing this equation in tensor form gives

$$\begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{12} & \sigma_{22} & \sigma_{23} \\ \sigma_{13} & \sigma_{23} & \sigma_{33} \end{bmatrix}_M (t) = \int_0^t e^{\frac{E}{\eta}(\tau-t)} E \left(\lambda_1 \begin{bmatrix} tr(\dot{\boldsymbol{\varepsilon}}) & 0 & 0 \\ 0 & tr(\dot{\boldsymbol{\varepsilon}}) & 0 \\ 0 & 0 & tr(\dot{\boldsymbol{\varepsilon}}) \end{bmatrix} (\tau) + \lambda_2 \begin{bmatrix} \dot{\varepsilon}_{11} & \dot{\varepsilon}_{12} & \dot{\varepsilon}_{13} \\ \dot{\varepsilon}_{12} & \dot{\varepsilon}_{22} & \dot{\varepsilon}_{23} \\ \dot{\varepsilon}_{13} & \dot{\varepsilon}_{23} & \dot{\varepsilon}_{33} \end{bmatrix} (\tau) \right) d\tau \tag{4.24}$$

which can be written in Voigt notation as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}_M (t) = \int_0^t e^{\frac{E}{\eta}(\tau-t)} E \begin{bmatrix} \eta\left(\lambda_1(\dot{\varepsilon}_{11}+\dot{\varepsilon}_{22}+\dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{11}\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11}+\dot{\varepsilon}_{22}+\dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{22}\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11}+\dot{\varepsilon}_{22}+\dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{33}\right) \\ \eta\frac{\lambda_2}{2}2\dot{\varepsilon}_{23} \\ \eta\frac{\lambda_2}{2}2\dot{\varepsilon}_{13} \\ \eta\frac{\lambda_2}{2}2\dot{\varepsilon}_{12} \end{bmatrix} (\tau)\, d\tau. \tag{4.25}$$

Chapter 5

Visco-elastic Theory as Applied to Cortical Bone Modelling

Chapters 3 and 4 outlined the visco-elastic theory necessary to understand the models that will be investigated and expanded upon in this dissertation. The visco-elastic models, introduced in Chapter 2, by Shim *et al.* [1], Bekker *et al.* [2] and Cloete *et al.* [3] are explored in more detail in this chapter.

5.1 Shim Model

As mentioned in Chapter 2.3, Shim *et al.* [1] proposed a model for cancellous bone. The model consists of a density-dependent linear elastic element (labelled σ^e) and a rate-dependent non-linear visco-elastic element (labelled σ^{v}), shown in Figure 5.1.

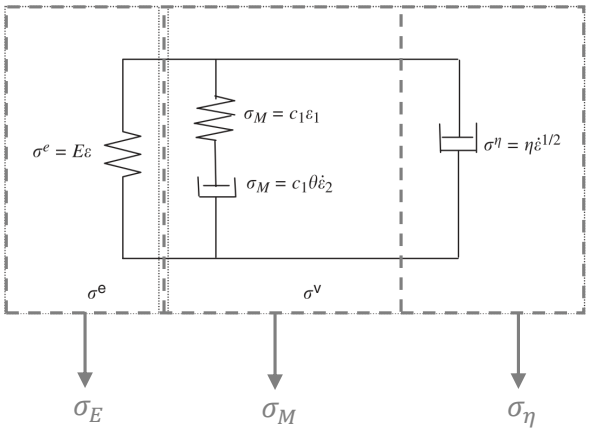


Figure 5.1: The arrangement of the augmented form of the model proposed by Shim *et al.* [1].

The non-linear visco-elastic portion of the model consists of a linear Maxwell element and non-linear Newtonian dashpot in parallel. The stress in the rate dependent portion is defined as

$$\sigma^{\mathrm{v}} = \sigma_M + \sigma_\eta \tag{5.1}$$

where σ_M is the stress contribution from the Maxwell element and σ_η is the stress contribution from the dashpot.

The stress contribution from the dashpot is defined as

$$\sigma_\eta = \eta \dot{\varepsilon}^{\frac{1}{2}} \tag{5.2}$$

where η is the dashpot coefficient.

The stress contribution from the Maxwell element is defined as

$$\sigma_M = \int_0^t c_1 \dot{\varepsilon}(\tau) e^{\left(-\frac{t-\tau}{\theta}\right)} d\tau \tag{5.3}$$

where c_1 is the spring stiffness, t is the current time, τ is the dummy time variable, and θ is the ratio between the spring stiffness and the dashpot coefficient.

The stress in the density-dependent portion is defined to be

$$\sigma^e = \sigma_E = E\varepsilon \tag{5.4}$$

where E is the stiffness of the spring.

Shim *et al.* proposed the elastic modulus as a function of the density, ρ, a constant modulus, E_0, and a power, β, such that

$$E = E_0 \rho^\beta \tag{5.5}$$

where β is a scaling power used to match the model to experimental data.

The stress in the complete model is found additively as

$$\sigma = \sigma_E + \sigma_M + \sigma_\eta = E_0 \rho^\beta \varepsilon + \int_0^t c_1 \dot{\varepsilon}(\tau) e^{\left(-\frac{t-\tau}{\theta}\right)} d\tau + \eta \dot{\varepsilon}^{\frac{1}{2}}. \tag{5.6}$$

Shim *et al.* propose an isotropic extension of the model to three dimensions as a first step in the development of a representative constitutive model. Despite the highly anisotropic nature of bone, most experimental results report on the uniaxial behaviour of bone and most models have been in one dimension. A thorough investigation into the behaviour of an isotropic model is required before such a model could be extended to an anisotropic model in three dimensions.

Under the assumption of isotropic, infinitesimal strain, the elastic stress can be defined as

$$\boldsymbol{\sigma}_E = \frac{\mathfrak{C}}{2}\left(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T\right) = \frac{E}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\boldsymbol{\varepsilon})\boldsymbol{I} + \boldsymbol{\varepsilon}\right) \tag{5.7}$$

where $\mathfrak{C}$ is the fourth order stiffness tensor, $\boldsymbol{u}$ is the displacement vector, E is the elastic modulus, $\boldsymbol{\varepsilon}$ is the infinitesimal strain tensor, ν is Poison's ratio, and $\boldsymbol{I}$ is the identity tensor.

Using the definition of the elastic modulus in equation (5.5), the three-dimensional elastic stress tensor, $\boldsymbol{\sigma}_E$, was then defined as

$$\boldsymbol{\sigma}_E = \frac{E_0 \rho^\beta}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\boldsymbol{\varepsilon})\boldsymbol{I} + \boldsymbol{\varepsilon}\right). \tag{5.8}$$

From equations (5.4) and (5.7), the three-dimensional strain tensor can be defined as

$$\boldsymbol{\varepsilon}_{3D} = \frac{1}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\boldsymbol{\varepsilon})\boldsymbol{I} + \boldsymbol{\varepsilon}\right). \tag{5.9}$$

The stress - strain relationship was extended by defining a three-dimensional equivalent to the strain rate. This was proposed to be the time derivative of the three-dimensional strain tensor defined in equation (5.9)

$$\dot{\boldsymbol{\varepsilon}}_{3D} = \frac{D\boldsymbol{\varepsilon}_{3D}}{Dt} = \frac{1}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\dot{\boldsymbol{\varepsilon}})\boldsymbol{I} + \dot{\boldsymbol{\varepsilon}}\right) \tag{5.10}$$

where $\dot{\boldsymbol{\varepsilon}}$ is the strain rate tensor.

This allows the total stress tensor, $\boldsymbol{\sigma}$, of the model to be defined as

$$\boldsymbol{\sigma} = \frac{E_0 \rho^\beta}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\boldsymbol{\varepsilon})\boldsymbol{I} + \boldsymbol{\varepsilon}\right) + \int_0^t \frac{c_1}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\dot{\boldsymbol{\varepsilon}})\boldsymbol{I} + \dot{\boldsymbol{\varepsilon}}\right)\dot{\varepsilon}(\tau) e^{\left(-\frac{t-\tau}{\theta}\right)} d\tau + \frac{\eta}{1+\nu}\left(\frac{\nu}{1-2\nu} tr(\dot{\boldsymbol{\varepsilon}})\boldsymbol{I} + \dot{\boldsymbol{\varepsilon}}\right)^{\frac{1}{2}}. \tag{5.11}$$

5.2 Bekker Model

Bekker *et al.* [2] developed a frame-invariant model, shown in Figure 5.2, that focuses on the material response of cortical bovine bone up to the point of yield or failure, building on the work by Shim *et al.* [1].

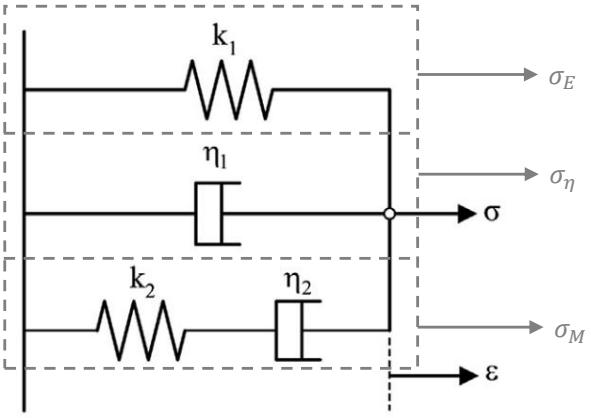


Figure 5.2: The arrangement of the modified Shim model proposed by Bekker *et al.* (augmented from [2]).

Where Shim *et al.* had defined the dashpot stress, σ_η, to be defined in terms of the strain rate raised to a power of $\frac{1}{2}$, Bekker *et al.* made the power a parameter such that the dashpot stress was generalised to

$$\sigma_\eta = \eta \dot{\varepsilon}^P \tag{5.12}$$

where η is the dashpot coefficient and P is the power.

To generalise the three-dimensional dashpot stress tensor, $\boldsymbol{\sigma}_\eta$, that was proposed by Shim *et al.* , the stress - strain rate relationship was regrouped as

$$\boldsymbol{\sigma}_\eta = \eta \dot{\boldsymbol{\varepsilon}}_{3D} = \eta \frac{\dot{\varepsilon}_{eq}}{\dot{\varepsilon}_{eq}} \dot{\boldsymbol{\varepsilon}}_{3D} = \eta \dot{\varepsilon}_{eq} \left(\frac{\dot{\boldsymbol{\varepsilon}}_{3D}}{\dot{\varepsilon}_{eq}} \right) \tag{5.13}$$

Where the scalar equivalent strain rate, $\dot{\varepsilon}_{eq}$, was defined as

$$\dot{\varepsilon}_{eq} = \sqrt{\frac{\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}}{1 + 2\nu^2}} \, . \tag{5.14}$$

In order to capture effects when the strain rate is high, the stress - strain rate relationship was made to be non-linear by raising the magnitude of the strain rate, $\dot{\varepsilon}_{eq}$, to a power P, resulting in

$$\boldsymbol{\sigma}_\eta = \eta \left(\dot{\varepsilon}_{eq} \right)^P \left(\frac{\dot{\boldsymbol{\varepsilon}}_{3D}}{\dot{\varepsilon}_{eq}} \right) = \eta \left(\dot{\varepsilon}_{eq} \right)^{P-1} \dot{\boldsymbol{\varepsilon}}_{3D}. \tag{5.15}$$

With these modifications, the three-dimensional model proposed by Bekker *et al.* is

$$\boldsymbol{\sigma}(t) = k_1 \boldsymbol{\varepsilon}_{3D}(t) + \eta_1 \left(\dot{\varepsilon}_{eq} \right)^{P-1} \dot{\boldsymbol{\varepsilon}}_{3D}(t) + \int_0^t k_2 \dot{\boldsymbol{\varepsilon}}_{3D}(\tau) e^{\left(-\frac{t-\tau}{\theta_2} \right)} d\tau. \tag{5.16}$$

5.3 Cloete Model

Cloete *et al.* [3] built up a visco-elastic model in a similar way to the Shim and Bekker models with non-linear Maxwell elements. The proposed model consists of a non-linear Kelvin-Voigt element and two non-linear Maxwell elements in parallel, shown in Figure 5.3.

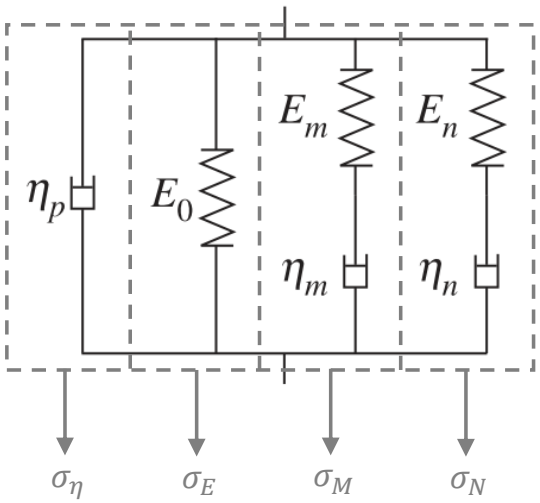


Figure 5.3: The arrangement of the model proposed by Cloete *et al.* (augmented from [3]).

The elastic portion, σ_E, is designed to capture the modulus at low strain rates and the dashpot contribution, σ_η, which is non-linear, is designed to capture the response at high strain rates. The two non-linear Maxwell elements (any spring and dashpot in series), σ_M and σ_N, have strain rate terms which are raised to different powers. One has a power $m < 1$, which captures the response at low strain rates and the other has a power $n > 1$, which is designed to match the gradient of the transition response in the ISR regime.

The stress contribution from the non-linear Maxwell element in this model is computed using a non-linear adaptation to the hereditary integral for a linear Maxwell element and is defined as

$$\sigma(t) = E_0\varepsilon(t) + \eta_p\dot{\varepsilon}(t)^p + \int_0^t E_m\dot{\varepsilon}(\tau)^m e^{\left(-\frac{E_m}{\eta_m}\right)(t-\tau)}d\tau + \int_0^t E_n\dot{\varepsilon}(\tau)^n e^{\left(-\frac{E_n}{\eta_n}\right)(t-\tau)}d\tau \tag{5.17}$$

where E_0, E_m and E_n are the stiffness of the associated springs, η_p, η_m and η_n are the coefficients of the dashpots and p, m and n are the respective powers.

With the assumption of constant strain rates, this was evaluated and simplified to

$$\sigma = E_0\varepsilon + \eta_p\dot{\varepsilon}^p + \eta_m\dot{\varepsilon}^m\left[1 - e^{\left(-\frac{E_m\varepsilon}{\eta_m\dot{\varepsilon}^m}\right)}\right] + \eta_n\dot{\varepsilon}^n\left[1 - e^{\left(-\frac{E_n\varepsilon}{\eta_n\dot{\varepsilon}^n}\right)}\right] \tag{5.18}$$

in one dimension.

Chapter 6

One-dimensional Visco-elastic Models and Non-linearity

To simulate the visco-elastic models presented in Chapter 3, we need to develop a system that is both spatially and temporally discrete. For a Maxwell element which is defined as a spring and dashpot in series, it is not sufficient to look at the element as a whole. We need to be able to track the displacement at the interface between the spring and dashpot in a Maxwell element, which is done by introducing a small intermediate mass between them. This is referred to as the intermediate mass implementation and was formulated for the linear and non-linear Maxwell element. Non-linearity with respect to the Maxwell dashpot also needs to be investigated. Two possible implementations are outlined, namely raising the velocity to a power and a dashpot coefficient function as a function of the velocity.

6.1 Discrete Implementation

Using the traditional central difference method, a temporally discrete solution was developed. A model comprised of a mass matrix, $\boldsymbol{M}$, stiffness matrix, $\boldsymbol{K}$, damping matrix, $\boldsymbol{C}$, and applied force vector, $\boldsymbol{f}$, can be modelled as either a single or multi-element solution with an equation of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \ , \tag{6.1}$$

with initial conditions defined as

$$\boldsymbol{u} = \boldsymbol{u}_0 \quad \text{and} \quad \dot{\boldsymbol{u}} = \dot{\boldsymbol{u}}_0 \tag{6.2}$$

where $\boldsymbol{u}$ is a vector of nodal displacements and $\dot{\boldsymbol{u}}$ is a vector of nodal velocities.

Using the traditional central difference approach [23] to discretise the system temporally where

$$\dot{\boldsymbol{u}}_t = \frac{1}{2\Delta t}\left(\boldsymbol{u}_{t+1} - \boldsymbol{u}_{t-1}\right) \qquad \text{and} \qquad \ddot{\boldsymbol{u}}_t = \frac{1}{\Delta t^2}\left(\boldsymbol{u}_{t+1} - 2\boldsymbol{u}_t + \boldsymbol{u}_{t-1}\right), \tag{6.3}$$

where $\boldsymbol{u}_t$ is a vector of the nodal displacements at time t.

Using the corresponding initial conditions in equation (6.2) and solving the equation of motion in equation (6.1) gives the initial acceleration as

$$\ddot{\boldsymbol{u}}_0 = \boldsymbol{M}^{-1}\left(\boldsymbol{f}_0 - \boldsymbol{C}\dot{\boldsymbol{u}}_0 - \boldsymbol{K}\boldsymbol{u}_0\right). \tag{6.4}$$

In order to start the central difference scheme at time step 0, the displacement at time step -1 is needed and this is found using a second order Taylor approximation. From equations (6.3) the displacement at time step -1 is

$$\begin{aligned}
\boldsymbol{u}_{-1} &= \boldsymbol{u}_0 - \Delta t\dot{\boldsymbol{u}}_0 + \frac{\Delta t^2}{2}\ddot{\boldsymbol{u}}_0 \\
&= \boldsymbol{u}_0 - \Delta t\dot{\boldsymbol{u}}_0 + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\left(\boldsymbol{f}_0 - \boldsymbol{C}\dot{\boldsymbol{u}}_0 - \boldsymbol{K}\boldsymbol{u}_0\right) \\
&= \left(1 - \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{K}\right)\boldsymbol{u}_0 - \left(\Delta t + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{C}\right)\dot{\boldsymbol{u}}_0 + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{f}_0.
\end{aligned} \tag{6.5}$$

This is what is known as a consistent starting condition.

Substituting equations (6.3) into (6.1) and rearranging, gives an equation for the update of the displacement in each time step based on previous values as

$$\boldsymbol{u}_{t+1} = \left[\frac{1}{\Delta t^2}\boldsymbol{M} + \frac{1}{2\Delta t}\boldsymbol{C}\right]^{-1}\left(\boldsymbol{f}_t + \left[\frac{2}{\Delta t^2}\boldsymbol{M} - \boldsymbol{K}\right]\boldsymbol{u}_t + \left[\frac{1}{2\Delta t}\boldsymbol{C} - \frac{1}{\Delta t^2}\boldsymbol{M}\right]\boldsymbol{u}_{t-1}\right). \tag{6.6}$$

Solving the system in this manner can be computationally expensive due to the need to invert the first part of equation (6.6) which is likely to not be diagonal and can be very large when working with a multi-element problem involving a large number of elements. The inversion of $\boldsymbol{M}$ is trivial when $\boldsymbol{M}$ is diagonalised therefore the issue lies with the damping matrix. To avoid needing to invert this damping matrix, the approximation used for $\dot{\boldsymbol{u}}_t$ is adjusted from the explicit one used before to [23]

$$\dot{\boldsymbol{u}}_t \approx \dot{\boldsymbol{u}}_{t-\frac{1}{2}} = \frac{1}{\Delta t}\left(\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\right). \tag{6.7}$$

This produces a solution free of an inverted damping matrix such that the update of the displacement is

$$\boldsymbol{u}_{t+1} = \left[\frac{1}{\Delta t^2}\boldsymbol{M}\right]^{-1}\left(\boldsymbol{f}_t + \left[\frac{2}{\Delta t^2}\boldsymbol{M} - \boldsymbol{K} - \frac{1}{\Delta t}\boldsymbol{C}\right]\boldsymbol{u}_t + \left[\frac{1}{\Delta t}\boldsymbol{C} - \frac{1}{\Delta t^2}\boldsymbol{M}\right]\boldsymbol{u}_{t-1}\right). \tag{6.8}$$

In all of the discrete simulations performed, this modified central difference method was used to obtain the nodal displacements over time.

6.2 Maxwell Element (with Linear Dashpot)

For a Maxwell element special care needs to be taken when formulating a solution. This is because when there is a dashpot in series with a spring within an element, the forces acting on the nodes are dependent on the interface between the spring and the dashpot. Two different discrete solutions were investigated, the first of which was an intermediate mass solution, which tracks the interface between the spring and dashpot, while the second utilised a time-stepping hereditary integral.

6.2.1 Intermediate Mass Implementation

In order to obtain a diagonal mass matrix the mass of each element was lumped equally at either node, $2n-1$ and $2n+1$, where the mass of a single element is m and the mass at each of the end nodes is $\frac{m}{2}$, see Figure 6.1. The mass of the node at the interface between the spring and the dashpot is referred to as the intermediate mass and is denoted by m_{int}.

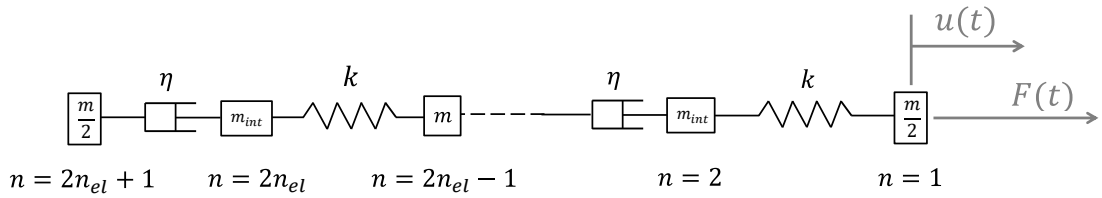


Figure 6.1: The arrangement of a discrete multi Maxwell element system in a dashpot-spring configuration with lumped masses and intermediate masses, where n is the node number.

The equation of motion described for a general system in equation (6.1) can be defined for this multi-element Maxwell system as

$$\begin{bmatrix} \frac{m}{2} & 0 & & 0 & 0 & 0 \\ 0 & m_{int} & & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & m & 0 & 0 \\ 0 & 0 & & 0 & m_{int} & 0 \\ 0 & 0 & & 0 & 0 & \frac{m}{2} \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \vdots \\ \ddot{u}_{2n_{el}-1} \\ \ddot{u}_{2n_{el}} \\ \ddot{u}_{2n_{el}+1} \end{bmatrix} + \begin{bmatrix} 0 & 0 & & 0 & 0 & 0 \\ 0 & \eta & & -\eta & 0 & 0 \\ & & \ddots & & & \\ 0 & -\eta & & \eta & 0 & 0 \\ 0 & 0 & & 0 & \eta & -\eta \\ 0 & 0 & & 0 & -\eta & \eta \end{bmatrix} \begin{bmatrix} \dot{u}_1 \\ \dot{u}_2 \\ \vdots \\ \dot{u}_{2n_{el}-1} \\ \dot{u}_{2n_{el}} \\ \dot{u}_{2n_{el}+1} \end{bmatrix}$$

$$+ \begin{bmatrix} k & -k & & 0 & 0 & 0 \\ -k & k & & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & k & -k & 0 \\ 0 & 0 & & -k & k & 0 \\ 0 & 0 & & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{2n_{el}-1} \\ u_{2n_{el}} \\ u_{2n_{el}+1} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_{2n_{el}-1} \\ f_{2n_{el}} \\ f_{2n_{el}+1} \end{bmatrix} \tag{6.9}$$

The intermediate mass, m_{int}, of the internal node was made to be small enough such that the force response due to the intermediate mass was negligible compared to that of the spring and dashpot on either side. This was determined by relating the forces on the spring and the dashpot to the acceleration of the intermediate mass. The force due to the acceleration of this mass should be small compared to the force response of the spring and dashpot.

The multi-element system above was reduced to a single element in a Dashpot-Spring configuration, see Figure 6.2. Constraining node 3 of this single Maxwell element and applying an instantaneous displacement u_0 to node 1 should produce relaxation behaviour similar to that shown in Chapter 3, Figure 3.11. Initially, the dashpot resists the sudden loading, node 2 does not move, while the spring reacts to the applied displacement. If the prescribed displacement is applied to node 1 in a positive direction (to the right), with node 3 fixed, the spring stretches by the initial amount, u_0, and node 2 relaxes (moves) in a positive direction until the spring returns to its original dimensions, Figure 6.3a.

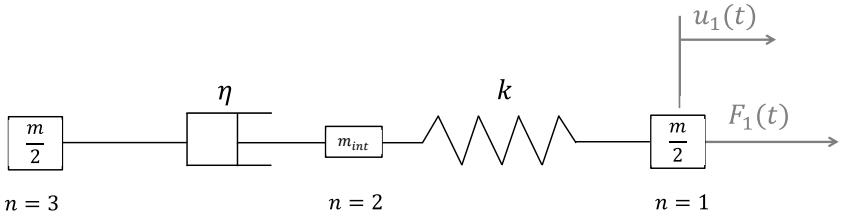


Figure 6.2: The arrangement of the discrete Maxwell element in a dashpot-spring configuration with lumped masses and intermediate mass, where n is the node number.

In order to investigate the effect of initial conditions on this element, we start with the definition of the displacement at time step -1, from equation (6.5),

$$\boldsymbol{u}_{-1} = \left(1 - \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{K}\right)\boldsymbol{u}_0 - \left(\Delta t + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{C}\right)\dot{\boldsymbol{u}}_0 + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{f}_0 \tag{6.10}$$

For an initial displacement u_0 applied to node 1 of this element, with node 3 constrained, the initial conditions $\boldsymbol{u}_0$, $\dot{\boldsymbol{u}}_0$ and $\boldsymbol{f}_0$ are defined as follows

$$\boldsymbol{u}_0 = \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix}, \qquad \dot{\boldsymbol{u}}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{f}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{6.11}$$

Using these initial conditions in equation (6.10) gives

$$\boldsymbol{u}_{-1} = \left(1 - \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{K}\right)\begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} - \left(\Delta t + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\boldsymbol{C}\right)\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + \frac{\Delta t^2}{2}\boldsymbol{M}^{-1}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \tag{6.12}$$

where only the first term of this equation will affect the displacement at time -1.

Defining the diagonal mass matrix $\boldsymbol{M}$ for this system as

$$\boldsymbol{M} = \begin{bmatrix} \frac{m}{2} & 0 & 0 \\ 0 & m_{int} & 0 \\ 0 & 0 & \frac{m}{2} \end{bmatrix}, \tag{6.13}$$

the inverse of the mass matrix $\boldsymbol{M}^{-1}$ as

$$\boldsymbol{M}^{-1} = \begin{bmatrix} \frac{2}{m} & 0 & 0 \\ 0 & \frac{1}{m_{int}} & 0 \\ 0 & 0 & \frac{2}{m} \end{bmatrix}, \tag{6.14}$$

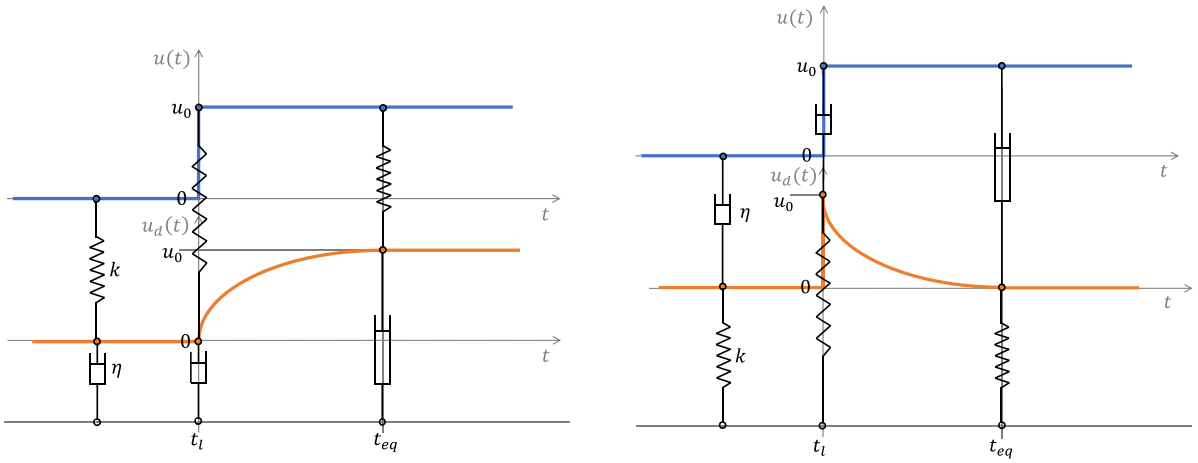
and the stiffness matrix $\boldsymbol{K}$ as

$$\boldsymbol{K} = \begin{bmatrix} k & -k & 0 \\ k & -k & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{6.15}$$

Substituting these into equation (6.12), and removing the zero terms, gives

$$\begin{aligned} \boldsymbol{u}_{-1} &= \left(1 - \frac{\Delta t^2}{2} \begin{bmatrix} \frac{2}{m} & 0 & 0 \\ 0 & \frac{1}{m_{int}} & 0 \\ 0 & 0 & \frac{2}{m} \end{bmatrix} \begin{bmatrix} k & -k & 0 \\ -k & k & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} \\ &= \left(1 - \frac{\Delta t^2}{2} \begin{bmatrix} \frac{2k}{m} & \frac{-2k}{m} & 0 \\ \frac{-k}{m_{int}} & \frac{k}{m_{int}} & 0 \\ 0 & 0 & 0 \end{bmatrix}\right) \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} - \frac{\Delta t^2}{2} \begin{bmatrix} \frac{2k}{m} u_0 \\ \frac{-k}{m_{int}} u_0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \left(1 - \frac{2k\Delta t^2}{2m}\right) u_0 \\ \frac{k\Delta t^2}{2m_{int}} u_0 \\ 0 \end{bmatrix} \end{aligned} \tag{6.16}$$

From this we can see that an approximation of the displacement of node 1 and node 2 at time step -1 has been made.



(a) Dashpot-Spring configuration.

(b) Spring-Dashpot configuration.

Figure 6.3: Relaxation behaviour of Maxwell element in the two different orientations.

Considering a reoriented Maxwell element, in a Spring-Dashpot orientation, the prescribed displacement is applied to node 1 of this Maxwell element shown in Figure 6.4 in a positive direction (to the right), with node 3 fixed. As discussed in Chapter 3.2.4, the displacement applied should result in the dashpot not moving instantaneously and the spring should stretch by the initial amount, u_0. Node 2 then relaxes (moves) in a negative direction until the spring returns to its original dimensions, see Figure 6.3b. However, with the intermediate mass solution, applying a displacement to node 1 in this arrangement does not produce the expected response. The information that node 1 has moved is not being communicated to the system via the starting conditions.

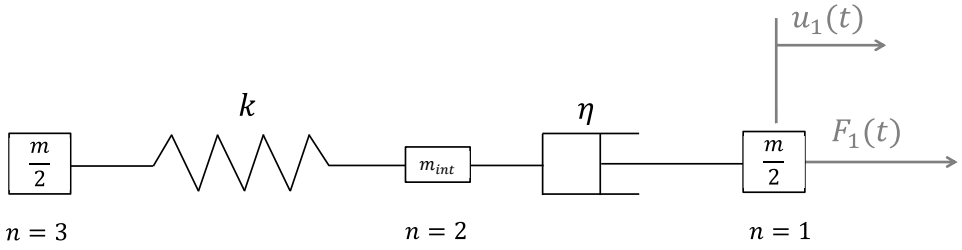


Figure 6.4: The arrangement of the discrete Maxwell element in a spring-dashpot configuration, with lumped masses and intermediate mass.

In order to confirm that the issue is due to the initial conditions, we can use equation (6.12) to find the displacement of this system at time step -1 when an initial displacement of u_0 is applied to node 1. For this system, however, the stiffness matrix is

$$\boldsymbol{K} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & k & -k \\ 0 & -k & k \end{bmatrix}. \tag{6.17}$$

Substituting this and the same mass matrix into equation (6.12) gives

$$\begin{aligned}
\boldsymbol{u}_{-1} &= \left(1 - \frac{\Delta t^2}{2}\begin{bmatrix} \frac{2}{m} & 0 & 0 \\ 0 & \frac{1}{m_{int}} & 0 \\ 0 & 0 & \frac{2}{m} \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 \\ 0 & k & -k \\ 0 & -k & k \end{bmatrix}\right)\begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} \\
&= \left(1 - \frac{\Delta t^2}{2}\begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{k}{m_{int}} & \frac{-k}{m_{int}} \\ 0 & \frac{-2k}{m} & \frac{2k}{m} \end{bmatrix}\right)\begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix} - \frac{\Delta t^2}{2}\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \\
&= \begin{bmatrix} u_0 \\ 0 \\ 0 \end{bmatrix}
\end{aligned} \tag{6.18}$$

If consistent initial conditions are used (i.e. the displacement at time step -1 is approximated using a Taylor expansion of u about time step 0) the displacement of the first node at time step -1 is u_0 and the displacement of node 1 at time step 0 is also u_0. This shows that using consistent initial conditions does not calculate the displacements of the nodes correctly in order to initialise the central difference method. The spring does not deform and therefore does not exert a force on the dashpot resulting in an absence of the relaxation behaviour that is expected.

This investigation shows that the orientation of the spring and dashpot influences how the solution is implemented. One way in which the issue with the Spring-Dashpot orientation of the Maxwell element can be fixed is by using inconsistent starting conditions. This means that the displacement at time step -1, equation (6.5), is not calculated and the displacement $\boldsymbol{u}_{-1}$ is set to zero. Implementing inconsistent initial conditions allows the system to be initialised with the displacement of node 1 having changed from 0 at time step -1 to u_0 at time step 0, and therefore the individual components can react accordingly.

6.2.2 Discrete Hereditary Integral Solution

The introduction of an intermediate mass to track the displacement of the internal spring-dashpot interface has two main issues. The first is the near doubling of the degrees of freedom. The second is that the intermediate mass does not have as good an analogue in higher dimensions due to the extra degree of freedom. There is therefore a need to formulate a solution to the Maxwell element that does not require the tracking of the node at the interface between the spring and dashpot. This was presented in Chapter 3.2.6 as the hereditary integral and the implementation is outlined below.

Implementation of the Hereditary Integral

The hereditary integral developed in Chapter 3.2.6 can be reformulated to the increment in force required to take $F(t)$ to $F(t+\Delta t)$. This step is required for time-discrete implementations. Writing equation (3.34) at time $t+\Delta t$ gives

$$\begin{aligned} F(t+\Delta t) &= \int_0^{t+\Delta t} e^{\frac{k}{\eta}(\tau-(t+\Delta t))} k\dot{u}(\tau)d\tau \\ &= e^{-\frac{k}{\eta}\Delta t} \int_0^{t+\Delta t} e^{-\frac{k}{\eta}(t-\tau)} k\dot{u}(\tau)d\tau \ . \end{aligned} \tag{6.19}$$

This integral can now be split into the response up to time t and the response over the increment t to $t+\Delta t$, which we refer to as Δt

$$F(t+\Delta t) = e^{-\frac{k}{\eta}\Delta t}\left[\int_0^t e^{-\frac{k}{\eta}(t-\tau)} k\dot{u}(\tau)d\tau + \int_t^{t+\Delta t} e^{-\frac{k}{\eta}(t-\tau)} k\dot{u}(\tau)d\tau\right]. \tag{6.20}$$

Assuming that the change of velocity is negligible over small time increments Δt, such that $\dot{u}(\tau) \approx \dot{u}(t+\Delta t)$, this gives

$$\begin{aligned} F(t+\Delta t) &\approx e^{-\frac{k}{\eta}\Delta t}\left[F(t) + k\dot{u}(t)\int_t^{t+\Delta t} e^{-\frac{k}{\eta}(t-\tau)}d\tau\right] \\ &= e^{-\frac{k}{\eta}\Delta t}\left[F(t) + k\dot{u}(t)\left(\frac{\eta}{k}e^{-\frac{k}{\eta}(t-\tau)}\Big|_t^{t+\Delta t}\right)\right] \\ &= e^{-\frac{k}{\eta}\Delta t}\left[F(t) + \eta\dot{u}(t)\left(e^{\frac{k}{\eta}\Delta t} - 1\right)\right] \\ &= \eta\dot{u}(t)\left(1 - e^{-\frac{k}{\eta}\Delta t}\right) + F(t)e^{-\frac{k}{\eta}\Delta t}. \end{aligned} \tag{6.21}$$

In order to analyse the dynamic behaviour within a material, modelled using a Maxwell element, we introduce spatial discretisation. This leads to a multi-element discrete model where the hereditary integral is used to relate F to u, discussed in Chapter 3.2.6 for the Maxwell element and implemented similarly to the multi-element intermediate mass solution above. The difference is that the force of the Maxwell element is now stored in an internal force vector, $\boldsymbol{\mathcal{F}}$, instead of a stiffness and damping matrix. This eliminates the issue of spring and dashpot orientation, and is instead replaced with an internal force initialisation step.

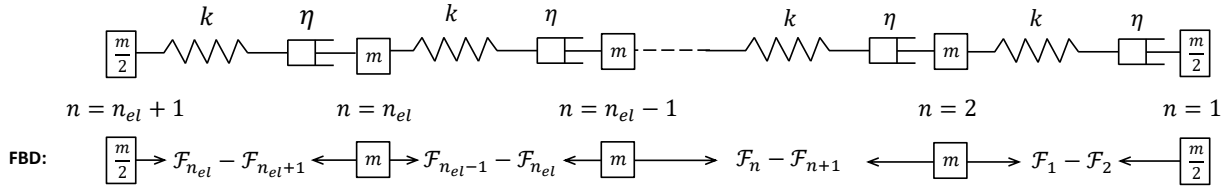


Figure 6.5: Multi-element discrete hereditary integral for Maxwell element, showing internal forces acting on nodal masses.

The force of the Maxwell element is implemented in an internal force vector. This tracks the effect of the internal force in each element on the nodes. The internal force at each node $\mathcal{F}_n$ is defined as

$$\mathcal{F}_n(t) = \eta \left(1 - e^{-\frac{k}{\eta}\Delta t}\right) \dot{u}_n(t) + \left(e^{-\frac{k}{\eta}\Delta t}\right) \mathcal{F}_n(t - \Delta t) \tag{6.22}$$

where $\dot{u}_n(t)$ is the velocity at the node, n, at time t.

The equation of motion for the system shown in Figure 6.5 would be

$$\begin{bmatrix} \frac{m}{2} & 0 & & 0 & 0 & 0 \\ 0 & m & & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & m & 0 & 0 \\ 0 & 0 & & 0 & m & 0 \\ 0 & 0 & & 0 & 0 & \frac{m}{2} \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \vdots \\ \ddot{u}_{n_{el}-1} \\ \ddot{u}_{n_{el}} \\ \ddot{u}_{n_{el}+1} \end{bmatrix} = - \begin{bmatrix} (\mathcal{F}_1 - \mathcal{F}_2) \\ (\mathcal{F}_2 - \mathcal{F}_3) - (\mathcal{F}_1 - \mathcal{F}_2) \\ \vdots \\ (\mathcal{F}_{n_{el}-1} - \mathcal{F}_{n_{el}}) - (\mathcal{F}_n - \mathcal{F}_{n+1}) \\ (\mathcal{F}_{n_{el}} - \mathcal{F}_{n_{el}+1}) - (\mathcal{F}_{n_{el}-1} - \mathcal{F}_{n_{el}}) \\ -(\mathcal{F}_{n_{el}} - \mathcal{F}_{n_{el}+1}) \end{bmatrix}. \tag{6.23}$$

This is of the form

$$\boldsymbol{M}\ddot{\boldsymbol{u}} = -\boldsymbol{\mathcal{F}}. \tag{6.24}$$

The displacement would then be updated by

$$\boldsymbol{u}_{t+1} = \left[\frac{1}{\Delta t^2}\boldsymbol{M}\right]^{-1} \left(-\boldsymbol{\mathcal{F}}_t + \left[\frac{2}{\Delta t^2}\boldsymbol{M}\right] \boldsymbol{u}_t + \left[-\frac{1}{\Delta t^2}\boldsymbol{M}\right] \boldsymbol{u}_{t-1}\right). \tag{6.25}$$

6.3 Maxwell Element (with Non-Linear Dashpot)

In order for the material model to capture the behaviour seen in the ISR regime, an implementation of a non-linear Maxwell dashpot is needed. There are many ways of considering non-linearity with respect to the dashpot. Due to the strain rate dependency of cortical bone, non-linearity with respect to the strain rate or velocity using a power relationship is considered. The Maxwell dashpot can be considered to be non-linearly related to the velocity by raising the velocity to a power p such that the force of the dashpot is

$$F_\eta = \bar{\eta}\, \dot{u}^p \tag{6.26}$$

where the units of $\bar{\eta}$ account for the value of p.

Alternatively the coefficient, η, can be defined as a function of the velocity, such that the force of the dashpot is

$$F_\eta = \eta(\dot{u})\, \dot{u}\ . \tag{6.27}$$

The function, $\eta(\dot{u})$, can be chosen in such a way as to give the same force in equation (6.26). This would be

$$\eta(\dot{u}) = \bar{\eta}\,\dot{u}^{p-1}, \tag{6.28}$$

and the force would therefore be

$$\begin{aligned} F_\eta &= \bar{\eta}\,\dot{u}^{p-1}\,\dot{u} \\ &= \bar{\eta}\,\dot{u}^{p}. \end{aligned} \tag{6.29}$$

6.3.1 Intermediate Mass Implementation

A formulation for a non-linear Maxwell element using the intermediate mass implementation was carried out. As was discussed in Chapter 6.2.1, the orientation issue in the intermediate mass implementation of a linear Maxwell element could be fixed using inconsistent initial conditions which set the displacement of node 1 at time step -1 to zero and the displacement at time step 0 to u_0. As the size of the time step decreases, the velocity of the element over the time step -1 tends towards infinity. The force of the dashpot is dependent on velocity and thus the initial force tends to infinity as the time step decreases. This force is exerted on the spring which displaces almost instantaneously as expected.

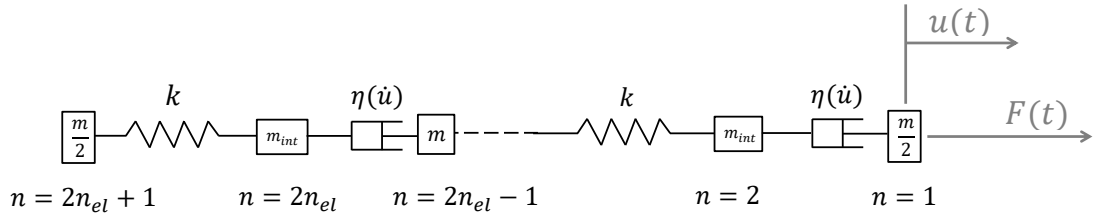


Figure 6.6: The arrangement of the multi-element discrete Maxwell element, with a non-linear dashpot.

When dealing with a non-linear dashpot in a Maxwell element, the force response is proportional to the velocity to some power, as per equation (6.26). For values of $p > 1$ the effect of using inconsistent initial conditions is increased. However, for values of $p < 1$, the effect of using inconsistent initial conditions is diminished. This is because the initial force of the dashpot will be decreased due to the inverse relationship with velocity for values of $p < 1$. As the velocity tends towards infinity, the initial force response tends towards zero and so the inconsistent initial conditions are no longer effective.

The use of inconsistent initial conditions is not effective with a non-linear dashpot and therefore the orientation of the Maxwell element is still an issue. There are two ways of dealing with this issue which do not depend on inconsistent initial conditions. The first is to determine the orientation of the element by looking at the stiffness matrix, and then applying initial conditions to the first and second nodes accordingly.

The following stiffness matrix, $\boldsymbol{K}$ shows that there is not a spring attached to the boundary node (node 1 in this case) that the displacement will be applied to,

$$\boldsymbol{K} = \begin{bmatrix} 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & k & -k & & 0 & 0 & 0 \\ 0 & -k & k & & 0 & 0 & 0 \\ & & & \ddots & & & \\ 0 & 0 & 0 & & 0 & 0 & 0 \\ 0 & 0 & 0 & & 0 & k & -k \\ 0 & 0 & 0 & & 0 & -k & k \end{bmatrix}. \tag{6.30}$$

This is indicated by the zeroes in row 1 and column 1. The system defined by the stiffness matrix above would require initial conditions to be applied to the first and second nodes. For example, if node one of this system was given a prescribed displacement, the initial displacement would need to be applied to both nodes, one and two.

The second is to ensure that a Maxwell dashpot is not connected to a boundary node. This can be done by attaching a fictitious element, a spring with a near infinite stiffness, to the Maxwell dashpot.

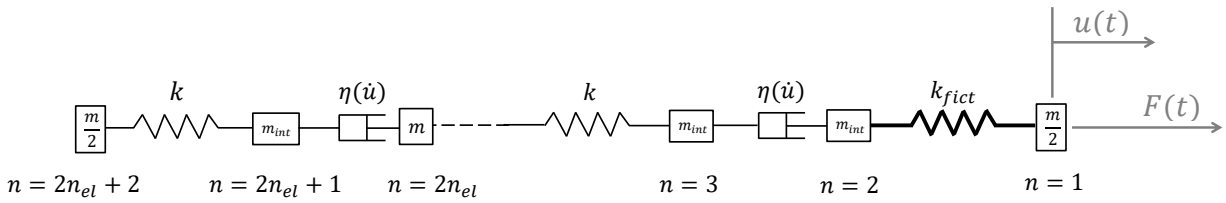


Figure 6.7: The arrangement of the multi-element discrete Maxwell element, with non-linear dashpot and a fictitious element.

By making the stiffness of the fictitious element relatively high, but not too high as this leads to numerical round-off error, the forces applied to the boundary node are transmitted through the fictitious element. This effectively allows the fictitious element to apply the necessary forces on the dashpot as defined by the boundary conditions.

6.3.2 Velocity Raised to a Power

Intermediate Mass Implementation

The non-linearity of the Maxwell element can be achieved by raising the velocity to a power, using the local velocity over the dashpot in the Maxwell element. This required the central difference method to be updated. Using the modified central difference method that was discussed in Chapter

6.1, updating the method was relatively simple due to the velocity term being independent of the u_{t+1} term. The velocity term, equation (6.7), was raised to a power, p, as follows

$$\dot{\boldsymbol{u}}_t^p \approx \dot{\boldsymbol{u}}_{t-\frac{1}{2}}^p = \frac{1}{\Delta t^p} \left(\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\right)^p . \tag{6.31}$$

This was substituted into the equation of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{C}\dot{\boldsymbol{u}}^p + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \ , \tag{6.32}$$

and using the acceleration $\ddot{\boldsymbol{u}}$ as defined in equation (6.3) gives the update of displacement as

$$\boldsymbol{u}_{t+1} = \left[\frac{1}{\Delta t^2}\boldsymbol{M}\right]^{-1} \left(\boldsymbol{F}_t + \left[\frac{2}{\Delta t^2}\boldsymbol{M} - \boldsymbol{K}\right]\boldsymbol{u}_t + \left[-\frac{1}{\Delta t^2}\boldsymbol{M}\right]\boldsymbol{u}_{t-1} - \frac{1}{\Delta t^p}\boldsymbol{C}\left(\boldsymbol{u}_t - \boldsymbol{u}_{t-1}\right)^p\right) . \tag{6.33}$$

6.3.3 Dashpot Coefficient Function

The second implementation of non-linearity that was investigated was the dashpot coefficient as a function of velocity. When implementing the dashpot coefficient function, several different functions for η were considered, and two were chosen to investigate the effect of this function on the behaviour of the non-linear Maxwell element. One of the functions was a simple power law equation, and the other included a linear portion

$$\eta(\dot{u}(t)) = \bar{\eta} \tag{6.34a}$$

$$\eta(\dot{u}(t)) = \bar{\eta}\,\dot{u}^{p-1} \tag{6.34b}$$

$$\eta(\dot{u}(t)) = \frac{1}{2}\bar{\eta}\left(1 + \dot{u}^{p-1}\right) \tag{6.34c}$$

where equations (6.34a), (6.34b) and (6.34c) reduce to the linear Maxwell element when $p = 1$.

Equations (6.34b) and (6.34c) are limited to values of $p \geq 1$ as $\eta(\dot{u}(t))$ becomes overly large for velocities near zero when $p < 1$. One way in which this can be accounted for is to add a small value $\dot{u}_\epsilon$ to the velocity before raising it to a power such that equations (6.34b) and (6.34c) would be

$$\eta(\dot{u}(t)) = \bar{\eta}\ \left(\dot{u} + \dot{u}_\epsilon\right)^{p-1} \tag{6.35a}$$

$$\eta(\dot{u}(t)) = \frac{1}{2}\bar{\eta}\left(1 + \left(\dot{u} + \dot{u}_\epsilon\right)^{p-1}\right) \tag{6.35b}$$

Intermediate Mass Implementation

The method of having the damping coefficient as a function of velocity was used in the intermediate mass implementation by recalculating the damping matrix $\boldsymbol{C}$ in each time step based on the velocity at the end of the previous time step. This could be done using either the velocity over each element

or the velocity over the dashpot, as the internal node is being tracked for this implementation. The assumption is made that η is constant over the time step.

Discrete Hereditary Integral Model

The dashpot coefficient function lends itself well to the use of the linear Maxwell hereditary integral since we assume that η is constant across a time step and allows the use of the hereditary integral across that time step. The formulation of this hereditary integral is taken directly from that of the linear Maxwell element, equation (6.21), replacing the coefficient terms, η, with a function of velocity, $\eta(\dot{u}(t))$, as follows

$$F(t+\Delta t) = \eta(\dot{u}(t))\dot{u}(t)\left(1 - e^{-\frac{k}{\eta(\dot{u}(t))}\Delta t}\right) + F(t)e^{-\frac{k}{\eta(\dot{u}(t))}\Delta t} \tag{6.36}$$

where t is the time at the start of the time step, and $t+\Delta t$ is the time at the end of the time step.

6.4 Incremental Time-stepping Scheme

Commercial finite element software necessitated the use of an alternative approach to the internal force vector, as discussed in Chapter 6.2.2. Only the strain (or stretch) increments are available to be used in a user-defined material model. The force increment over a single time step is calculated using the stretch increment and this force increment is added to the total force up to the beginning of that time step. This calculation and addition of the force increment is carried out in each time step and this approach will be referred to as the incremental time-stepping scheme.

$$\mathcal{F}(t) = \mathcal{F}(t-\Delta t) + \mathcal{F}(\Delta t) \tag{6.37}$$

6.4.1 Linear Elastic Element

To demonstrate how to analyse a system using the incremental time-stepping scheme, we will first look at a lumped mass linear elastic spring-mass system. A spring-mass system with internal forces is shown in Figure 6.8, where $\mathcal{F}_n^E$ is the internal elastic force due to the spring, k, in element n.

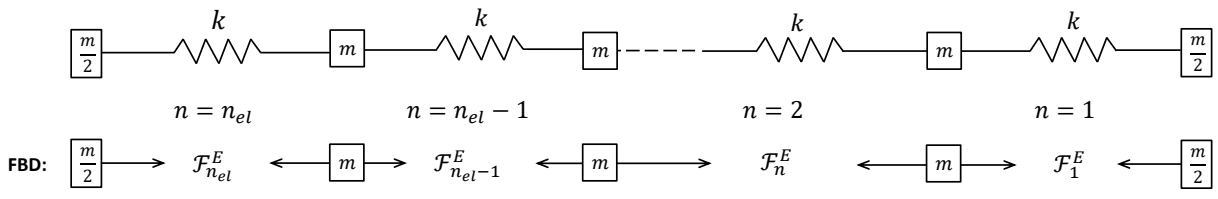


Figure 6.8: Multi-element linear elastic element, showing internal forces acting on nodal masses.

The equation of motion of the above multi-element spring-mass system is

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{K}\boldsymbol{u} = \boldsymbol{f} \ . \tag{6.38}$$

The free body diagram in Figure 6.8 shows that the forces acting on the nodes is calculated using an internal force $\mathcal{F}^E$, this internal force replaces the $\boldsymbol{K}\boldsymbol{u}$ term in the equation of motion (6.38) giving a new equation of motion

$$\boldsymbol{M}\ddot{\boldsymbol{u}} + \boldsymbol{\mathcal{F}}^E = \boldsymbol{f} \ . \tag{6.39}$$

The equation of motion (6.39) is similar to that of the linear Maxwell element when we considered an internal force vector in equation (6.24) with the applied force $\boldsymbol{f}$ equal to zero. The difference is the internal force vector is made up of the force over the element as opposed to the difference between the forces acting on the nodes

$$\begin{bmatrix} \frac{m}{2} & 0 & & 0 & 0 & 0 \\ 0 & m & & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & m & 0 & 0 \\ 0 & 0 & & 0 & m & 0 \\ 0 & 0 & & 0 & 0 & \frac{m}{2} \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \vdots \\ \ddot{u}_{n_{el}-1} \\ \ddot{u}_{n_{el}} \\ \ddot{u}_{n_{el}+1} \end{bmatrix} = - \begin{bmatrix} \mathcal{F}_1^E \\ \mathcal{F}_2^E - \mathcal{F}_1^E \\ \vdots \\ \mathcal{F}_{n+1}^E - \mathcal{F}_n^E \\ \mathcal{F}_{n_{el}}^E - \mathcal{F}_{n_{el}-1}^E \\ -\mathcal{F}_{n_{el}}^E \end{bmatrix} = -\boldsymbol{\mathcal{F}}^E \tag{6.40}$$

where the internal force in each element is calculated from the stretch increment of the element and updated in each time step as

$$\mathcal{F}_n^E(t) = \mathcal{F}_n^E(t - \Delta t) + k\ \lambda_n(\Delta t) \ , \tag{6.41}$$

and the stretch increment is defined as the change in length of an element over a time step Δt

$$\begin{aligned} \lambda_n(\Delta t) &= [l_0 + u_n(\Delta t)] - [l_0 + u_{n+1}(\Delta t)] \\ &= u_n(\Delta t) - u_{n+1}(\Delta t) \\ &= [u_n(t) - u_n(t - \Delta t)] - [u_{n+1}(t) - u_{n+1}(t - \Delta t)] \end{aligned} \tag{6.42}$$

where l_0 is the original length of the element.

6.4.2 Linear Maxwell Element

The diagram of a lumped mass multi-element Maxwell system is shown in Figure 6.9, where $\mathcal{F}_n^M$ is the internal force due to the Maxwell element, M, in element n.

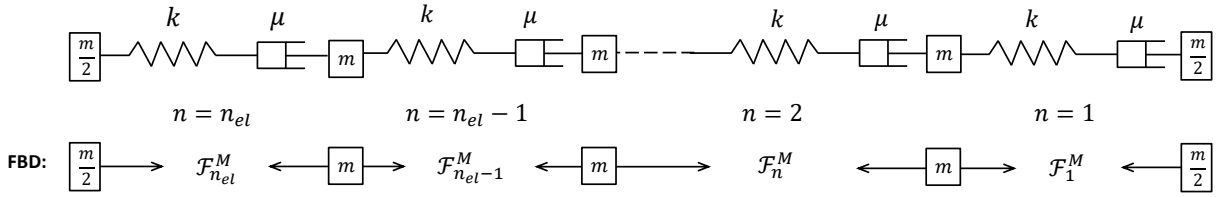


Figure 6.9: Multi-element discrete hereditary integral for Maxwell element, showing internal forces acting on nodal masses.

Similar to the linear elastic system, the equation of motion would be

$$\begin{bmatrix} \frac{m}{2} & 0 & & 0 & 0 & 0 \\ 0 & m & & 0 & 0 & 0 \\ & & \ddots & & & \\ 0 & 0 & & m & 0 & 0 \\ 0 & 0 & & 0 & m & 0 \\ 0 & 0 & & 0 & 0 & \frac{m}{2} \end{bmatrix} \begin{bmatrix} \ddot{u}_1 \\ \ddot{u}_2 \\ \vdots \\ \ddot{u}_{n_{el}-1} \\ \ddot{u}_{n_{el}} \\ \ddot{u}_{n_{el}+1} \end{bmatrix} = - \begin{bmatrix} \mathcal{F}_1^M \\ \mathcal{F}_2^M - \mathcal{F}_1^M \\ \vdots \\ \mathcal{F}_{n+1}^M - \mathcal{F}_n^M \\ \mathcal{F}_{n_{el}}^M - \mathcal{F}_{n_{el}-1}^M \\ -\mathcal{F}_{n_{el}}^M \end{bmatrix} = -\boldsymbol{\mathcal{F}}^M. \tag{6.43}$$

This internal force vector can be updated in different ways. The first is simply to update the internal force based on the stretch and stretch rate of an element. The second is to update the stress in the element based on the strain (engineering or true strain). The stress would then be multiplied by the cross-sectional area of the element to obtain the force to calculate the internal force vector and subsequently the displacement.

If the internal force is now defined for each element, n, and is calculated from the stretch rate, $\dot{\lambda}_n$, of the element then

$$\mathcal{F}_n^M(t) = \eta \left(1 - e^{-\frac{k}{\eta}\Delta t}\right) \dot{\lambda}_n(t) + \left(e^{-\frac{k}{\eta}\Delta t}\right) \mathcal{F}_n(t - \Delta t). \tag{6.44}$$

The stretch rate of the element is determined by the stretch increment of the element divided by the time step

$$\begin{aligned} \dot{\lambda}_n(t) &= \frac{1}{\Delta t} \left[\lambda_n(t) - \lambda_n(t - \Delta t)\right] \\ &= \frac{1}{\Delta t} \lambda_n(\Delta t) \end{aligned} \tag{6.45}$$

where the stretch increment is defined in equation (6.42).

If the internal force is calculated from the engineering strain rate of the element, $\dot{\varepsilon}_n$, and the stress

in the previous time step then

$$\mathcal{F}_n^M(t) = A(t)\left[\eta\left(1 - e^{-\frac{E}{\eta}\Delta t}\right)\dot{\varepsilon}_n(t) + \frac{1}{A(t)}\left(e^{-\frac{E}{\eta}\Delta t}\right)\mathcal{F}_n(t - \Delta t)\right] \tag{6.46}$$

where $A(t)$ is the cross-sectional area of the element at time t to convert the updated stress into force, and E is the Young's modulus of the material, which is defined as

$$E = \frac{kl}{A_0} \tag{6.47}$$

where k is the elastic stiffness, l is the length of the element, and A_0 is the initial cross-sectional area.

The engineering strain rate of the element is determined by the engineering strain increment of the element divided by the time step approximated by the backward-difference

$$\begin{aligned}\dot{\varepsilon}_n(t) &= \frac{1}{\Delta t}\left[\varepsilon_n(t) - \varepsilon_n(t - \Delta t)\right] \\ &= \frac{1}{\Delta t}\varepsilon_n(\Delta t)\end{aligned} \tag{6.48}$$

where $\varepsilon_n(\Delta t)$ is the engineering strain increment and is defined as the stretch increment divided by the original length

$$\varepsilon_n(\Delta t) = \frac{\left[u_n(t) - u_n(t - \Delta t)\right] - \left[u_{n+1}(t) - u_{n+1}(t - \Delta t)\right]}{l_0} \tag{6.49}$$

6.4.3 Non-Linear Maxwell Element

The non-linear Maxwell element, where the coefficient of the dashpot is a function of strain rate, lends itself well to the incremental time-stepping scheme above. In equations (6.44) and (6.46), the coefficient η would simply be replaced by a function of the stretch or strain rate, similar to the functions in equations (6.34b) and (6.34c).

Chapter 7

Results of One-Dimensional Numerical Models

Several validation tests on the one-dimensional models were performed. The first set of validation tests were for the intermediate mass implementation of a linear Maxwell model with a comparison to the discrete hereditary integral solution, and an investigation into the effect of the size of the intermediate mass. The next set of validation tests were for the different approaches to the non-linear Maxwell element, the velocity raised to a power and the dashpot coefficient function. The use of the velocity over the dashpot and the velocity over the element were explored for the dashpot coefficient function in the intermediate mass implementation, as well as the non-linear hereditary integral solution. This chapter will show an overview of the results of these tests, and the discussion of these results will take place in Chapter 8.

7.1 Maxwell Element (with Linear Dashpot)

The linear Maxwell element was investigated using the intermediate mass implementation and the discrete hereditary integral. A prescribed displacement was applied to the Maxwell element in order to verify that the resulting behaviour matched that outlined in Chapter 6.2.1. An investigation was then carried out on the effect of the size of the intermediate mass for a linear Maxwell element.

7.1.1 Intermediate Mass Implementation

The intermediate mass implementation was verified using a single Maxwell element to determine that the expected relaxation behaviour was achieved as was seen in Figure 6.3. Each test that was

carried out used a single linear Maxwell element that was fixed at one end and the other end was given a prescribed displacement of 0.05 m. The Maxwell element used in each test had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a dashpot with viscous coefficient $\eta = 21.2057\,\mathrm{N\,s\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$ and an intermediate mass size $m_{int} = 7.854 \times 10^{-5}\,\mathrm{kg}$. The time step for each test was calculated using the critical time step based on the elastic portion of the model and reduced by a factor. The size of the time step used was in the region of $10^{-}7\,\mathrm{s}$.

Dashpot-Spring

The first test was a single Maxwell element in the Dashpot-Spring orientation, see Figure 7.1, where node 3 was fixed and a prescribed displacement $u_1(t)$ was applied to node 1.

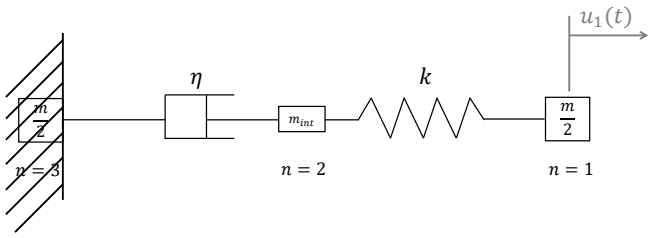


Figure 7.1: Maxwell element in a Dashpot-Spring orientation with fixed boundary conditions and prescribed displacement.

The displacement graphs are shown in Figure 7.2, where the blue line in each graph represents the prescribed displacement of node 1 to 0.05 m, and the yellow line represents node 3 which shows a displacement of 0 m for the duration of the simulation. The orange line represents node 2, which starts at a displacement of zero and moves towards node 1 until they have the same displacement. This relaxation behaviour represents the dashpot reacting to the force of the spring as the spring contracts and pulls node 2 towards node 1, and stretches the dashpot until the spring has regained its original dimensions.

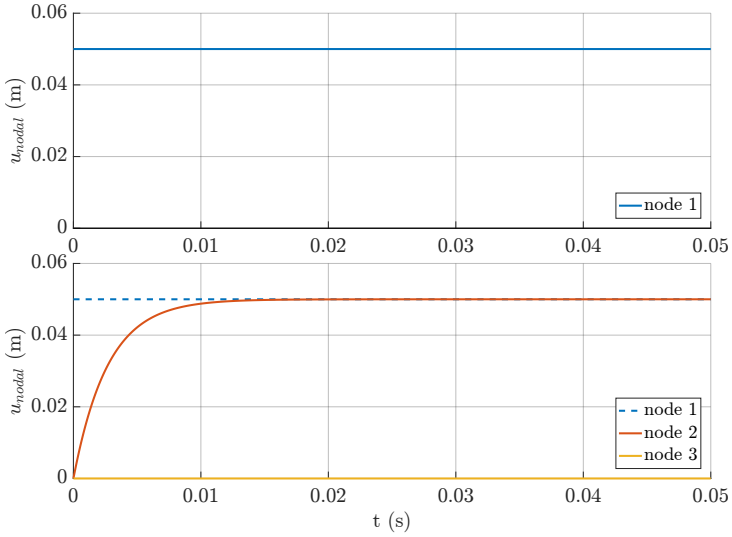


Figure 7.2: Displacement-time graphs of a single Maxwell element in a Dashpot-Spring arrangement given a prescribed displacement of 0.05 m.

For the above simulation, the forces acting on the spring and on the dashpot were plotted against time and the force acting on the intermediate mass was plotted on the other graph, shown in Figure 7.3. From a displacement of 0.05 m and a spring stiffness of 7853.9816 N m^{-1}, the expected force of the spring was 392.6991 N. This is seen in the first graph in the figure below by the orange and blue lines with a maximum force of approximately 400 N at time 0 s. As the element undergoes relaxation behaviour, the forces acting on the spring and dashpot decrease over time and they are equal for the spring and the dashpot, as expected. The force acting on the intermediate mass, despite an initial spike at t equal to zero, is negligible.

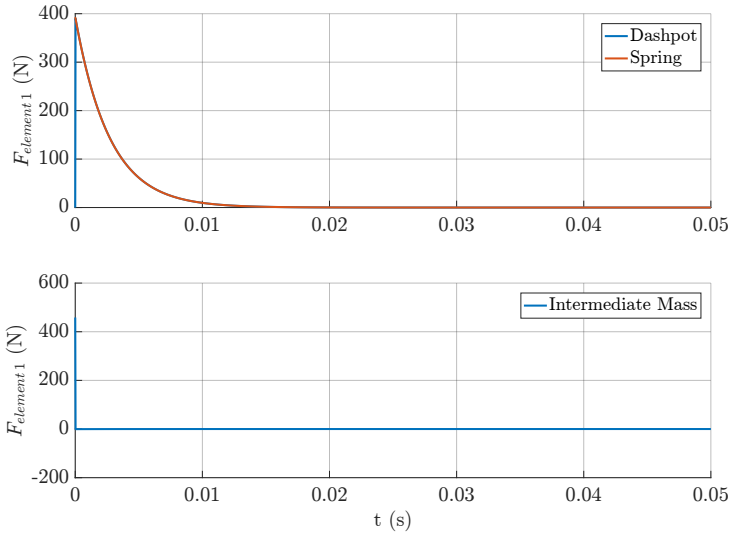


Figure 7.3: Force comparison graphs of a single Maxwell element in a Dashpot-Spring arrangement given a prescribed displacement of 0.05 m.

Spring-Dashpot

The next test was a single Maxwell element in the Spring-Dashpot orientation, see Figure 7.4. Again a prescribed displacement $u_1(t)$ was applied to node 1, node 3 was fixed while node 2 was tracked over time. This was done using consistent initial conditions.

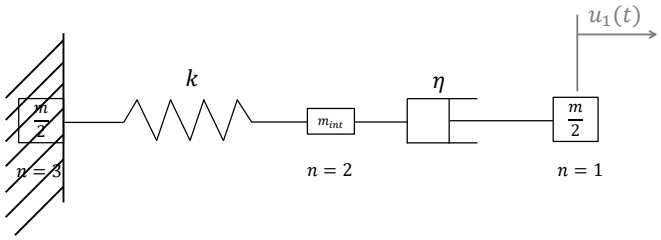


Figure 7.4: Maxwell element in a Dashpot-Spring orientation with fixed boundary conditions and prescribed displacement.

The displacements of the three nodes can be seen in Figure 7.5, where most notably node 2 did not displace. This is due to the orientation of the element producing a starting condition of zero as discussed in Chapter 6.2.1.

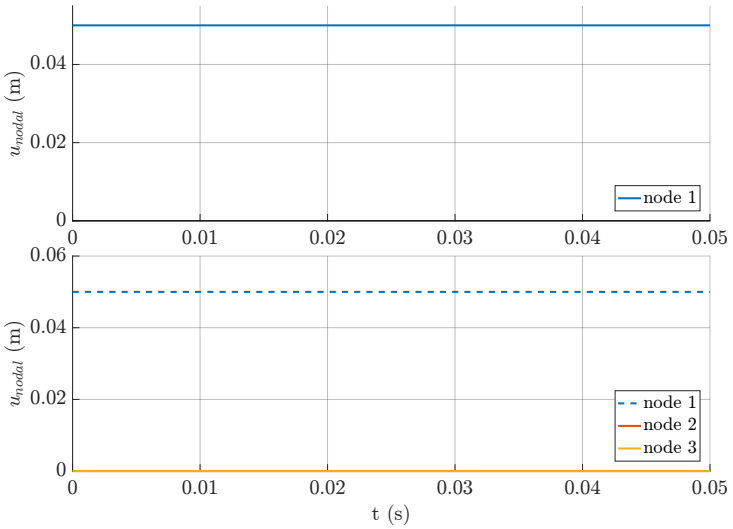


Figure 7.5: Displacement-time graphs of a single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m and using consistent initial conditions.

For this simulation, the forces acting on the spring and dashpot were plotted against time and the force acting on the intermediate mass was plotted on another graph. Figure 7.6 shows that there is no force as expected since there was no movement of the nodes as discussed in Chapter 6.2.1.

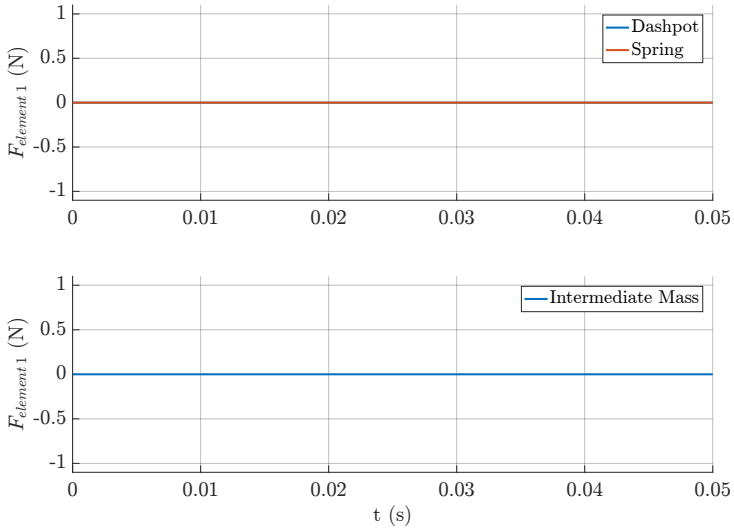


Figure 7.6: Force comparison graphs of a single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m and using consistent initial conditions.

The same prescribed displacement test was run on the Spring-Dashpot arrangement instead using inconsistent initial conditions, outlined in Chapter 6.2.1. The internal node, node 2, was tracked over time. In Figure 7.7, node 1 and node 3 can be seen to have displacements of 0.05 m and 0 m respectively, as was seen before. The orange line shows node 2 has displaced with node 1 at time t equal to zero (instantaneously), and shows the expected relaxation behaviour over time.

This represents the dashpot translating initially and the spring stretching by the prescribed displacement. The dashpot then reacts to the force of the spring as the spring contracts and pulls node 2 towards node 3 and stretches the dashpot until the spring has regained its original dimensions. This shows that the inconsistent initial conditions are able to provide the necessary starting conditions for the linear Maxwell element in this orientation.

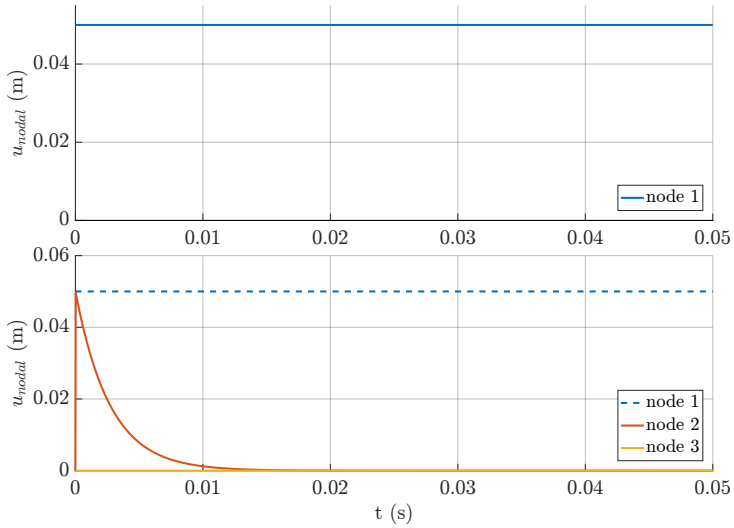


Figure 7.7: Displacement-time graphs of a single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m and using inconsistent initial conditions.

The forces acting on the spring and dashpot were plotted for this simulation, as well as the force acting on the intermediate mass. Figure 7.8 shows that the forces acting on the spring and dashpot are the same and follow a relaxation behaviour as was expected. The force acting on the intermediate mass is again negligible.

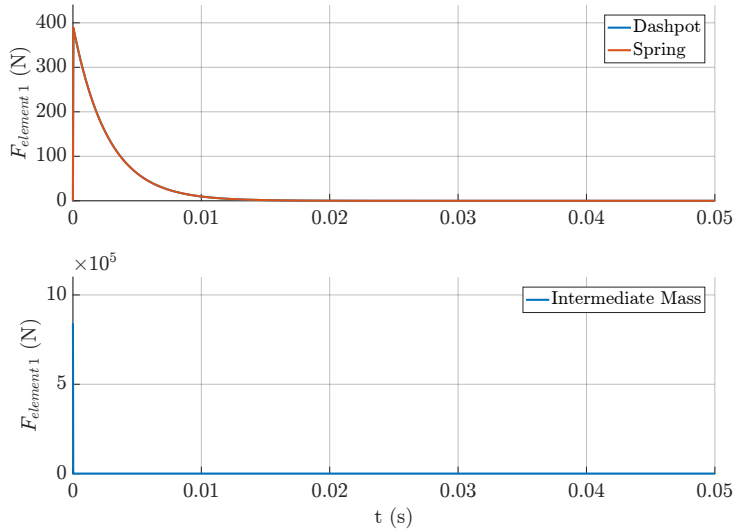


Figure 7.8: Force comparison graphs of a single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m and using inconsistent initial conditions.

7.1.2 Discrete Hereditary Integral Solution

The discrete hereditary integral was then tested using a single Maxwell element, see Figure 7.9. The hereditary integral solution does not have an internal node to track and therefore only the force of the element was plotted and compared to the intermediate mass implementation. The test that was carried out used the same variables as those in the intermediate mass implementation tests. The element was fixed at one end and the other end was given a prescribed displacement of 0.05 m. The Maxwell element used had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a dashpot with viscous coefficient $\eta = 21.2057\,\mathrm{N\,s\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$ and an intermediate mass size $m_{int} = 7.854 \times 10^{-5}\,\mathrm{kg}$.

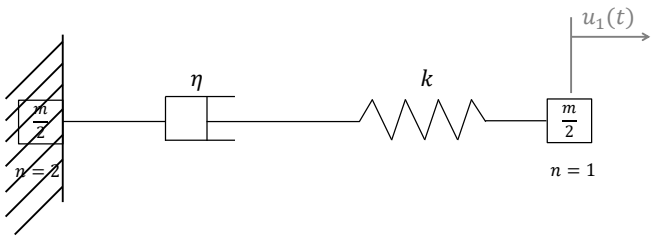


Figure 7.9: Maxwell element modelled by the hereditary integral with fixed boundary conditions and prescribed displacement.

For the test outlined above, the force from the hereditary integral was plotted on the same axis as the force acting on the spring and dashpot for the intermediate mass implementation using the Dashpot-Spring arrangement and using the Spring-Dashpot arrangement using inconsistent initial conditions, shown in Figure 7.10.

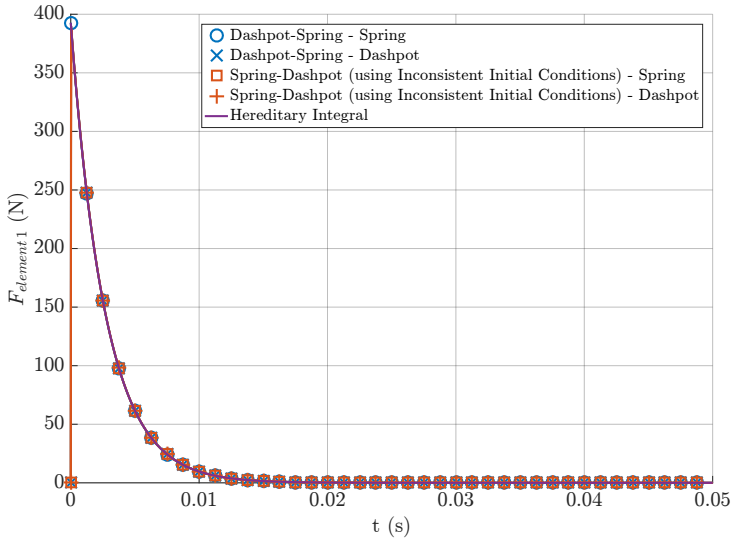


Figure 7.10: Intermediate mass implementation (Dashpot-Spring and Spring-Dashpot arrangements) and hereditary integral displacement comparison graphs for a single linear Maxwell element given a prescribed displacement of 0.05 m.

This shows the same force curve for the spring and dashpot as before, with the force of the hereditary integral tracking the intermediate mass implementation results closely. The force acting on the spring and dashpot in the Spring-Dashpot arrangement using inconsistent initial conditions match the force of the hereditary integral, confirming that using inconsistent initial conditions produces the exact desired result.

7.1.3 Intermediate Mass Investigation

An investigation into the effect of the choice of magnitude of the intermediate mass was carried out using a multi-element linear Maxwell in parallel with a Kelvin-Voigt spring to stabilise the model, shown in Figure 7.11, known as the Standard Linear Model. A free bar (no boundary constraints) test was carried out on a row of one hundred elements and an impulse force $F_1(t)$ was applied to node 1.

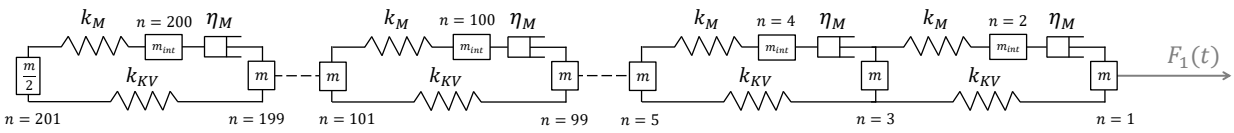


Figure 7.11: A row of one hundred Standard Linear Model elements with no boundary constraints and an applied force.

The impulse force follows a ramped S-shape curve from zero to a maximum value of 10 N for the first 0.004 s and then follows the inverted S-shape curve for the following 0.004 s until its value returns to zero, shown in the first graph in Figure 7.12. Each Standard Linear element has a Maxwell spring with stiffness $k_M = 785\,398.16\,\mathrm{N\,m^{-1}}$, a Maxwell dashpot with viscous coefficient $\eta_M = 10.0\,\mathrm{N\,s\,m^{-1}}$, a Kelvin-Voigt spring with stiffness $k_{KV} = 785\,398.16\,\mathrm{N\,m^{-1}}$ and a mass of $m = 7.854 \times 10^{-5}\,\mathrm{kg}$. Varying sizes of the intermediate mass m_{int} were investigated and their size is quoted as a factor of the nodal masses m where the relationship m_{fact} is defined as

$$m_{fact} = \frac{m}{m_{int}}. \tag{7.1}$$

Figure 7.12 shows the force acting on the middle element (element 50) of the multi-element simulation for an intermediate mass factor m_{fact} of 10. There are a few factors that can be extracted from these graphs for varying values of m_{fact}. The first of these is the difference between the forces acting on the spring and the dashpot. We would expect this difference to become smaller as the size of m_{fact} increases (the size of m_{int} becomes smaller). The second factor that can be tracked is the size of the force acting on the intermediate mass relative to the force acting on the spring. This ratio is expected to decrease as m_{fact} increases.

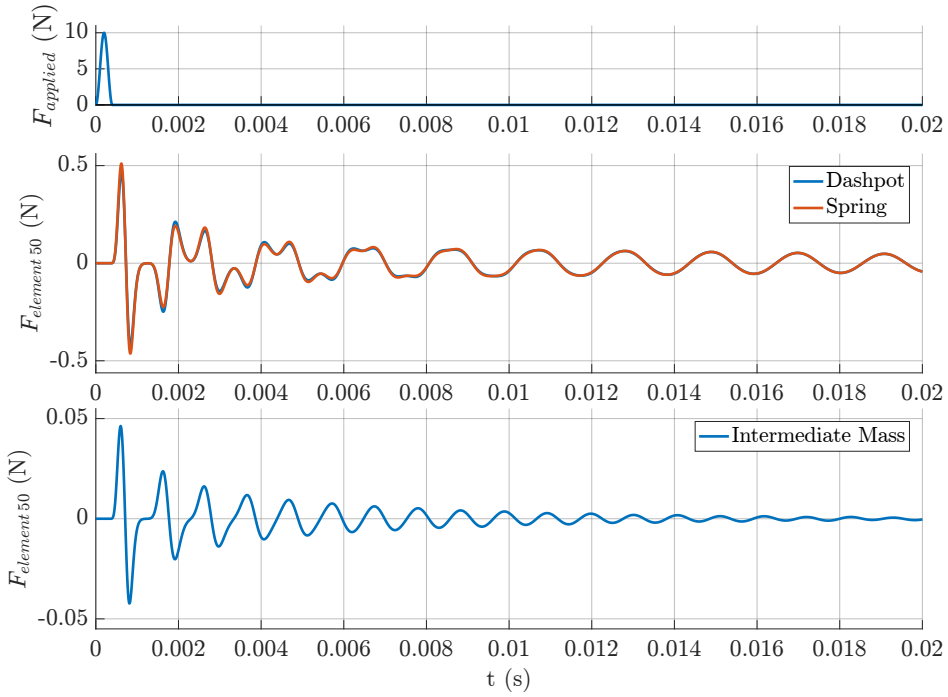


Figure 7.12: Force comparison graphs for a multi-element, free bar, Standard Linear model given an impulse force for an intermediate mass investigation.

For a decreasing intermediate mass size and increasing intermediate mass factor, m_{fact}, the percentage error was calculated as the maximum difference between the absolute amplitude of the force acting on the spring and the absolute amplitude of the force acting on the dashpot as a

percentage of the average amplitude of the two.

$$\%Error = max\left(\frac{abs\left(A_{MS}\right) - abs\left(A_{MD}\right)}{A_{ave}}\right) \times 100 \tag{7.2}$$

where A_{MS} is the amplitude of the force acting on the Maxwell spring, A_{MD} is the amplitude of the force acting on the Maxwell dashpot and A_{ave} is the average of the absolute amplitude of the forces acting on the spring and dashpot

$$A_{ave} = \frac{abs\left(A_{MS}\right) + abs\left(A_{MD}\right)}{2}. \tag{7.3}$$

Figure 7.13 shows that as the intermediate mass factor increases and the size of the intermediate mass decreases, the percentage error, equation (7.2), decreases significantly.

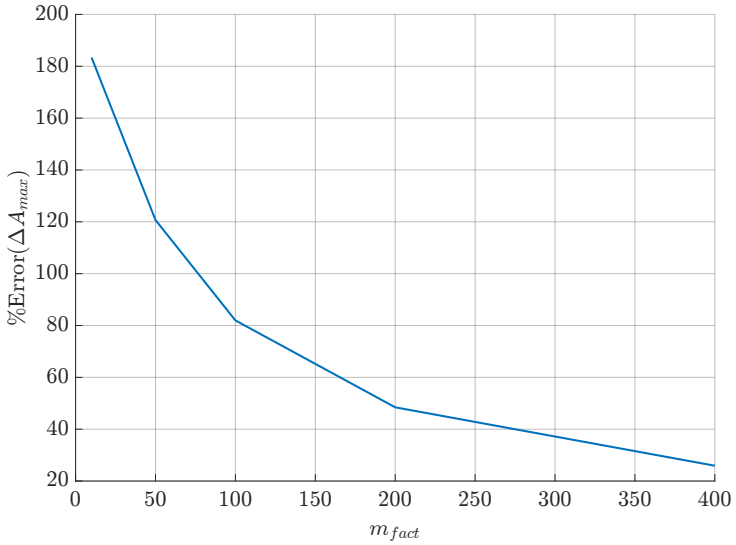


Figure 7.13: Percentage error (deviation between the forces acting on the spring and dashpot) vs intermediate mass factor graph for a multi-element, free bar, Standard Linear model given an impulse force for an intermediate mass investigation.

For a decreasing intermediate mass size and increasing intermediate mass factor, m_{fact}, the relative percentage was calculated as the maximum amplitude of the force acting on the intermediate mass as a percentage of the maximum amplitude of the force acting on the spring

$$Relative\ \% = \left(\frac{max\left(A_{m_{int}}\right)}{max\left(A_{MS}\right)}\right) \times 100 \tag{7.4}$$

where $A_{m_{int}}$ is the amplitude of the force acting on the intermediate mass and A_{MS} is the amplitude of the force acting on the Maxwell spring.

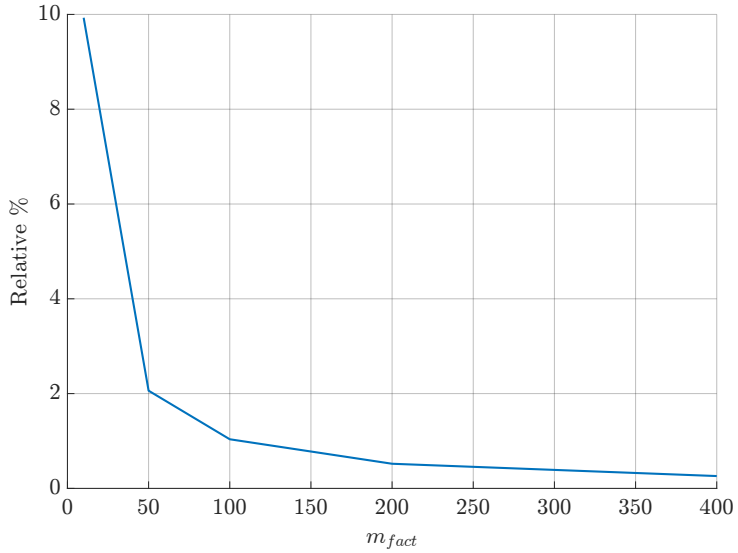


Figure 7.14: Relative percentage (percentage of the force acting on the intermediate mass) vs intermediate mass factor graph for a multi-element, free bar, Standard Linear model given an impulse force for an intermediate mass investigation.

In Figure 7.14, as the intermediate mass factor increases the relative percentage decreases. This shows that the force acting on the intermediate mass decreases with respect to the overall response of the element the smaller this mass becomes.

Further investigation of the effect of the size of the intermediate mass was carried out using the discrete hereditary integral solution as a benchmark. This was done using the same multi-element Standard Linear model as before. Varying sizes of the intermediate mass were investigated for a free bar test where the same impulse force was applied to the end of the set of one hundred elements.

The force acting on the spring and dashpot for the intermediate mass implementation for an intermediate mass factor of 10 is shown in Figure 7.15. This compared to the hereditary integral shows a significant difference in the phase angle of the oscillating response.

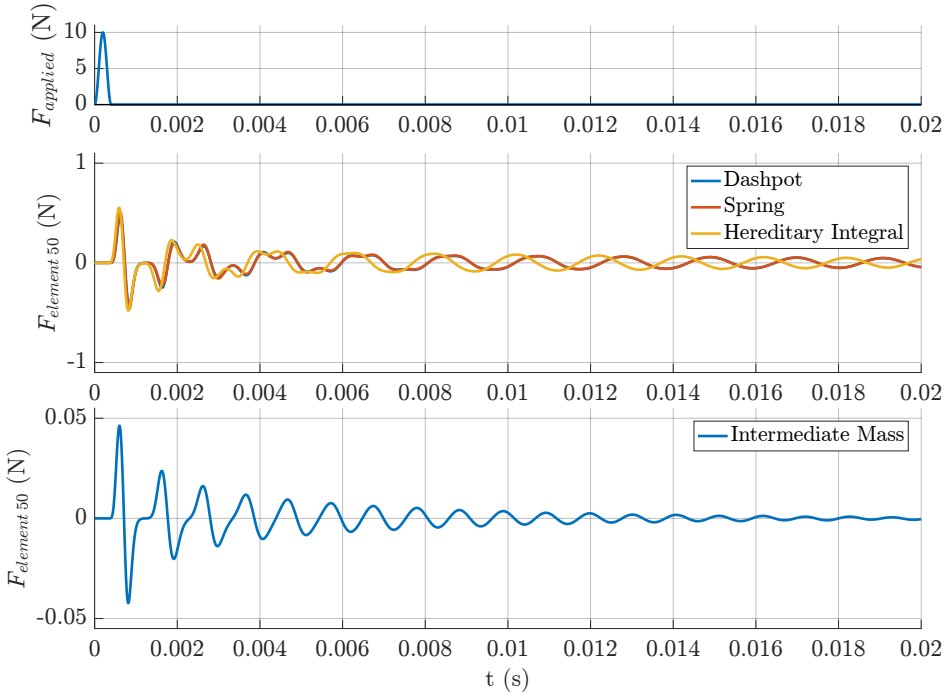


Figure 7.15: Intermediate mass implementation and hereditary integral force comparison graphs for a multi-element, free bar, Standard Linear model given an impulse force for an intermediate mass investigation.

For a decreasing intermediate mass size and increasing intermediate mass factor, m_{fact}, the phase difference between the force acting on the spring for the intermediate mass implementation and the force of the hereditary integral was calculated as

$$\Delta\phi = \phi_{MS} - \phi_{HI} \tag{7.5}$$

where ϕ_{MS} is the phase angle of the most significant frequency of the force acting on the Maxwell spring for the intermediate mass implementation, and ϕ_{HI} is the phase angle at the same frequency as the Maxwell spring, which is also the most significant frequency of the force of the hereditary integral.

Shown in Figure 7.16, the phase difference decreases significantly with an increase in the intermediate mass factor. This shows that the response of the intermediate mass implementation converges to that of the hereditary integral as the size of the intermediate mass decreases.

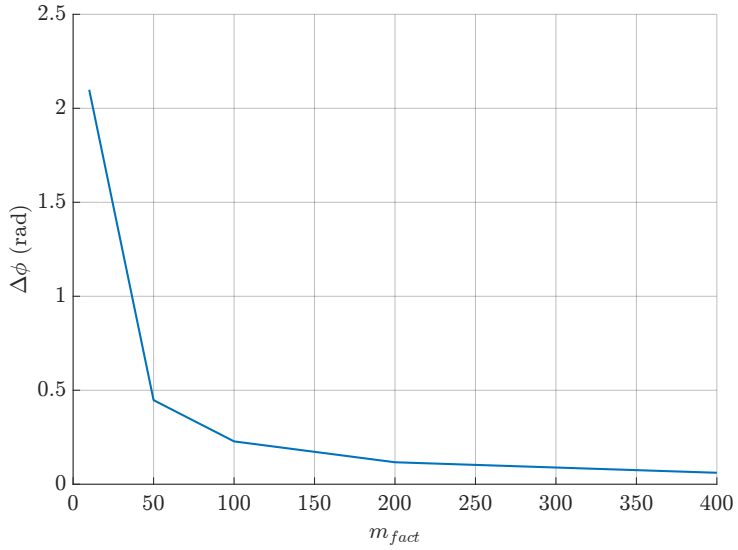


Figure 7.16: Phase difference (between spring and hereditary integral solution) vs intermediate mass factor graph for a multi-element, free bar, Standard Linear model given an impulse force for an intermediate mass investigation.

7.2 Maxwell Element (with Non-Linear Dashpot)

For a Maxwell element with a non-linear dashpot, the two implementations that were investigated were the velocity raised to a power and a dashpot coefficient function for varying values of a power $p = 0.5$, 0.7, 1, 1.2 and 1.5. The dashpot coefficient function was investigated using both the velocity over the dashpot and the velocity over the element for the intermediate mass implementation and the velocity over the element for the hereditary integral.

7.2.1 Velocity Raised to a Power

The approach using the velocity raised to a power was first investigated. This involved using equation (6.33) to calculate the displacements for the intermediate mass implementation.

Intermediate Mass Implementation

The intermediate mass implementation was tested using a single non-linear Maxwell element that was fixed at one end and the other end was given a prescribed displacement of 0.05 m. The Maxwell element used in each test had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a dashpot with viscous coefficient $\eta = 21.2057\,\mathrm{N\,s\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$ and an intermediate mass size $m_{int} = 7.854 \times 10^{-5}\,\mathrm{kg}$.

Dashpot-Spring

For the Dashpot-Spring arrangement, see Figure 7.17, node 1 was displaced by $u_1(t)$, node 3 was fixed and the displacement of node 2 was tracked over time for varying values of power p. The forces acting on the spring and dashpot were also plotted.

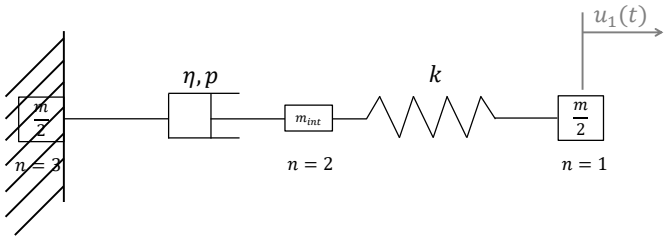


Figure 7.17: Non-linear Maxwell element in a Dashpot-Spring orientation with fixed boundary conditions and prescribed displacement, using the velocity raised to a power p.

In Figure 7.18, the displacement of node 2 and the force acting on the spring and dashpot were plotted for varying values of p. For both the displacement and the force response, as the value of p increases away from linear (p greater than 1), the gradient of the response is smaller (that is the element relaxes at a slower rate). As the value of p decreases away from linear (p less than 1), the gradient of the response is greater, meaning that the element relaxes at a faster rate.

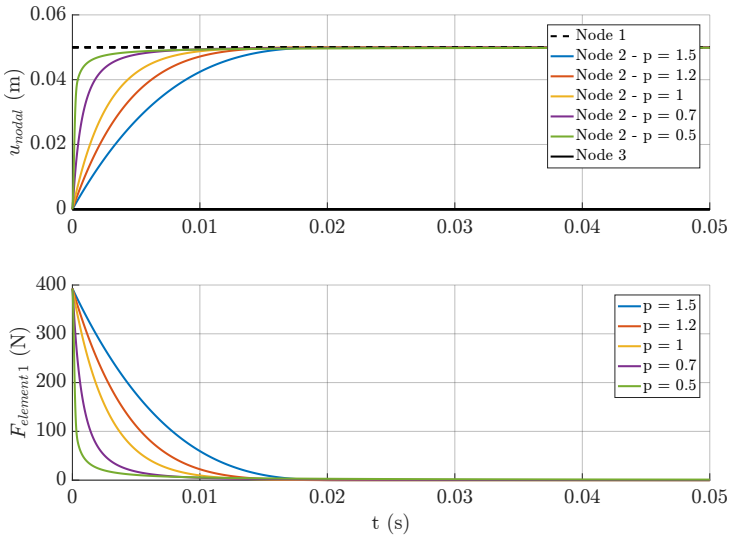


Figure 7.18: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Dashpot-Spring arrangement given a prescribed displacement of 0.05 m, using the velocity raised to a power.

Spring-Dashpot

For the Spring-Dashpot arrangement, see Figure 7.19, again the internal node and the force response of the element are plotted for varying values of p.

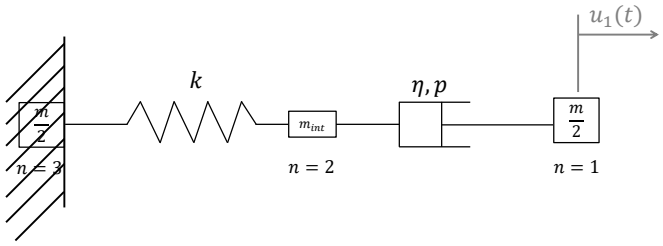


Figure 7.19: Non-linear Maxwell element in a Spring-Dashpot orientation with fixed boundary conditions and prescribed displacement, using the velocity raised to a power p.

Firstly the test was carried out using consistent initial conditions, Figure 7.20. As was seen with the linear Maxwell, the use of consistent initial conditions for the Spring-Dashpot arrangement returns no response. The internal node displacement and force response remain at zero.

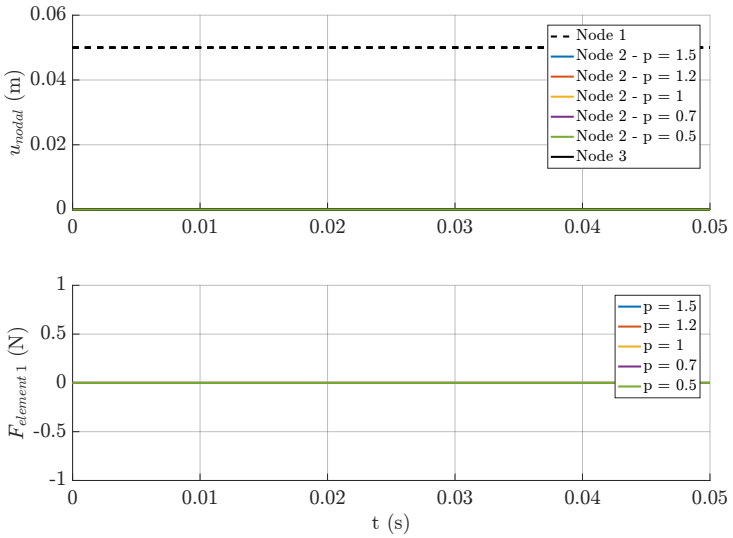


Figure 7.20: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using the velocity raised to a power and using consistent initial conditions.

The same test was carried out using inconsistent initial conditions, as this was shown to give the desired results with the linear Maxwell. Figure 7.21 shows that for a linear Maxwell (p equal to 1), the response of the element is as expected. The internal node displaces to the same amount as node 1, and relaxes over time. For values of p greater than 1, the internal node displacement exceeds that of node 1 which is unstable behaviour, and for a value of p equal to 1.5 the simulation returned invalid results (not shown). For values of p less than 1, as the value of p decreased, the initial displacement and force response of the element decreased.

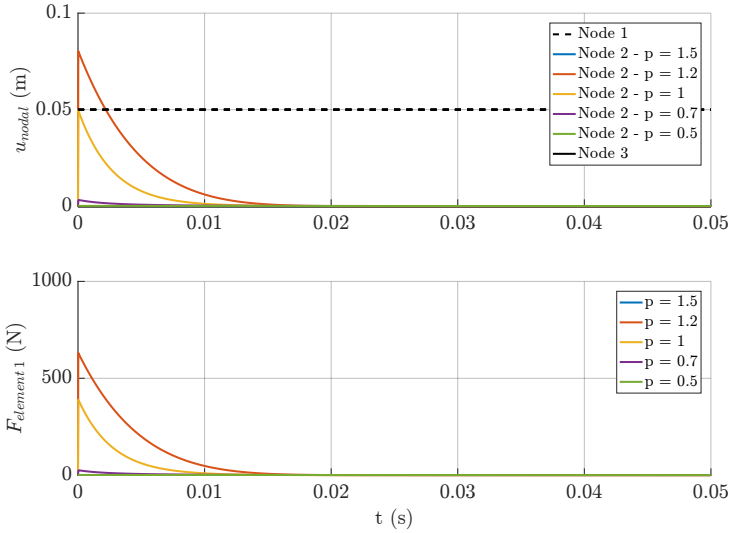


Figure 7.21: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using the velocity raised to a power and using inconsistent initial conditions.

For the Spring-Dashpot arrangement, this test was then carried out with a single Maxwell element using a fictitious element attached to the dashpot, see Figure 7.22.

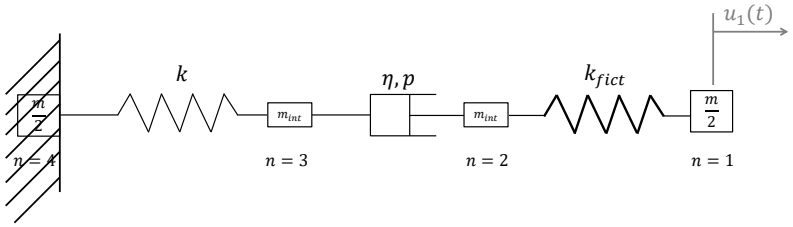


Figure 7.22: Non-linear Maxwell element in a Spring-Dashpot orientation with a fictitious element, fixed boundary conditions and prescribed displacement, using the velocity raised to a power p.

Figure 7.23 shows that again for the linear solution (p equal to 1), the element behaves as expected. For values of p greater than 1, the results become unstable, especially for p equal to 1.5. For values of p less than 1, there is still a diminished response of the element to the prescribed displacement.

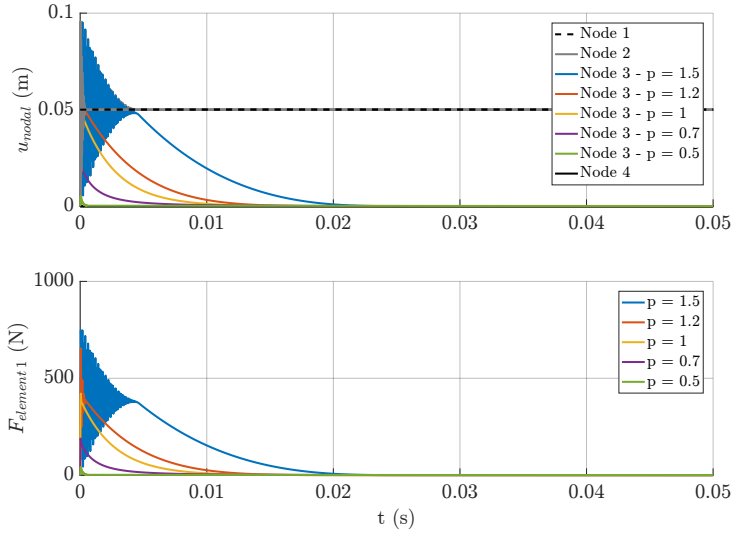


Figure 7.23: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using the velocity raised to a power and using a fictitious element.

7.2.2 Dashpot Coefficient Function - Velocity Over the Dashpot

The other implementation of a non-linear Maxwell dashpot is the dashpot coefficient function which can use either the velocity over the dashpot or the velocity over the element. The intermediate mass implementation of a non-linear Maxwell element was outlined in Chapter 6.3.3. First the velocity over the dashpot was investigated for the intermediate mass implementation.

Intermediate Mass Implementation

The intermediate mass implementation was investigated using a single non-linear Maxwell element that was fixed at one end and the other end was given a prescribed displacement of 0.05 m, as in previous tests. The Maxwell element used in each test had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$, an intermediate mass size $m_{int} = 2.618 \times 10^{-5}\,\mathrm{kg}$ and a dashpot with viscous coefficient as a function defined as

$$\eta\left(\dot{u}(t), p\right) = \bar{\eta}\,\dot{u}(t)^{p-1} \tag{7.6}$$

where $\bar{\eta} = 21.2057\,\mathrm{N\,s\,m^{-1}}$.

Dashpot-Spring

A single Maxwell element in the Dashpot-Spring orientation shown in Figure 7.24 had node 3 fixed and node 1 was given a prescribed displacement. The dashpot coefficient function was calculated using the velocity over the dashpot which is $\dot{u}_2 - \dot{u}_3$.

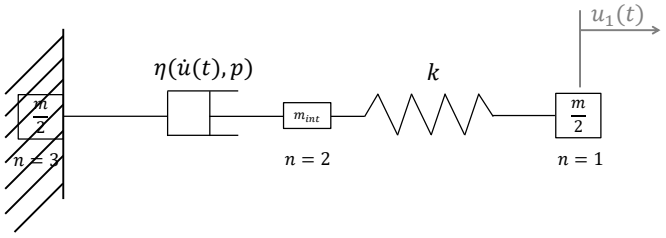


Figure 7.24: Non-linear Maxwell element in a Dashpot-Spring orientation with fixed boundary conditions and prescribed displacement, using a dashpot coefficient function of $\eta\left(\dot{u}_{dashpot}(t), p\right)$.

The Dashpot-Spring arrangement was tested using consistent initial conditions and the internal node was tracked. Figure 7.31 shows the displacement and force response of the non-linear Maxwell element for varying values of p. This shows that as the value of p increases above linear (p greater than 1), the rate of relaxation decreases in both the displacement and force graphs. As the value of p decreases below linear (p less than 1), the rate of relaxation increases.

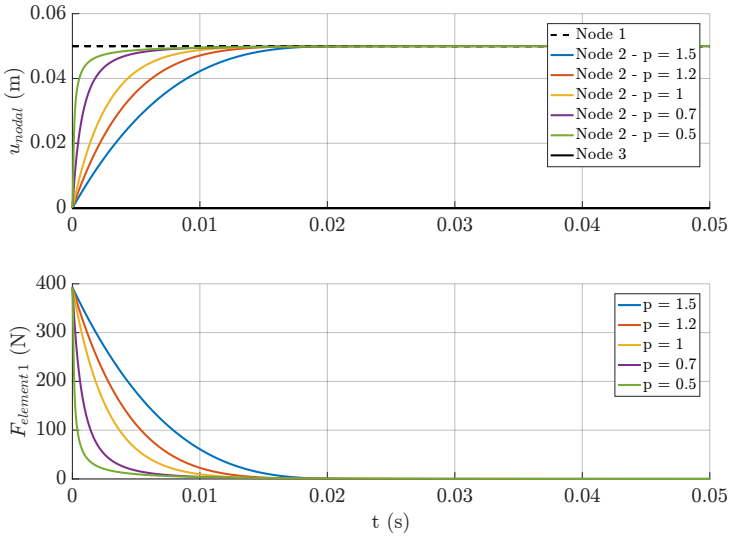


Figure 7.25: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Dashpot-Spring arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function as a function of the velocity over the dashpot.

Spring-Dashpot

The Spring-Dashpot arrangement, shown in Figure 7.26, was then tested using the same prescribed displacement test.

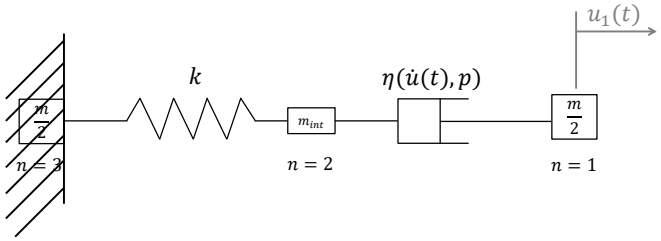


Figure 7.26: Non-linear Maxwell element in a Spring-Dashpot orientation with fixed boundary conditions and prescribed displacement, using a dashpot coefficient function of $\eta(\dot{u}(t), p)$.

The simulation was first carried out using consistent initial conditions. Figure 7.32 shows that, as expected, there is no response for this arrangement.

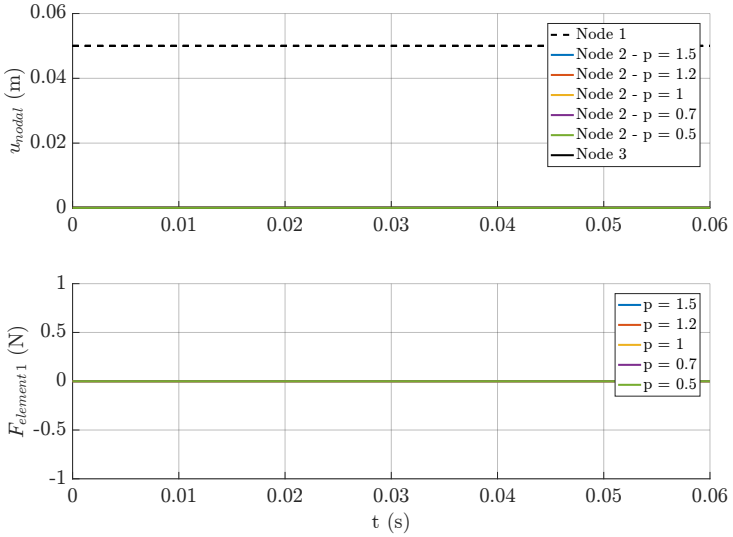


Figure 7.27: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using consistent initial conditions.

The same arrangement was then tested using inconsistent initial conditions. Figure 7.33 shows that the displacement and force response are exhibiting the same behaviour as the velocity raised to a power. For increasing values of p greater than 1, the initial force and displacement exceed that of the linear simulation with a value of $p = 1.5$ giving invalid results (not shown). As the value of p decreases below a value of 1, the initial displacement and force of the element decrease which is not the desired result as this is expected to align with the Dashpot-Spring element results.

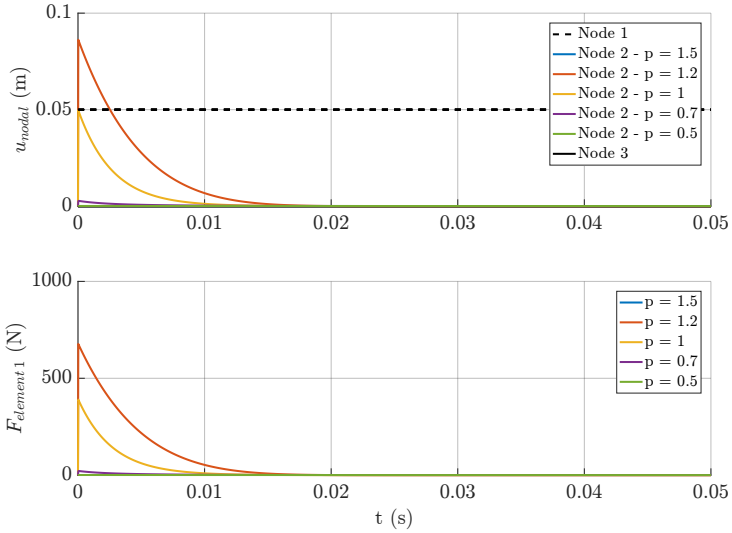


Figure 7.28: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using inconsistent initial conditions.

A fictitious element was then attached to the dashpot, see Figure 7.29, and the same prescribed displacement test was carried out. The internal nodes were tracked, the nodes on either side of the fictitious element displaced to the same amount, see Figure 7.30, and the node between the spring and dashpot was tracked for varying values of p. As was seen before with the velocity raised to a power, the displacement and force results become unstable with an increasing value of p above linear (p greater than 1) and the displacement and force results diminish as p decreases below a value of 1.

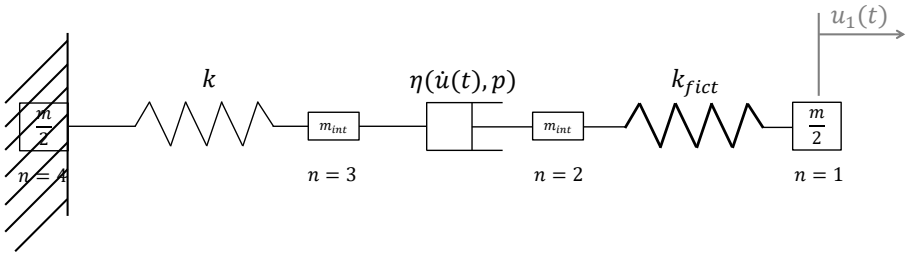


Figure 7.29: Non-linear Maxwell element in a Dashpot-Spring orientation with a fictitious element, fixed boundary conditions and prescribed displacement, using a dashpot coefficient function of $\eta\left(\dot{u}(t), p\right)$.

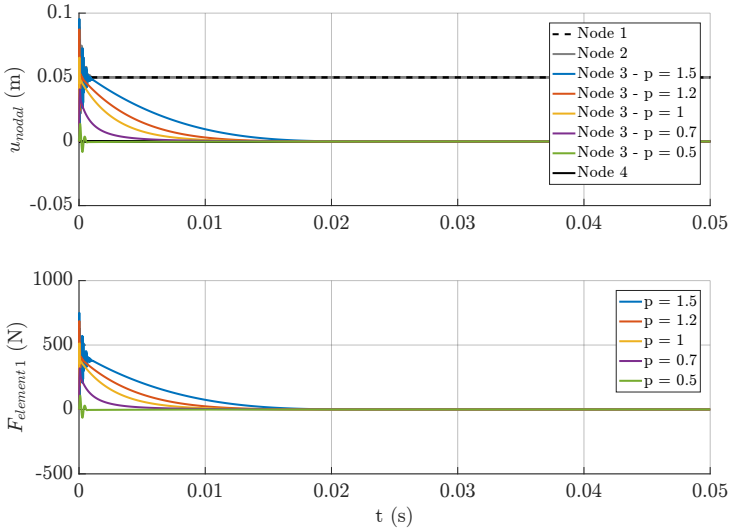


Figure 7.30: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using a fictitious element.

7.2.3 Dashpot Coefficient Function - Velocity Over the Element

The other implementation of the dashpot coefficient function is to use the velocity over the element. The intermediate mass implementation was outlined in Chapter 6.3.3, and the hereditary integral solution uses equation (6.36). First a stress-relaxation test was carried out for both the intermediate mass implementation and hereditary integral where a single element was given a prescribed displacement and the velocity over the element is zero. This means that the value of the dashpot coefficient is constant throughout the simulation. In order to investigate the dashpot coefficient function in the hereditary integral, constant velocity tests were carried out for different velocities and values of p.

Intermediate Mass Implementation

The intermediate mass implementation was again investigated using a single non-linear Maxwell element that was fixed at one end and the other end was given a prescribed displacement of 0.05 m as in previous tests. The Maxwell element used in each test had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$ and an intermediate mass size $m_{int} = 2.618 \times 10^{-5}\,\mathrm{kg}$. The dashpot coefficient function used the velocity over the element which for a prescribed displacement test is zero. This in turn means that the value of the dashpot coefficient is zero for values of $p > 1$ and tends towards infinity for value of $p < 1$. This would cause numerical instability and therefore a small value $\dot{u}_\epsilon$ was introduced such that the dashpot coefficient was very small for values of $p > 1$ and very large for values of $p < 1$ instead of zero and infinity respectively. The

dashpot coefficient function was defined as

$$\eta\left(\dot{u}(t), p\right) = \bar{\eta}\left(\dot{u}(t) + \dot{u}_{\epsilon}\right)^{p-1} \tag{7.7}$$

where $\bar{\eta} = 21.2057\,\mathrm{N\,s\,m^{-1}}$.

Dashpot-Spring

The Dashpot-Spring arrangement was tested using consistent initial conditions and the internal node was tracked. Figure 7.31 shows the displacement and force response of the non-linear Maxwell element for varying values of p. This shows that as the value of p increases above 1 (an increasing viscous coefficient), the rate of relaxation increases in both the displacement and force graphs. As the value of p decreases below 1 (a decreasing dashpot coefficient), the rate of relaxation decreases. This is the opposite behaviour to what was seen when using the velocity over the dashpot.

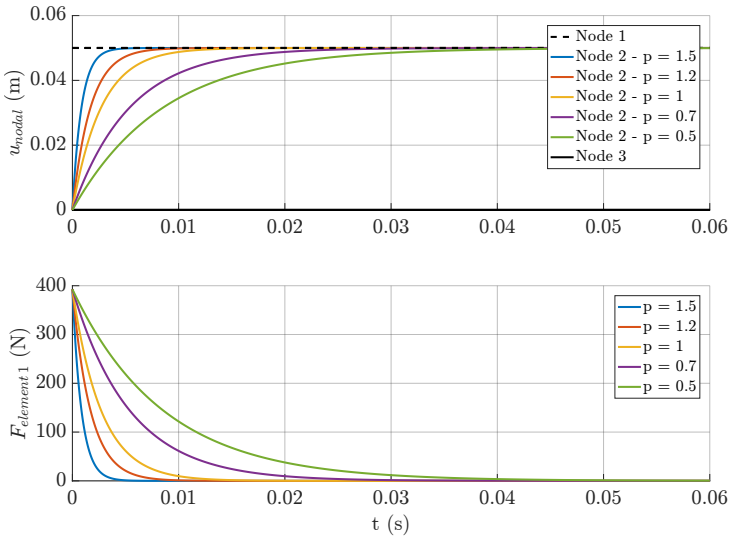


Figure 7.31: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Dashpot-Spring arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function.

Spring-Dashpot

The Spring-Dashpot arrangement was then tested using the same prescribed displacement test and using consistent initial conditions. Figure 7.32 shows that, as expected, there is no response for this arrangement.

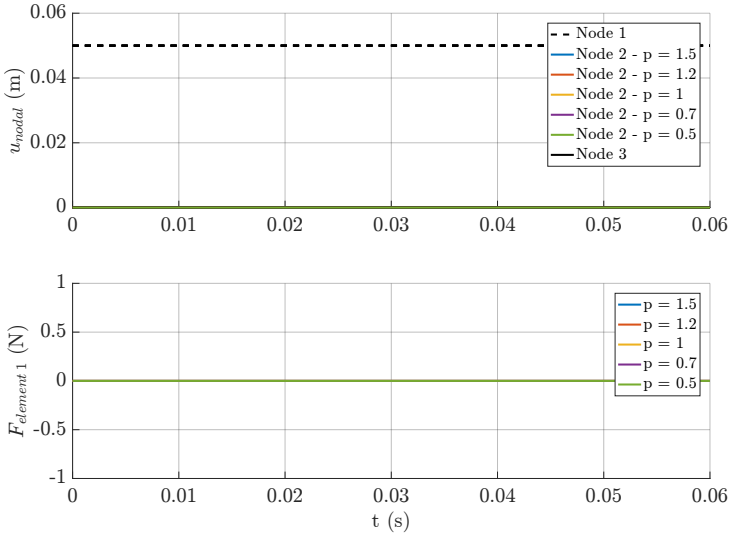


Figure 7.32: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using consistent initial conditions.

The arrangement was then tested using inconsistent initial conditions. Figure 7.33 shows that the displacement and force response are exhibiting expected behaviour. For increasing values of p greater than 1, the rate of relaxation of the element increases. As the value of p decreases below a value of 1, the rate of relaxation of the element decreases. This is the same behaviour as was seen for the Dashpot-Spring arrangement, however, the displacement of the internal node relaxes in the opposite direction as expected.

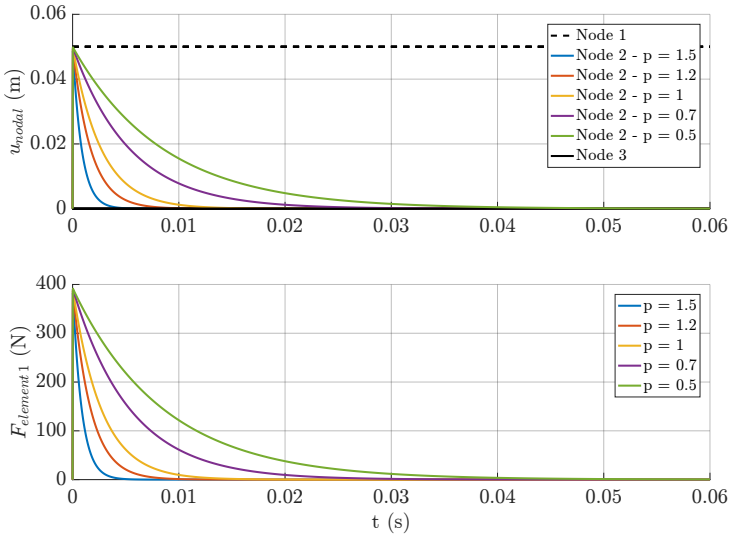


Figure 7.33: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using inconsistent initial conditions.

A fictitious element was then attached to the dashpot, see Figure 7.29, and the same prescribed

displacement test was carried out. The nodes on either side of the fictitious element were displaced to the same amount, see Figure 7.34, and the internal nodes were tracked, as well as the node between the spring and dashpot, for varying values of p. As was seen before, the rate of relaxation increases with an increasing value of p above 1 and the rate of relaxation decreases as the value of p decreases below 1.

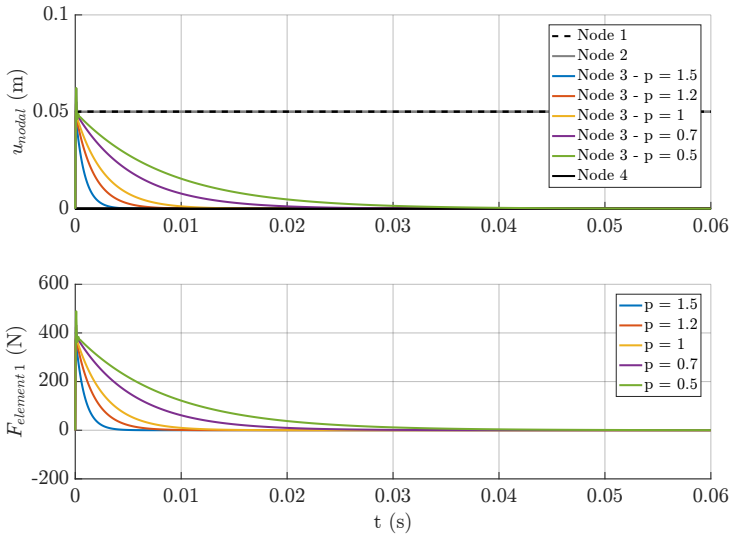


Figure 7.34: Displacement-time and force-time comparison graphs for varying values of p for a non-linear single Maxwell element in a Spring-Dashpot arrangement given a prescribed displacement of 0.05 m, using a dashpot coefficient function and using a fictitious element.

Discrete Hereditary Integral Model

The discrete hereditary integral solution was tested for a single non-linear Maxwell element, see Figure 7.35. The same prescribed displacement test was performed using the same system variables as previous tests. The hereditary integral solution uses equation (6.36).

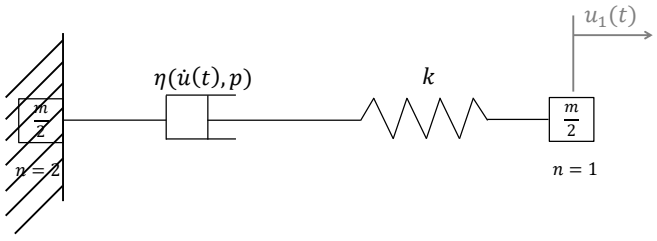


Figure 7.35: Non-linear Maxwell element modelled by a hereditary integral with fixed boundary conditions and prescribed displacement, using a dashpot coefficient function of $\eta\left(\dot{u}(t), p\right)$.

Only the force response was plotted since there is no internal node to track. In Figure 7.36, the force response of the element was plotted for varying values of the power, p. This shows the same behaviour as was exhibited by the intermediate mass implementation using a dashpot coefficient

function as a function of the velocity over the element, see Figure 7.37. For values of p greater than 1, as the value of p increases the rate of relaxation increases. For values of p less than 1, as the value of p decreases so does the rate of relaxation of the element.

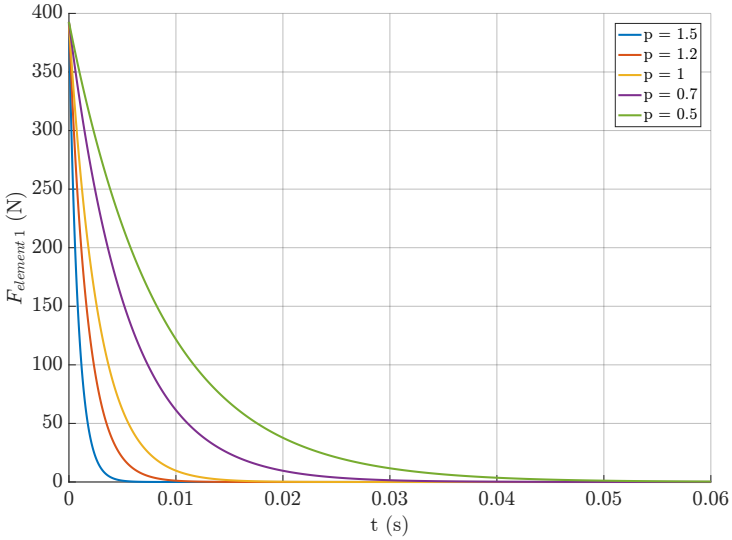


Figure 7.36: Force-time comparison graphs for varying values of p for a non-linear single Maxwell element using the hereditary integral solution given a prescribed displacement of 0.05 m, using a dashpot coefficient function.

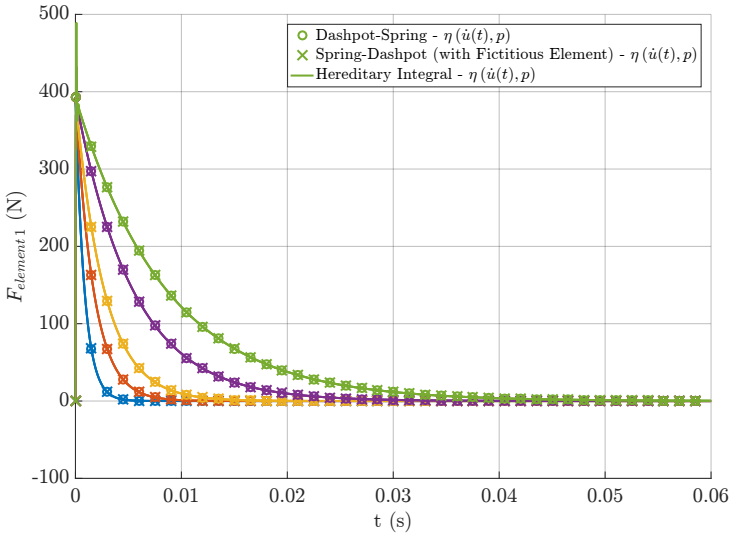


Figure 7.37: Force-time comparison graphs for varying values of p for a non-linear single Maxwell element using the hereditary integral solution compared to the intermediate mass implementation results, given a prescribed displacement of 0.05 m and using a dashpot coefficient function.

Incremental Time-stepping Scheme

The incremental time-stepping scheme was tested using the same prescribed displacement test as before. The test was run on an unconstrained element, and the force response was plotted for varying values of p. Figure 7.38 shows the same behaviour that was seen in the discrete hereditary

integral results above. The maximum force response is the same as previous tests using this prescribed displacement, just under 400 N.

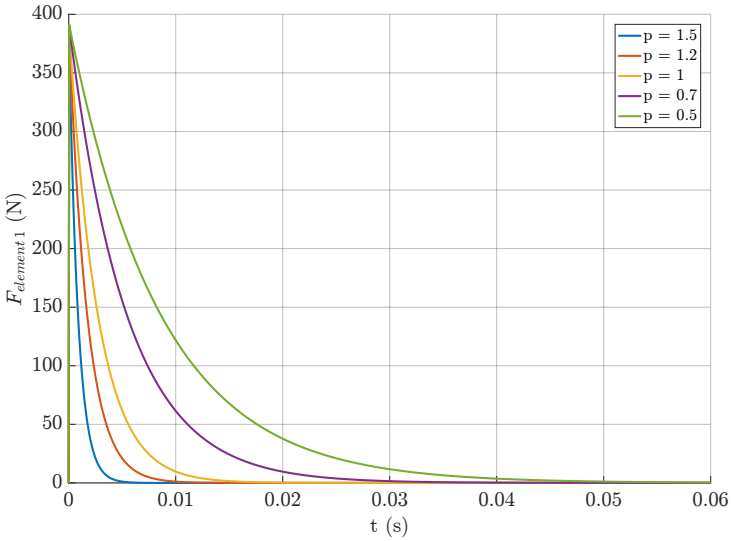


Figure 7.38: Displacement-time and force-time comparison graphs for varying values of p for an unconstrained single non-linear Maxwell element using the hereditary integral solution given a prescribed displacement of 0.05 m, using a dashpot coefficient function and the incremental time-stepping scheme.

Hereditary Integral and Intermediate Mass Implementation Comparison

In order to investigate the use of the dashpot coefficient in the hereditary integral using the velocity over the element, a single element simulation was carried out and compared to the intermediate mass implementation. The Maxwell element had a spring with stiffness $k = 7853.9816\,\mathrm{N\,m^{-1}}$, a mass of $m = 0.007854\,\mathrm{kg}$, an intermediate mass size $m_{int} = 2.618 \times 10^{-5}\,\mathrm{kg}$ and a dashpot with viscous coefficient

$$\eta\left(\dot{u}(t), p\right) = \bar{\eta}\,\dot{u}(t)^{p-1} \tag{7.8}$$

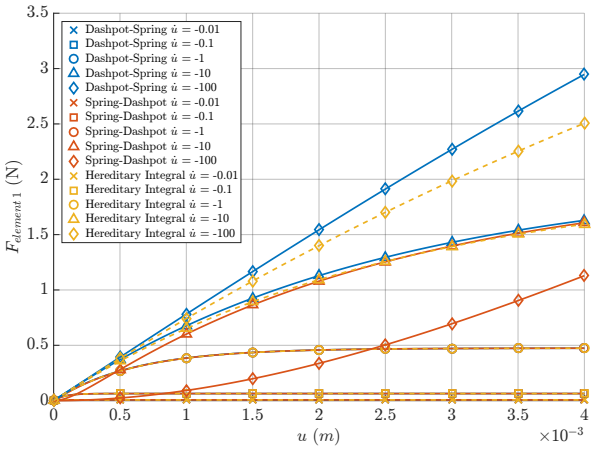
where $\bar{\eta} = 2.12057\,\mathrm{N\,s\,m^{-1}}$.

A constant velocity was applied to one end of the element with the other end fixed. The simulation was carried out for varying velocities and varying values of p. The hereditary integral element, shown in Figure 7.35, was compared to the two intermediate mass implementation elements. The two orientations that were investigated using this simulation were the Dashpot-Spring, shown in Figure 7.24, and the Spring-Dashpot element, shown in Figure 7.26.

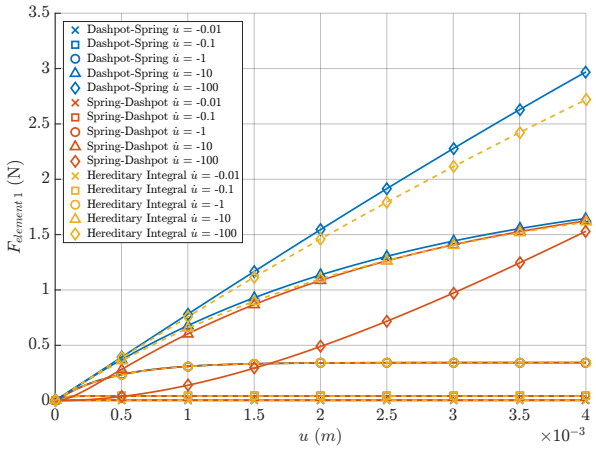
For each velocity, the force acting on the element was plotted over the displacement of the element and this graph was plotted for each value of p, see Figure 7.39. Each simulation was run until the element achieved the same stretch of $4 \times 10^{-3}\,\mathrm{m}$.

For all values of p, and all velocities except for a velocity of $-100\,\mathrm{m\,s^{-1}}$, the hereditary integral,

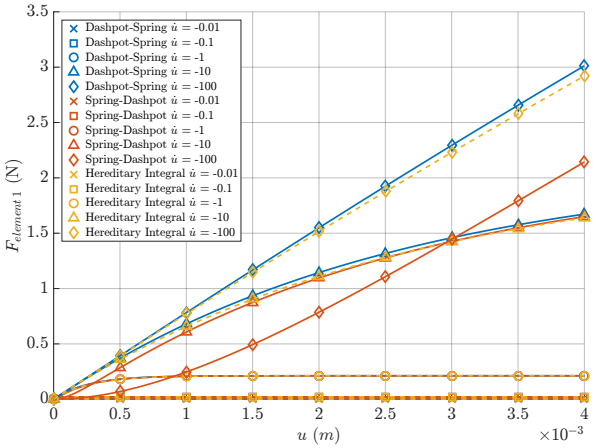
Dashpot-Spring and Spring-Dashpot results track closely. For a velocity of $-100\,\mathrm{m\,s^{-1}}$ and a value of $p = 1.5$ the hereditary integral and the Dashpot-Spring results are aligned, with a large discrepancy with the Spring-Dashpot. As the value of p decreases, the discrepancy grows and a difference between the hereditary integral and the Dashpot-Spring results appears.



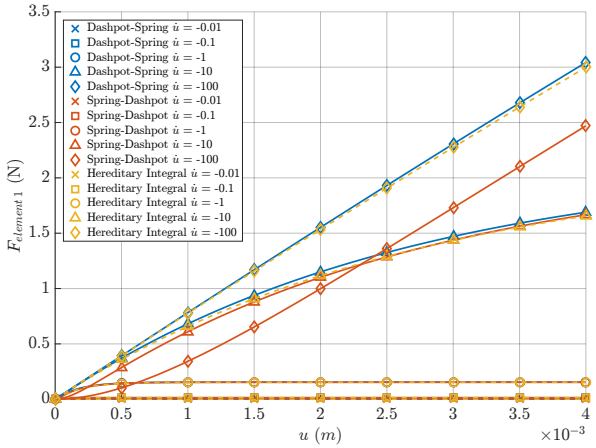
(a) Force-Displacement - $p = 0.5$.



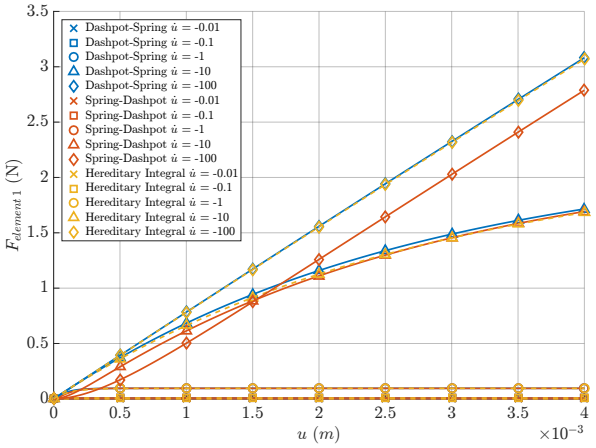
(b) Force-Displacement - $p = 0.7$.



(c) Force-Displacement - $p = 1$.



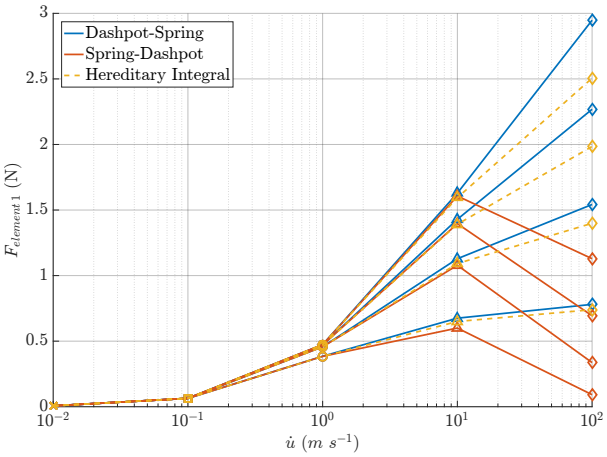
(d) Force-Displacement - $p = 1.2$.



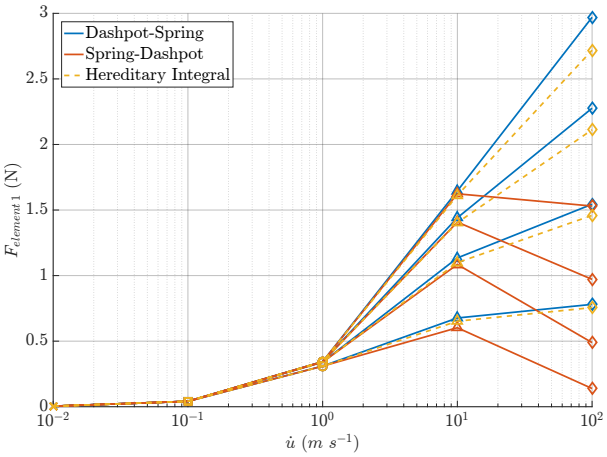
(e) Force-Displacement - $p = 1.5$.

Figure 7.39: Single Element Maxwell - $\eta(\dot{u}_{el})$ - Force-Displacement graphs for constant velocity tests.

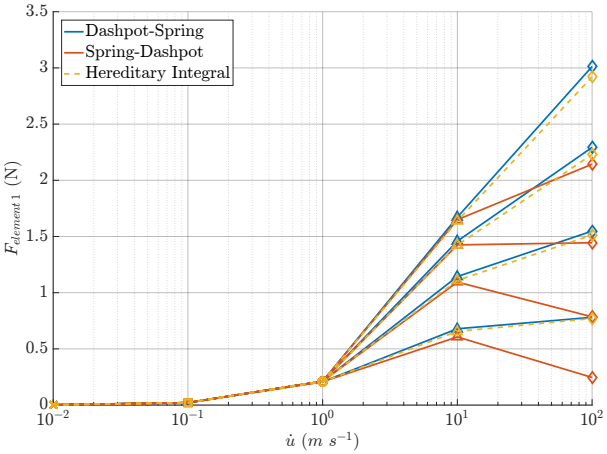
For selected displacements, the force for each velocity was sampled and plotted. This was done for each value of p and shown in Figure 7.40. This shows the same trends as described above, with an increase in the gradient of the response between velocities of $1\,\mathrm{m\,s^{-1}}$ and $100\,\mathrm{m\,s^{-1}}$ as the value of p increases.



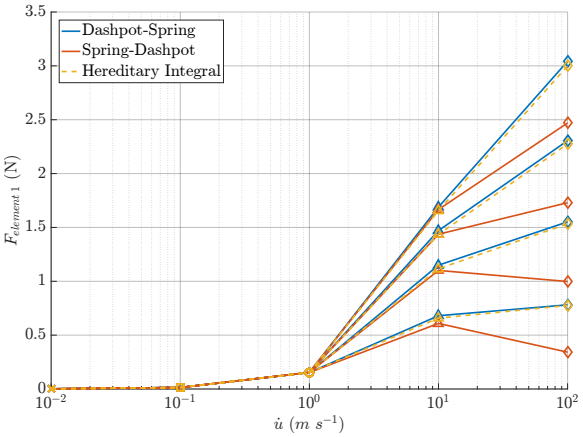
(a) Force-Velocity - $p = 0.5$.



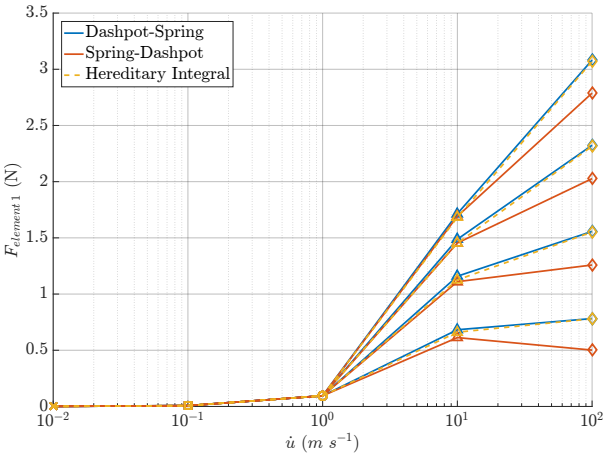
(b) Force-Velocity - $p = 0.7$.



(c) Force-Velocity - $p = 1$.



(d) Force-Velocity - $p = 1.2$.



(e) Force-Velocity - $p = 1.5$.

Figure 7.40: Single Element Maxwell - $\eta(\dot{u}_{el})$ - Force-Velocity curves for constant velocity tests.

Chapter 8

Discussion of One-Dimensional Numerical Model Results

This chapter discusses the results presented in Chapter 7. The linear Maxwell results for the different orientations and initial conditions are discussed, as well as the investigation into the effect of the size of the intermediate mass. Then the non-linear Maxwell results are examined for the two different implementations of a non-linear dashpot. The use of the strain rate over the dashpot and over the element is also discussed in the dashpot coefficient function implementation.

8.1 Linear Maxwell Model

The linear Maxwell model was investigated using the intermediate mass implementation and the hereditary integral. The effect of the size of the intermediate mass was also investigated.

8.1.1 Intermediate Mass Implementation Verification

The intermediate mass implementation was built on the central difference method. This was modified, as discussed in Chapter 6.1, in order to allow for the expansion to a multi-element model, and was first investigated using a linear Maxwell element.

A prescribed displacement was applied to a single Maxwell element to test that the relaxation behaviour was as expected for the different orientations of the element. The same test was carried out for both possible orientations of the Maxwell element. For the Dashpot-Spring arrangement, the expected behaviour outlined by the theory in Chapter 3 was presented in Figure 6.3a and this

matches the actual behaviour that was seen in Figure 7.2. The force of this element in Figure 7.3 also shows the relaxation behaviour of the element that we would expect. Likewise, the force response of the spring and of the dashpot are also equal which is aligned with the behaviour discussed in the theory in Chapter 3. Figure 7.3 also shows the force acting on the internal mass is negligible compared to the force acting on the spring and dashpot. Therefore, the Spring-Dashpot arrangement of the linear Maxwell element using the intermediate mass implementation gives the exact results that were expected.

The Spring-Dashpot arrangement was then investigated and was expected to output the same results as the Dashpot-Spring arrangement. However, when the same test was applied to the Spring-Dashpot orientated element, no response to the prescribed displacement was shown, as seen in the displacement-time graph in Figure 7.5 and the force acting on the spring and dashpot in Figure 7.6 was zero. This confirms the theory set out in Chapter 6.2.1 where due to the starting conditions of the simulation, the displacement of the node at time step -1 is the same as the displacement at time step 0 and therefore the boundary conditions are not initialised as they should be. Inconsistent initial conditions fixed this issue and the behaviour of the Spring-Dashpot arrangement, shown in Figure 7.7, now matches that of the expected behaviour presented by the theory in Figure 6.3b. The force acting on the spring and dashpot are equal and representative of the relaxation behaviour that is expected, as seen in Figure 7.8. The force acting on the internal mass in Figure 7.8 is also negligible compared to that of the spring and dashpot.

The Spring-Dashpot arrangement therefore only produces the expected results by applying inconsistent initial conditions, where the displacement at time step -1 is set to zero, which is a large assumption to make about the actual behaviour of the element. The implementation of the inconsistent initial conditions reduces the accuracy of the results. For a linear Maxwell element, however, this solves the issue temporarily.

8.1.2 Discrete Hereditary Integral Solution

The hereditary integral was then compared to the intermediate mass implementation discussed above. As there is no intermediate mass to track in the hereditary integral solution, only the force was compared. Figure 7.10 shows that the force of the hereditary integral solution is what is expected for each arrangement of the Maxwell element. Due to there being no stiffness or damping matrix in the implementation of the hereditary integral, the orientation of the spring and dashpot has no effect on the results which show that the force response is as expected regardless of initial conditions.

8.1.3 Investigation into the Effect of the Size of the Intermediate Mass

Investigation of the size of the intermediate mass was carried out for the Maxwell element. Only the size of the mass was adjusted between tests. The test that was carried out was the application of an impulse force to one end of a row of 100 Standard Linear elements with both ends unconstrained. The result that was expected was a stress wave travelling through the bar and decreasing in amplitude due to the dashpot in the Maxwell element. This is the result shown in Figure 7.12.

The force response of the spring and dashpot is expected to be equal and the force response of the internal mass is expected to be negligible compared to the force response of the spring and dashpot. To measure this, the percentage error between the maximum amplitude of the force response of the spring and dashpot was calculated for varying sizes of the intermediate mass. The maximum amplitude of the force response of the internal mass as a percentage of the maximum amplitude of the force response of the spring was also calculated for varying sizes of internal masses. Both Figure 7.13 and 7.14 show that the difference between the force response of the spring and dashpot decreases, and the difference between the force response of the internal mass as compared to the spring and dashpot, decreases as the size of the intermediate mass decreases.

The hereditary integral developed was then used as a reference to gauge the effect of the size of the internal mass as this does not affect the result of the hereditary integral. The same configuration was used and the force of the hereditary integral was compared to the force acting on the spring and dashpot for varying sizes of the intermediate mass. Figure 7.15 shows that the intermediate mass implementation lags the hereditary integral solution for a relatively large intermediate mass. The difference between the force acting on the spring and dashpot for the intermediate mass implementation and the hereditary integral were measured using the phase difference. The effect of decreasing the intermediate mass is shown in Figure 7.16. As the size of the intermediate mass decreases the force response of the discrete Maxwell tended towards the hereditary integral result. The choice of too small a size for the intermediate mass, however, will lead to divide by zero errors.

8.2 Non-Linear Maxwell Model

The non-linear Maxwell element was implemented in two different ways. The first implementation that was considered was raising the nodal velocity in the intermediate mass implementation to a power. Then the dashpot coefficient was considered as a function of both the velocity over the dashpot and the velocity over the element.

8.2.1 Velocity Raised to a Power

A prescribed displacement test similar to that used for the linear Maxwell was used to investigate raising the nodal velocity to a power. First, the relaxation behaviour of the Dashpot-Spring configuration was examined for varying values of the power p, which is presented in Figure 7.18. Varying degrees of relaxation can be seen, with the linear result for $p = 1$ being replicated from the previous linear Maxwell results. As the value of p increases above 1 the force required to move the dashpot at the same rate increases. The expected effect on the relaxation behaviour would be an increase in the amount of time for the element to return to an equilibrium state. This is what is seen in Figure 7.18 with the displacement of the intermediate node taking longer to reach that of the originally displaced node for $p = 2.5$ than for $p = 1$. Conversely, as p decreases below 1, we would expect the element to take less time to return to a state of equilibrium, since the force needed to move the dashpot at the same rate decreases. This is also confirmed by Figure 7.18 where the displacement of the intermediate node for $p = 0.5$ takes less time to reach that of the originally displaced node than for $p = 1$. Therefore, the Spring-Dashpot arrangement exhibits the behaviour that is expected.

The relaxation behaviour of the Spring-Dashpot arrangement was then investigated using consistent initial conditions where the displacement and force graphs, shown in Figure 7.20, showed no response, the same as the linear model due to the initial conditions not being sufficiently initialised. With the linear model, the use of inconsistent initial conditions fixed this issue, however, as seen in Figure 7.21, this was not the case for the non-linear model. For values of p less than 1, as the value of p gets smaller, the initial force applied to the intermediate node by the inconsistent initial conditions gets smaller. For values of p greater than 1, as the value of p gets larger, the initial force applied to the intermediate node by the inconsistent initial conditions gets greater. This shows that using inconsistent initial conditions for the non-linear model, where the velocity is raised to a power, is not a valid solution. The alternative way to apply an initial force to the intermediate node is by using a fictitious element. This also exhibits the same issues as using inconsistent initial conditions, as seen in Figure 7.23 the force response is diminished for values of p less than 1 and increased for values of p greater than 1. For large values of p, there are also numerical issues and instability. This could be due to the size of the time step that was used and could need to be adjusted when rates are raised to a power.

8.2.2 Dashpot Coefficient Function

The second method for non-linearisation of the Maxwell element is to make the dashpot coefficient a function of velocity. This velocity can have many different implementations however the two that were investigated were the velocity over the dashpot (local velocity) and the velocity over the element (global velocity). First, using the velocity over the dashpot was investigated by applying a prescribed displacement to the intermediate mass implementation models as the internal node

is tracked, and therefore the velocity over the dashpot can be calculated. When a prescribed displacement is applied to the Dashpot-Spring configuration of this non-linear model, the relaxation behaviour that is exhibited is what is expected, as shown in Figure 7.31, and is aligned with the results using the velocity raised to a power implementation. When the same test is applied to the Spring-Dashpot configuration using consistent initial conditions, there is no response due to the initial conditions not being correct, shown in Figure 7.32, much like the previous results. Using inconsistent initial conditions with this solution, shown in Figure 7.33, does not fix the issue as was seen in the previous non-linear model. Using a fictitious element also produces the same results as the velocity raised to a power, Figure 7.34. This proves that using the velocity raised to a power and a dashpot coefficient function, as a function of the velocity over the dashpot, in a stress-relaxation test are equivalent implementations as is expected.

The prescribed displacement tests were then carried out using a dashpot coefficient function as a function of the velocity over the element. This was investigated for the intermediate mass implementation, the hereditary integral and the incremental time-stepping scheme. The velocity over the element for this test is zero throughout the simulation and for a dashpot coefficient function, where the velocity is raised to a power of $p-1$, this means that the dashpot coefficient will have a value of zero for $p > 1$ and tend toward infinity for $p < 1$.

In order to avoid numerical issues, and still investigate the effect of using the velocity over the element, a small value was introduced and added to the velocity before raising it to a power. This means that the dashpot coefficient had a very small value for $p > 1$ instead of zero and a very large value for $p < 1$ instead of tending toward infinity. Since the velocity of the element does not change throughout the simulation, the value of the dashpot coefficient is also a constant that only changes with the power p. For values of $p > 1$ the value of the coefficient gets smaller which means that the element will relax quicker and for values of $p < 1$ the element will relax slower, shown in Figure 7.31. This is the opposite to what was seen when the velocity over the dashpot was used. This is the expected behaviour for the given values of p when the velocity is zero or very small. The results shown for both arrangements of the intermediate mass implementation followed the same trend of behaviour between the different implementations as with the linear dashpot. The Dashpot-Spring arrangement provided the expected trends for varying values of p. The Spring-Dashpot arrangement using consistent initial conditions again showed no response. Using inconsistent initial conditions and a fictitious element for this arrangement also showed expected results.

Discrete Hereditary Integral Solution

The hereditary integral shows the same relaxation behaviour as the intermediate mass implementation, shown in Figure 7.36. This shows that for a constant dashpot coefficient the hereditary integral and intermediate mass implementation are equivalent. The incremental time-stepping scheme was also investigated for the Maxwell element using the hereditary integral. This

was done using the same dashpot coefficient function and shows the expected relaxation behaviour in Figure 7.38.

Hereditary Integral and Intermediate Mass Implementation Comparison

The behaviour exhibited by the dashpot coefficient function using the velocity over the element when the velocity is zero is expected, however, this does not give a meaningful result against which to compare the hereditary integral. In order to investigate the non-linear dashpot coefficient function used in the hereditary integral, a test where the velocity over the element is not zero was required. The test that was carried out was a constant velocity test which was carried out at different velocities and for varying values of p.

Figure 7.39 shows force-displacement graphs and that for all values of p the trend of the gradient of these graphs is similar for all implementations. At very low velocities the modulus is very small, if not zero, which is aligned with the theory of a Maxwell element. At low velocities the dashpot offers very little resistance to the movement of the element and therefore exhibits a very low force. As the velocity increases the modulus increases through the intermediate velocities, with a non-linear curve up until a high velocity where the modulus is high and constant. The transition between low and high velocities is what is mainly affected by the value of p. The graphs were plotted using the Dashpot-Spring and Spring-Dashpot arrangements of the intermediate mass implementation and the hereditary integral. For values of $p \geq 1$ the Dashpot-Spring and hereditary integral showed similar, if not identical, behaviour. As the value of p decreased below 1 the hereditary integral and Dashpot-Spring solutions drift apart. This is likely due to the acceleration of the intermediate mass at high velocities. The Spring-Dashpot results largely follow the Dashpot-Spring and hereditary integral results for intermediate to low velocities for all values of p, however, at high velocities for all values of p the Spring-Dashpot results differ quite significantly.

These trends are shown in a slightly different way in Figure 7.40 where the force at various displacements for the different velocity tests were plotted. This is normally how the effect of velocity on the behaviour of a material is best seen. For the hereditary integral results specifically an increase of the power p increases the gradient of the transition period quite significantly. This will be used in the following chapters to develop and simulate a material model for cortical bone with a focus on capturing the steepness of this transition period.

8.3 Summary

The outcome of the tests is presented in Tables 8.1 and 8.2. Table 8.1 summarises the prescribed-displacement tests for both linear and non-linear Maxwell elements.

Table 8.1: Matrix of expected behaviour for prescribed displacement one-dimensional numerical results.

		Dashpot-Spring	Spring-Dashpot			Hereditary Integral
		Consistent	Consistent	Inconsistent	Fictitious Element	
Linear Maxwell		●	○	●	–	●
Non-Linear Maxwell						
Velocity Raised to a Power	$p = 0.5$	●	○	○	○	–
	$p = 0.7$	●	○	○	○	–
	$p = 1$	●	○	●	●	–
	$p = 1.2$	●	○	○	○	–
	$p = 1.5$	●	○	○	○	–
Dashpot Coefficient Function $\eta(\dot{u}_\eta)$	$p = 0.5$	●	○	○	○	–
	$p = 0.7$	●	○	○	○	–
	$p = 1$	●	○	●	●	–
	$p = 1.2$	●	○	○	○	–
	$p = 1.5$	●	○	○	○	–
Dashpot Coefficient Function $\eta(\dot{u}_{el})$	$p = 0.5$	●	○	●	●	●
	$p = 0.7$	●	○	●	●	●
	$p = 1$	●	○	●	●	●
	$p = 1.2$	●	○	●	●	●
	$p = 1.5$	●	○	●	●	●

Key: ● Model returns expected results ○ Model does not return expected results

Table 8.2 summarises the constant strain rate tests for the non-linear Maxwell elements. These tests were carried out using the dashpot coefficient function as a function of the velocity over the element.

Table 8.2: Matrix of expected behaviour for constant strain rate one-dimensional numerical results.

		Dashpot-Spring	Spring-Dashpot	Hereditary
		Consistent	Consistent	Integral
Non-Linear Maxwell				
Dashpot Coefficient Function $\eta(\dot{u}_{el})$				
	$\dot{u} = 0.01$	●	●	●
	$\dot{u} = 0.1$	●	●	●
$p = 0.5$	$\dot{u} = 1$	●	●	●
	$\dot{u} = 10$	●	●	●
	$\dot{u} = 100$	○	○	●
	$\dot{u} = 0.01$	●	●	●
	$\dot{u} = 0.1$	●	●	●
$p = 0.7$	$\dot{u} = 1$	●	●	●
	$\dot{u} = 10$	●	●	●
	$\dot{u} = 100$	○	○	●
	$\dot{u} = 0.01$	●	●	●
	$\dot{u} = 0.1$	●	●	●
$p = 1$	$\dot{u} = 1$	●	●	●
	$\dot{u} = 10$	●	●	●
	$\dot{u} = 100$	○	○	●
	$\dot{u} = 0.01$	●	●	●
	$\dot{u} = 0.1$	●	●	●
$p = 1.2$	$\dot{u} = 1$	●	●	●
	$\dot{u} = 10$	●	●	●
	$\dot{u} = 100$	●	○	●
	$\dot{u} = 0.01$	●	●	●
	$\dot{u} = 0.1$	●	●	●
$p = 1.5$	$\dot{u} = 1$	●	●	●
	$\dot{u} = 10$	●	●	●
	$\dot{u} = 100$	●	○	●

Key: ● Model returns expected results ○ Model does not return expected results

Chapter 9

Material Model for Cortical Bone

The model used is based on the one proposed by Cloete *et al.* [3] in Chapter 5.3 and comprised of the same sub-elements. This model focuses on capturing the rate-dependent transition effects and on a non-linear Maxwell sub-element which was introduced in Chapter 6.3.3. The difference between the proposed model and that in the literature is the implementation of the non-linear Maxwell elements.

9.1 Constitutive Equations for the Material Model of Cortical Bone

The hereditary integral formulation with the dashpot as a function of the elemental strain rate is used (equation (6.36)), and is consistent with the phenomenological model developed by Cloete *et al.* [3]. This consists of a Kelvin-Voigt model with a non-linear dashpot in parallel with two non-linear Maxwell elements, shown in Figure 9.1.

The total stress response of the material model for cortical bone at time t is made up of the sum of the response of the individual elements at that time

$$\sigma_{Cortical}(t) = \sigma_{E_{KV}}(t) + \sigma_{\eta_{KV}}(t) + \sigma_{M1}(t) + \sigma_{M2}(t). \tag{9.1}$$

The Kelvin-Voigt spring with modulus E_{KV} captures the elastic behaviour at low strain rates, and the stress across this element at time t is

$$\sigma_{E_{KV}}(t) = E_{KV}\left(\frac{\dot{\varepsilon}(t)}{\dot{\varepsilon}_{ref\,E_{KV}}}\right), \tag{9.2}$$

where $\dot{\varepsilon}(t)$ is the strain rate at time t, and the reference strain rate, $\dot{\varepsilon}_{ref\,E_{KV}}$, is a constant.

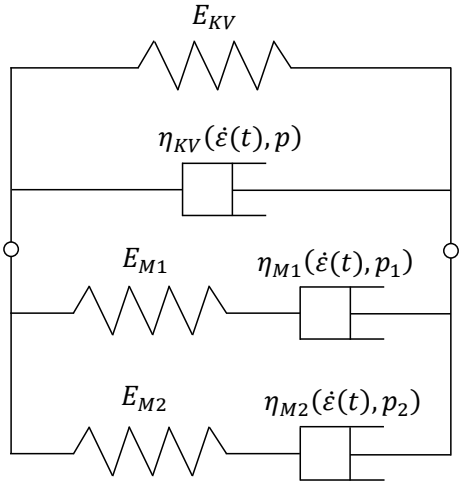


Figure 9.1: Material model for cortical bone.

The Kelvin-Voigt dashpot with coefficient $\eta_{KV}(\dot{\varepsilon}(t), p)$ captures the rate hardening behaviour observed at high strain rates. The stress across this element at time t is

$$\sigma_{\eta_{KV}}(t) = \bar{\eta}_{KV} \left(\frac{\dot{\varepsilon}(t)}{\dot{\varepsilon}_{ref\,\eta_{KV}}} \right)^p \tag{9.3}$$

where the power p determines the degree of rate hardening, $\dot{\varepsilon}(t)$ is the strain rate at time t, and the reference strain rate, $\dot{\varepsilon}_{ref\,\eta_{KV}}$, is a constant.

The combination of the Kelvin-Voigt spring and the two Maxwell springs with moduli E_{M1} and E_{M2} capture the elastic behaviour at high strain rates.

The first Maxwell dashpot with coefficient $\eta_{M1}(\dot{\varepsilon}(t), p_1)$ captures the steep transition behaviour in the ISR regime with a value of p_1 greater than one. The stress across this Maxwell element is

$$\sigma_{M1}(t) = \sigma_{M1}(t - \Delta t) \left(e^{-\frac{E_{M1}}{\eta_{M1}(\dot{\varepsilon}(t - \Delta t))} \Delta t} \right) + \eta_{M1}(\dot{\varepsilon}(t)) \left(1 - e^{-\frac{E_{M1}}{\eta_{M1}(\dot{\varepsilon}(t))} \Delta t} \right) \dot{\varepsilon}(t) \tag{9.4}$$

where $\sigma_{M1}(t - \Delta t)$ is the stress across the element at the end of the previous time step and the dashpot coefficient function is

$$\eta_{M1}(\dot{\varepsilon}(t)) = \bar{\eta}_{M1} \left(1 + \left(\frac{\dot{\varepsilon}(t)}{\dot{\varepsilon}_{ref_{M1}}} \right)^{p_1 - 1} \right). \tag{9.5}$$

The second Maxwell dashpot with coefficient $\eta_{M2}(\dot{\varepsilon}(t), p_2)$ captures the gradual transition behaviour

at low strain rates with a value of p_2 less than one. The stress across this Maxwell element is

$$\sigma_{M2}(t) = \sigma_{M2}(t-\Delta t)\left(e^{-\frac{E_{M2}}{\eta_{M2}(\dot{\varepsilon}(t-\Delta t))}\Delta t}\right) + \eta_{M2}(\dot{\varepsilon}(t))\left(1-e^{-\frac{E_{M2}}{\eta_{M2}(\dot{\varepsilon}(t))}\Delta t}\right)\dot{\varepsilon}(t) \tag{9.6}$$

where $\sigma_{M2}(t-\Delta t)$ is the stress across the element at the end of the previous time step, and the dashpot coefficient function is

$$\eta_{M2}(\dot{\varepsilon}(t)) = \bar{\eta}_{M2}\left(1+\left(\frac{\dot{\varepsilon}(t)}{\dot{\varepsilon}_{ref_{M2}}}\right)^{p_2-1}\right). \tag{9.7}$$

9.1.1 Calibration of $\dot{\varepsilon}_{ref}$ to Experimental Data

As the value of the power p changes, the gradient of the transition of the stress - strain rate curve changes, however, the point at which the transition starts does not change. The stress - strain rate curve needs to be shifted along the x-axis which can be done by adjusting the reference strain rate. Figure 9.2 shows a stress - strain rate curve for a power $p = 1$. Increasing the value of p to 3 shows a shift of the curve to the left. In order to shift the curve back to the right the stress at the inflection point for both values of p must be equal.

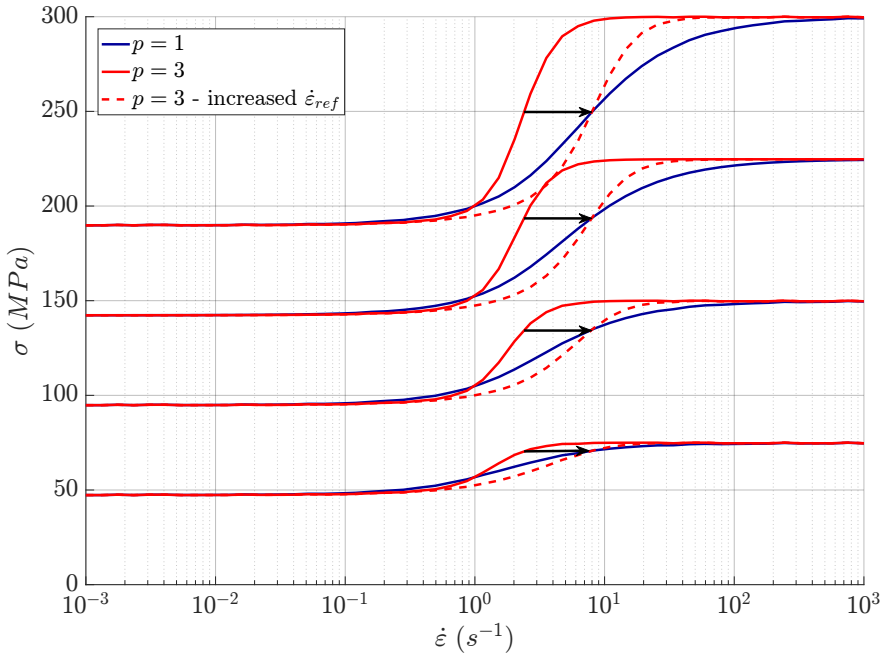


Figure 9.2: Reference strain rate calibration.

The dashpot coefficient function is the only part of the stress that is affected by a change in the power p, see equations (9.6) and (9.7). This means that the dashpot coefficient function at the

strain rate inflection point, $\dot{\varepsilon}_{\text{flex}}$, of the transition curve must be equal for a general value of the power p and $\dot{\varepsilon}_{ref}$ and a known value pair referred to as p_{norm} and $\dot{\varepsilon}_{ref\ \text{norm}}$

$$\eta\left(\dot{\varepsilon}_{\text{flex}}, \dot{\varepsilon}_{ref}, p\right) = \eta\left(\dot{\varepsilon}_{\text{flex}}, \dot{\varepsilon}_{ref\ \text{norm}}, p_{\text{norm}}\right) \tag{9.8}$$

The function for the dashpot coefficient is defined as

$$\eta(\dot{\varepsilon}, \dot{\varepsilon}_{ref}, p) = \bar{\eta}\left(1 + \left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_{ref}}\right)^{p-1}\right). \tag{9.9}$$

Equating the dashpot coefficients by evaluating equation (9.8) gives

$$\bar{\eta}\left(1 + \left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref}}\right)^{p-1}\right) = \bar{\eta}\left(1 + \left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref\ \text{norm}}}\right)^{p_{\text{norm}}-1}\right) \tag{9.10}$$

which, when rearranged and simplified, gives an equation for the reference strain rate in terms of the power, p,

$$\dot{\varepsilon}_{ref} = \dot{\varepsilon}_{\text{flex}}\left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref\ \text{norm}}}\right)^{-\left(\frac{p_{\text{norm}}-1}{p-1}\right)}. \tag{9.11}$$

Equation (9.11) does not allow for values of $p = 1$, however, this can be substituted back into equation (9.9) to give

$$\begin{aligned}
\eta(\dot{\varepsilon}) &= \bar{\eta}\left(1 + \left(\frac{\dot{\varepsilon}}{\dot{\varepsilon}_{\text{flex}}\left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref\ \text{norm}}}\right)^{-\left(\frac{p_{\text{norm}}-1}{p-1}\right)}}\right)^{p-1}\right) \\
&= \bar{\eta}\left(1 + \left(\frac{\dot{\varepsilon}^{p-1}}{\dot{\varepsilon}_{\text{flex}}^{p-1}\left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref\ \text{norm}}}\right)^{1-p_{\text{norm}}}}\right)\right) \\
&= \bar{\eta}\left(1 + \left(\dot{\varepsilon}_{\text{flex}}^{1-p}\left(\frac{\dot{\varepsilon}_{\text{flex}}}{\dot{\varepsilon}_{ref\ \text{norm}}}\right)^{p_{\text{norm}}-1}\dot{\varepsilon}^{p-1}\right)\right)
\end{aligned} \tag{9.12}$$

which can accommodate a value of $p = 1$.

9.2 Parameters for Cortical Bone Material Model

A material model for cortical bone was proposed using the same parameters to those used by Cloete *et al.* [3], with the addition of the use of reference strain rates. The development of the calibration of the reference strain rate in the dashpot coefficient function above was also implemented in this model, and the norm and inflection values are also outlined in the table below. The proposed material model uses the formulation of the dashpot coefficient function in equation (9.12) for varying values of p.

Table 9.1: Parameters for cortical bone material model.

		Proposed Material Model	Cloete *et al.* [3] Model
E_{KV}	(MPa)	9000	9000
$\dot{\varepsilon}_{ref\ E_{KV}}$	(s^{-1})	1	-
η_{KV}	(MPa s)	0.2	0.2
$\dot{\varepsilon}_{ref\ \eta_{KV}}$	(s^{-1})	1	-
p		0.7	0.7
E_{M1}	(MPa)	1000	1000
$\bar{\eta}_{M1}$	(MPa s)	40	50
p_1		0.5	0.5
$\dot{\varepsilon}_{ref\ \text{norm}\ M1}$	(s^{-1})	1	-
$p_{1,\text{norm}\ M1}$		0.5	-
$\dot{\varepsilon}_{\text{flex}\ M1}$	(s^{-1})	6.5	-
E_{M2}	(MPa)	5500	5500
$\bar{\eta}_{M2}$	(MPa s)	0.1	0.15
p_2		3	3
$\dot{\varepsilon}_{ref\ \text{norm}\ M2}$	(s^{-1})	0.85	-
$p_{2,\text{norm}\ M2}$		3	-
$\dot{\varepsilon}_{\text{flex}\ M2}$	(s^{-1})	6.5	-

For the values defined above, the dashpot coefficient function is shown on a log plot in Figure 9.3. The blue curve represents the dashpot coefficient function for the Maxwell element $M1$, with a value of $p_1 = 0.5$. As the strain rate tends towards zero the value of the dashpot coefficient tends towards a large number. As the strain rate tends toward large magnitude values, the dashpot coefficient value tends towards $\bar{\eta}_{M1}$, the constant portion of equation (9.12).

The orange curve in Figure 9.3 represents the dashpot coefficient function for the Maxwell element $M2$, with a value of $p_2 = 2.5$. With $p > 0$, as the strain rate tends towards zero, the value of the dashpot coefficient function tends towards $\bar{\eta}_{M2}$, the constant portion of equation (9.12). As the strain rate tends towards large magnitude values, the value of the dashpot coefficient function increases.

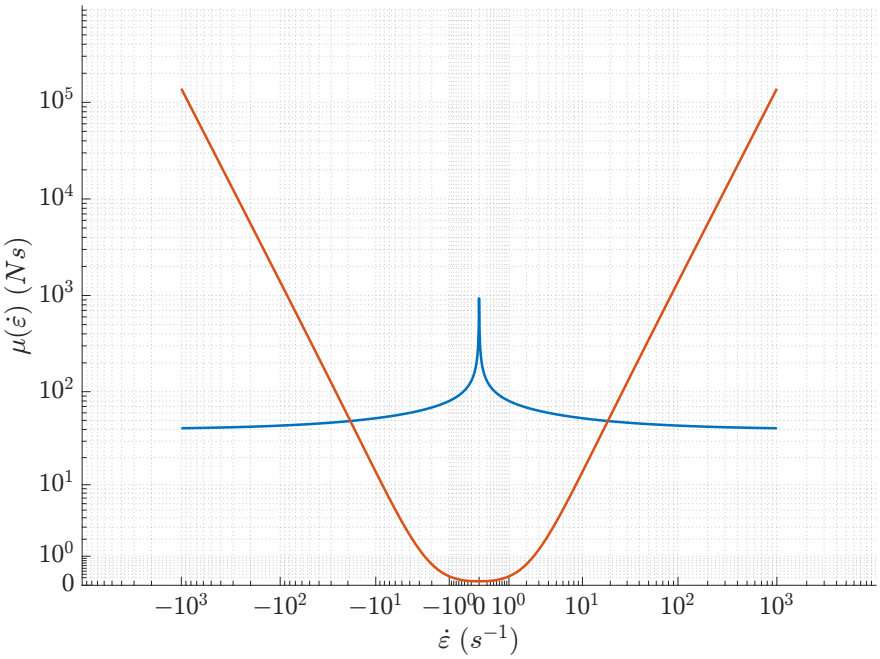


Figure 9.3: Cortical bone material model - dashpot coefficient function.

Using the model shown in Figure 9.1, with the values defined in Table 9.1, a constant strain rate was applied and the stress was sampled at four different strains matching those recorded by Cloete *et al.* [3]. This was carried out for several different strain rates between $10^{-3}s^{-1}$ and $10^{3}s^{-1}$. The sampled stress for each of the strains for each of the strain rate simulations is plotted on a stress - strain rate plot, Figure 9.4. This shows the strain rate dependent behaviour of the one-dimensional material mode for cortical bone and the hereditary integral implementation, compared to the experimental data from the work by Cloete *et al.* [3] and their theoretical implementation.

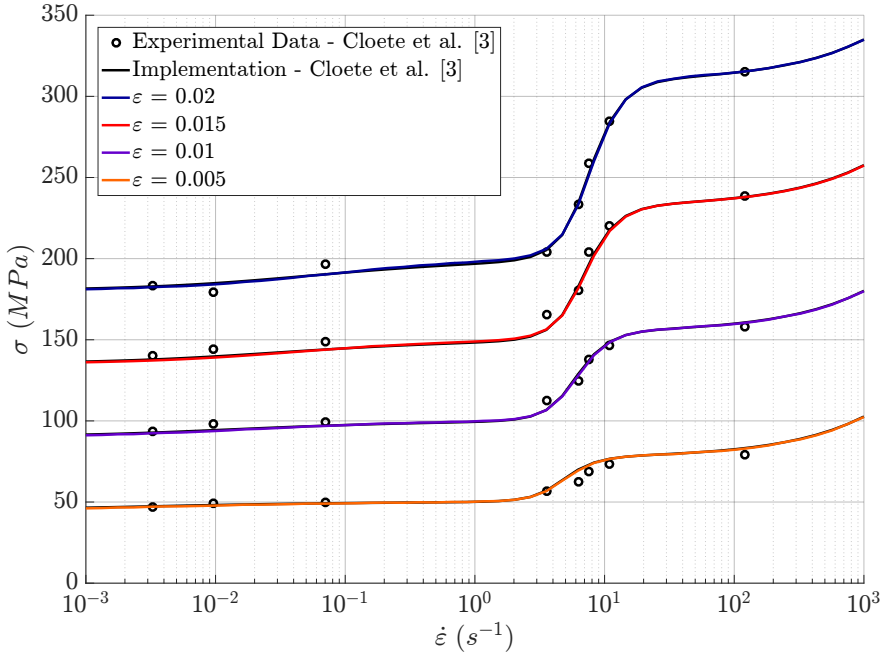


Figure 9.4: Material model for cortical bone - Stress - Strain Rate graphs.

Chapter 10

Implementation of Three-Dimensional Formulation in FEA Software

The three-dimensional formulation was implemented in two different commercial finite element analysis (FEA) software packages, LS Dyna and Abaqus. Initially only LS Dyna was used, but during the course of verification, issues were found which required the use of a second software package, Abaqus, for an independent comparison. After the comparison it was decided to continue with Abaqus for the remainder of the simulations. This chapter looks at the general implementation of the three-dimensional models outlined in Chapter 4 and an implementation of the non-linear Maxwell element as discussed in Chapter 6.3 in three dimensions.

10.1 UMAT/VUMAT

Implementing a user material model in a commercial software package requires a user subroutine. In LS Dyna, the user material subroutine, UMAT, was used. This was written in Fortran and compiled to produce a binary executable file [24]. In Abaqus, a vectorised user material subroutine, VUMAT, was used, which is a vector form of UMAT. The VUMAT subroutine was also written in Fortran and supplied to Abaqus as a user subroutine file. In both cases, the user material model requires input parameters to be specified.

10.1.1 Stress and Strain in LS Dyna and Abaqus

In both the UMAT in LS Dyna and the VUMAT in Abaqus, the stress is updated in every time step. This is done incrementally using the strain increment, $\Delta\varepsilon_t$, between the previous and current

time steps.

The total strain is calculated by a summation of the strain increments over time. This can be expressed as

$$\varepsilon_{tot} = \sum_{t=0}^{t_{tot}} \Delta\varepsilon(t) = \sum_{t} \frac{L(t) - L(t-1)}{L(0)} = \sum_{t} \frac{\delta L}{L(0)} \tag{10.1}$$

$$\lim_{\Delta t \to 0} \sum_{t} \Delta\varepsilon(t) = \int_0^t \frac{\delta L}{L(0)} = \ln(L)|_0^t = \ln(L(t)) - \ln(L(0)) = \ln\left(\frac{L(t)}{L(0)}\right) \tag{10.2}$$

Strain computed in this manner is logarithmic strain, which is what is outputted by both LS Dyna and Abaqus.

In the implementation of the user material model, the strain at the previous and current time steps are tracked in arrays, and updated for each entry in the strain tensor as

$$\varepsilon_{ij}(t) = \varepsilon_{ij}(t - \Delta t) + \Delta\varepsilon_{ij}(t) \tag{10.3}$$

where $\varepsilon_{ij}(t)$ is the strain at the current time t for index ij in the strain tensor which is stored in a variable at the end of the time step to be used in the next time step. The strain at the previous time step $\varepsilon_{ij}(t - \Delta t)$ for the entry ij in the strain tensor is stored at the end of the previous time step, and $\Delta\varepsilon_{ij}(t)$ is the strain increment provided in the current time step for the entry ij in the strain tensor.

The stress $\sigma_{ij}(t)$ is then updated using either the incrementally calculated strain $\varepsilon_{ij}(t)$ or the strain increment $\Delta\varepsilon_{ij}(t)$. For each sub-element of the visco-elastic model, the stress $\sigma_{ij}(t)$ is calculated in a different way which is outlined in the following sections.

10.1.2 Spring Sub-Element

The implementation of the stress of a spring sub-element in the user-defined model uses equation (4.4) that was developed in Chapter 4.1

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^E (t) = \begin{bmatrix} E\left(\lambda_1(\varepsilon_{11}(t) + \varepsilon_{22}(t) + \varepsilon_{33}(t)) + \lambda_2\varepsilon_{11}(t)\right) \\ E\left(\lambda_1(\varepsilon_{11}(t) + \varepsilon_{22}(t) + \varepsilon_{33}(t)) + \lambda_2\varepsilon_{22}(t)\right) \\ E\left(\lambda_1(\varepsilon_{11}(t) + \varepsilon_{22}(t) + \varepsilon_{33}(t)) + \lambda_2\varepsilon_{33}(t)\right) \\ E\left(\lambda_2\varepsilon_{23}(t)\right) \\ E\left(\lambda_2\varepsilon_{13}(t)\right) \\ E\left(\lambda_2\varepsilon_{12}(t)\right) \end{bmatrix} \tag{10.4}$$

where λ_1 and λ_2 are

$$\lambda_1 = \frac{\nu}{(1+\nu)(1-2\nu)} \qquad \text{and} \qquad \lambda_2 = \frac{1}{1+\nu} \tag{10.5}$$

and the strains ε_{ij} were calculated as per equation (10.3).

10.1.3 Dashpot Sub-Element

For a dashpot sub-element the stress was calculated using equation (4.19) that was developed in Chapter 4.2. This was written in terms of the constants λ_1 and λ_2 (equations (10.5))

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^V (t) = \begin{bmatrix} \eta\left(\lambda_1(\dot{\varepsilon}_{11}(t) + \dot{\varepsilon}_{22}(t) + \dot{\varepsilon}_{33}(t)) + \lambda_2\dot{\varepsilon}_{11}(t)\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11}(t) + \dot{\varepsilon}_{22}(t) + \dot{\varepsilon}_{33}(t)) + \lambda_2\dot{\varepsilon}_{22}(t)\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11}(t) + \dot{\varepsilon}_{22}(t) + \dot{\varepsilon}_{33}(t)) + \lambda_2\dot{\varepsilon}_{33}\right) \\ \eta\lambda_2\dot{\varepsilon}_{23}(t) \\ \eta\lambda_2\dot{\varepsilon}_{13}(t) \\ \eta\lambda_2\dot{\varepsilon}_{12}(t) \end{bmatrix} \tag{10.6}$$

where the strain rates $\dot{\varepsilon}_{ij}$ were calculated using the strain increments

$$\dot{\varepsilon}_{ij}(t) = \frac{\Delta\varepsilon_{ij}(t)}{\Delta t}. \tag{10.7}$$

10.1.4 Maxwell Sub-Element

The Maxwell element requires the tracking of the stress at the previous time step and therefore this was stored in a variable in the user-defined model.

Linear Maxwell Sub-Element

For the linear Maxwell element, the stress equation used was

$$\boldsymbol{\sigma}^M(t) = e^{-\frac{\Delta t}{\theta}}\boldsymbol{\sigma}^M(t-\Delta t) + \eta\left(1 - e^{-\frac{\Delta t}{\theta}}\right)\dot{\boldsymbol{\varepsilon}}^{3D}(t) \tag{10.8}$$

where

$$\theta = \frac{\eta}{E}, \tag{10.9}$$

and the three-dimensional strain rate tensor $\dot{\boldsymbol{\varepsilon}}^{3D}(t)$ is defined as in equation (4.15). Substituting in the definition for the three-dimensional strain rate tensor gives the stress equation as

$$\boldsymbol{\sigma}^M(t) = e^{-\frac{\Delta t}{\theta}}\boldsymbol{\sigma}^M(t-\Delta t) + \eta\left(1 - e^{-\frac{\Delta t}{\theta}}\right)\left(\lambda_1 tr\left(\dot{\boldsymbol{\varepsilon}}(t)\right)\boldsymbol{I} + \lambda_2\dot{\boldsymbol{\varepsilon}}(t)\right). \tag{10.10}$$

This can be written in Voigt notation and was implemented as

$$\begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^M (t) = e^{-\frac{\Delta t}{\theta}} \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{23} \\ \sigma_{13} \\ \sigma_{12} \end{bmatrix}^M (t - \Delta t) + \left(1 - e^{-\frac{\Delta t}{\theta}}\right) \begin{bmatrix} \eta\left(\lambda_1(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{11}\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{22}\right) \\ \eta\left(\lambda_1(\dot{\varepsilon}_{11} + \dot{\varepsilon}_{22} + \dot{\varepsilon}_{33}) + \lambda_2\dot{\varepsilon}_{33}\right) \\ \eta\lambda_2\dot{\varepsilon}_{23} \\ \eta\lambda_2\dot{\varepsilon}_{13} \\ \eta\lambda_2\dot{\varepsilon}_{12} \end{bmatrix}. \quad (10.11)$$

Non-linear Maxwell Sub-Element

Using the concept of having the dashpot coefficient as a function of the strain rate where the general form of the function is

$$\eta(\dot{\varepsilon}) = \eta\left(\frac{|\dot{\varepsilon}|}{\dot{\varepsilon}_{ref}}\right)^{n-1}, \quad (10.12)$$

there are three possible 3D implementations. The function, $\eta(\dot{\varepsilon})$, can be defined in each direction as

$$\eta(\dot{\varepsilon}_{ij}) = \eta\left(\frac{|\dot{\varepsilon}_{ij}|}{\dot{\varepsilon}_{ref}}\right)^{n-1}. \quad (10.13)$$

Alternatively, the function $\eta(\dot{\varepsilon})$ can be defined as a function of the dominant strain rate, for example $\dot{\varepsilon}_{11}$, as

$$\eta(\dot{\varepsilon}_{11}) = \eta\left(\frac{|\dot{\varepsilon}_{11}|}{\dot{\varepsilon}_{ref}}\right)^{n-1}. \quad (10.14)$$

Or as a function of an equivalent strain rate, $\dot{\varepsilon}_{eq}$, as

$$\eta(\dot{\varepsilon}_{eq}) = \eta\left(\frac{|\dot{\varepsilon}_{eq}|}{\dot{\varepsilon}_{ref}}\right)^{n-1}. \quad (10.15)$$

This equivalent strain rate can be any normalised value of the global strain rate. In this dissertation, the equivalent strain rate will be that which was presented by Bekker *et al.* [2], and will be defined as

$$\begin{aligned} \dot{\varepsilon}_{eq} &= \sqrt{\frac{\dot{\boldsymbol{\varepsilon}} : \dot{\boldsymbol{\varepsilon}}}{1 + 2\nu^2}} \\ &= \sqrt{\frac{\dot{\varepsilon}_{11}^2 + \dot{\varepsilon}_{22}^2 + \dot{\varepsilon}_{33}^2 + 2\dot{\varepsilon}_{12}^2 + 2\dot{\varepsilon}_{13}^2 + 2\dot{\varepsilon}_{23}^2}{1 + 2\nu^2}}. \end{aligned} \quad (10.16)$$

10.2 Mass Scaling in Abaqus

Mass scaling in Abaqus is used to control the size of the stable time increment [25], especially in quasi-static analyses in order to decrease the computation time of long running simulations. There are two types of mass scaling available for use in the Abaqus explicit solver, fixed or variable mass scaling. This dissertation makes use of fixed mass scaling, which is performed at the beginning of the time step and can be specified either by a mass scaling factor or a minimum stable time step if desired. We will make use of the mass scaling factor in future chapters. The mass scaling factor in Abaqus multiplies the masses of the element by that factor and therefore the density increases by that factor. The dilatational wave speed c_d is calculated using the density as follows

$$c_d = \sqrt{\frac{E}{\rho}} \tag{10.17}$$

where E is the elastic modulus of the material and ρ is the density. If the density is increased by a factor of f^2, the wave speed decreases by a factor of f. The stable time increment is then calculated using this wave speed

$$\Delta t_{crit} = \frac{L_e}{c_d} \tag{10.18}$$

where L_e is the characteristic length of the element and c_d is the dilatational wave speed shown above. If c_d is reduced by a factor of f, the stable time increment would be increased by a factor of f. For the same run time the computation time would decrease by the factor f due to the larger stable time increment.

10.3 Damping in LS Dyna and Abaqus

Both LS Dyna and Abaqus use reduced integration (one point integration) in their solving schemes for first order elements (8-node bricks) [26, 25], to prevent volumetric blocking. One point integration, however, is prone to hourglassing [26], which is the existence of zero-energy modes with no strain or stress at the integration points [27]. The main way in which LS Dyna and Abaqus prevent hourglassing is by applying viscous damping. The disadvantage to this is that the energy in the system is not constant over time and since we are mainly interested in the viscous (i.e. dissipative) effects of the material model, the additional damping could obscure these effects.

The amount of viscous damping applied to the element to prevent hourglassing can be controlled in both Abaqus and LS Dyna by adjusting the bulk viscosity control parameters in the simulation definition. The bulk viscosity control is implemented by an additional pressure term q. This pressure term is defined in LS Dyna as

$$q = \rho l \left(Q_1 l \dot{\varepsilon}_{kk}^2 - Q_2 a \dot{\varepsilon}_{kk} \right) \tag{10.19}$$

where Q_1 is the dimensionless quadratic viscosity coefficient (default value of 1.5), Q_2 is the dimensionless linear viscosity coefficient (default value of 0.06), ρ is the density of the material, l is the characteristic length ($\sqrt{A}$ in two dimensions and $\sqrt[3]{v}$ in three dimensions) and a is the local sound speed [26].

In Abaqus the pressure term is separated into the linear and quadratic terms. The quadratic pressure term is defined as

$$q_1 = \rho \left(Q_1 L_e \dot{\varepsilon}_{vol}\right)^2 \tag{10.20}$$

and the linear pressure term is defined as

$$q_2 = Q_2 \rho c_d L_e \dot{\varepsilon}_{vol} \tag{10.21}$$

where Q_1 is the dimensionless quadratic viscosity coefficient (default value of 1.2), Q_2 is the dimensionless linear viscosity coefficient (default value of 0.06), ρ is the density of the material, L_e is the characteristic length, c_d is the current dilatational wave speed and $\dot{\varepsilon}_{vol}$ is the volumetric strain rate [28].

The linear hourglassing control parameter Q_2 needs to be small enough that the dissipative effects of the extra damping do not interfere with the dissipative effect of the Maxwell dashpot. This parameter does, however, need to be large enough that hourglassing is prevented. Varying values of this hourglassing control parameter were investigated and the results are outlined in the following chapter.

Chapter 11

Results of Simulations

By analysing the results of the simulation, the LS-Dyna and Abaqus implementation approaches were first verified using the built-in linear elastic models and a linear elastic user-defined model. The hereditary integral model was then compared to FEA simulations that were run in Abaqus. Finally, the material model for cortical bone was simulated in Abaqus and compared to the experimental data from Cloete *et al.* using a three-dimensional quarter symmetry simulation.

11.1 Verification of LS Dyna Simulations

The LS Dyna simulations were verified by simulating a single row of linear elastic elements shown in Figure 11.1. This mesh was constrained, on all edges, not to move in the y and z directions and an impulse stress wave $\sigma(t)$ was applied to one end in the negative x direction. Varying values of the hourglassing control parameter Q_2 were used.

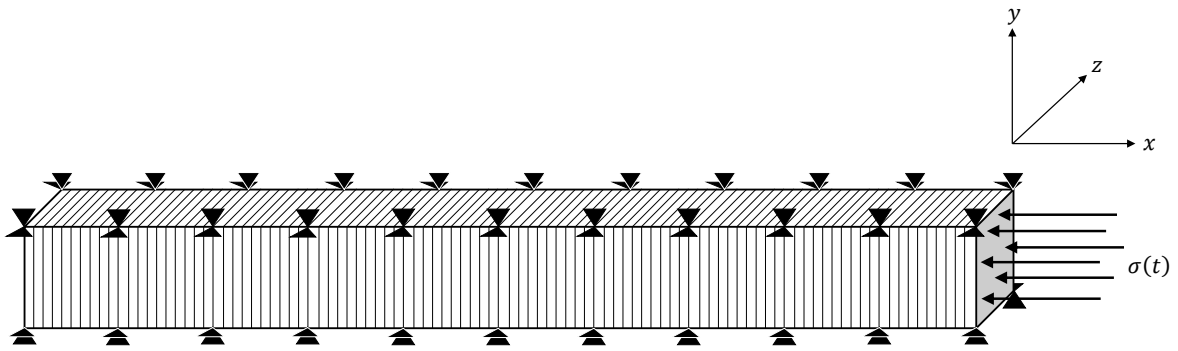


Figure 11.1: Row of 100 elements constrained to move only in the x-direction and an applied stress $\sigma(t)$ in the negative x-direction.

The simulation was run using the built-in linear elastic model and a user-defined linear elastic model (the code for which can be found in Appendix A.1) with an elastic modulus of $20.7GPa$, a density of $1500kg\,m^{-3}$ and a Poisson's ratio of 0.3. The applied stress wave follows an S-shape curve from zero to the maximum applied stress of $-1015MPa$ and follows the inverted S-shape curve until the stress applied returns to zero, over a time of $9.2 \times 10^{-6}s$. The cross sectional dimensions of the mesh were $0.008862m \times 0.008862m$ with a total length of $0.1m$.

A typical result of these tests is shown in Figures 11.2 and 11.3, where the applied stress over time is shown in the first graph of each figure. In Figure 11.2 the displacement of the end nodes and the middle node was plotted for the built-in linear elastic model, a user-defined linear elastic model and a one-dimensional spring-mass system using the incremental time-stepping scheme.

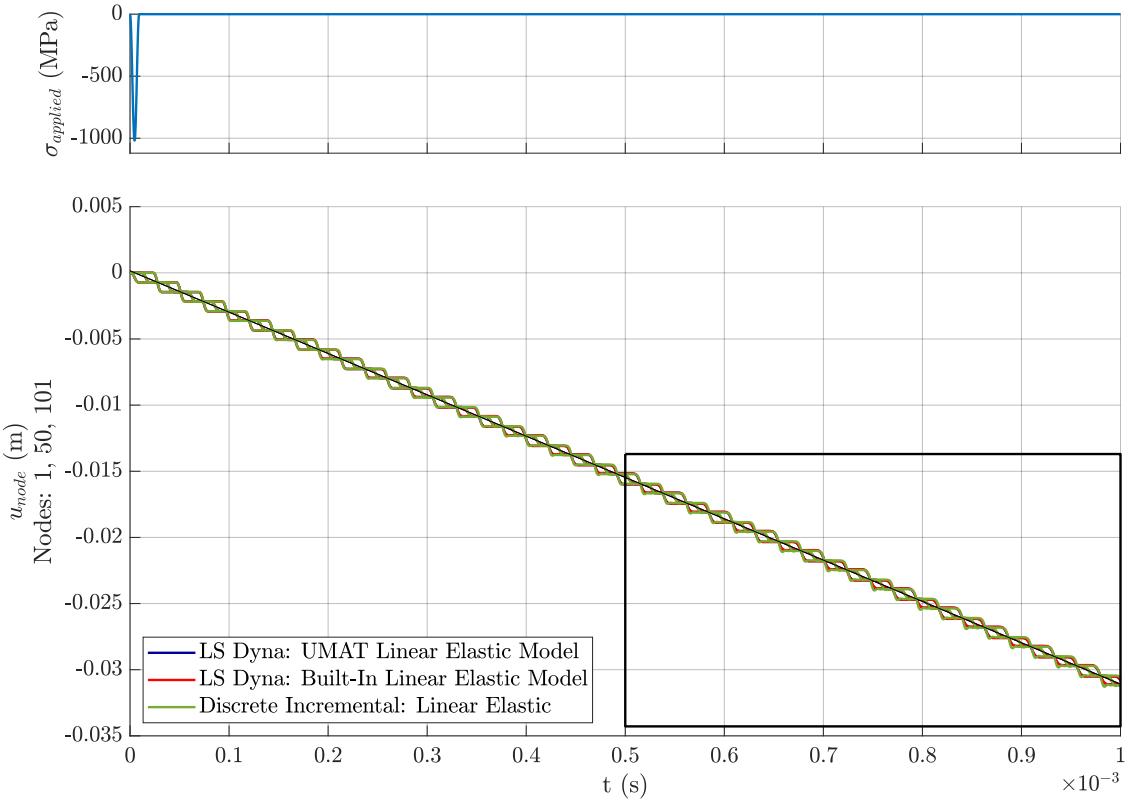


Figure 11.2: The applied stress-time graph and displacement-time graph comparison of the results from LS Dyna of a linear elastic model with no hourglassing control and a one-dimensional spring-mass system.

In Figure 11.3 the strain and stress of the middle element was plotted. This shows the shape of the stress and strain waves that move through the row of elements over time. For the results from the LS Dyna simulations the amplitude of the wave diminishes over time which is due to the bulk viscosity that is applied in order to prevent hourglassing.

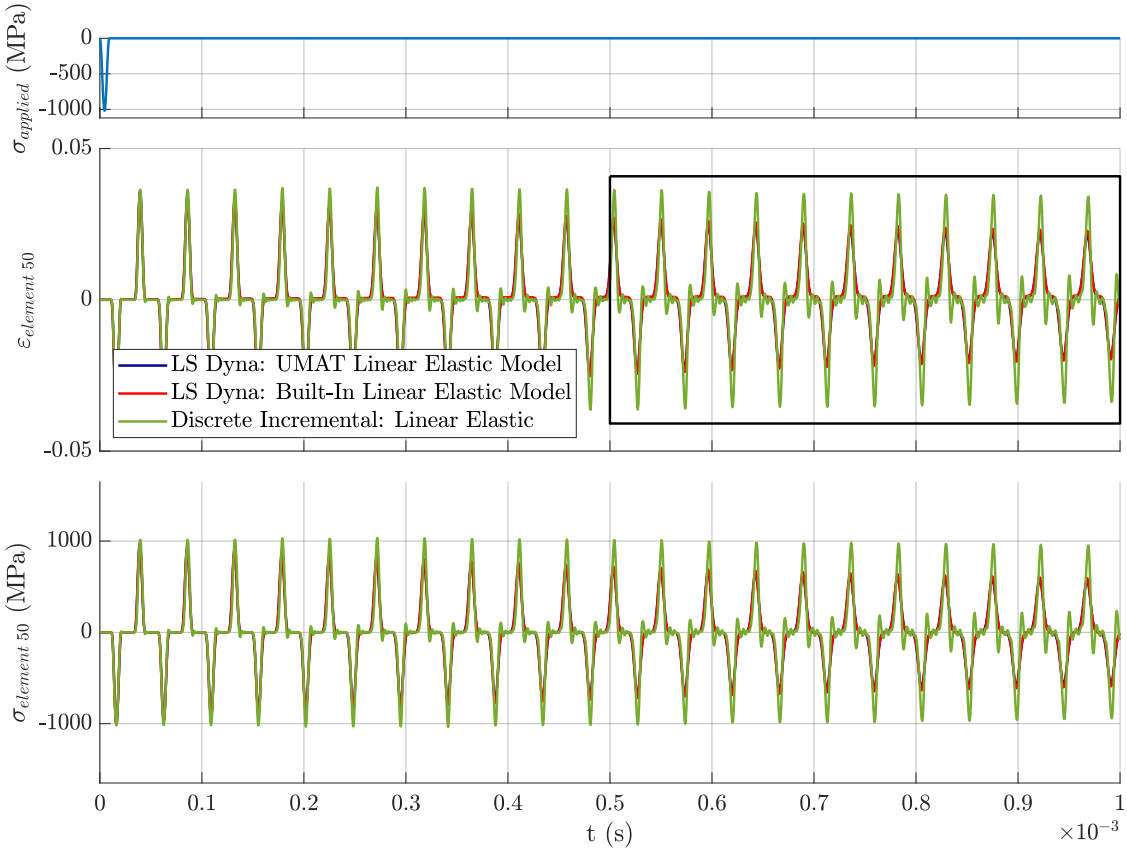


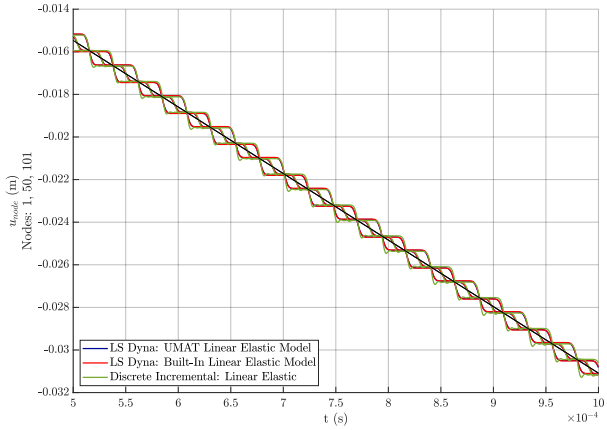
Figure 11.3: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs.

Hourglassing control was done through the bulk viscosity parameters and three scenarios were investigated: no hourglassing control (i.e. default bulk viscosity parameters were used); hourglassing control turned on using the default values $Q_1 = 1.2$ and $Q_2 = 6 \times 10^{-2}$; and hourglassing control parameters $Q_1 = 1.2$ and $Q_2 = 6 \times 10^{-4}$. An enlarged portion of the displacement and strain graphs are shown side by side in Figure 11.4 for each of the hourglassing control scenarios. The full-size versions of these results can be found in Appendix C.1.

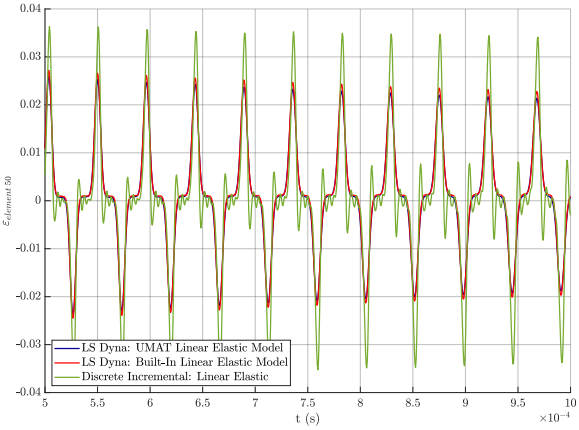
The displacement of the nodes and the strain of the elements from the LS Dyna simulations should match those from the one-dimensional spring-mass system for a linear elastic model. Figures 11.4a and 11.4b show the displacement and strain of the LS Dyna simulations with the hourglassing control turned off. The amplitude of the strain graph is decreasing and there is a slight deviation of the LS Dyna displacements from the one-dimensional spring-mass system displacement. The hourglassing control was then turned on and the default values were specified, displacement and strain graphs in Figures 11.4c and 11.4d show the same decrease in amplitude of the LS Dyna strain graph and the difference in displacements remains the same.

For an hourglassing control parameter $Q_2 = 6 \times 10^{-4}$, the displacement and strain graphs are shown in Figures 11.4e and 11.4f. The strain from the LS Dyna results was calculated logarithmically from the displacements and outputted directly from the results. This shows that the amplitude

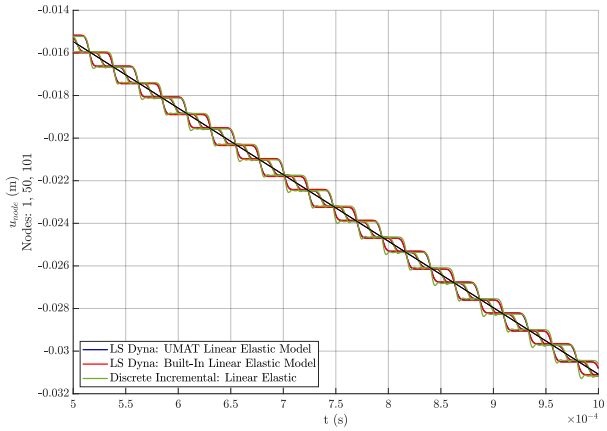
of the LS Dyna strain is not decreasing, however, there is a drift of the logarithmic strain graphs from the zero line. The displacement graph also shows a significant deviation of the built-in and user-defined model results from the one-dimensional spring-mass system.



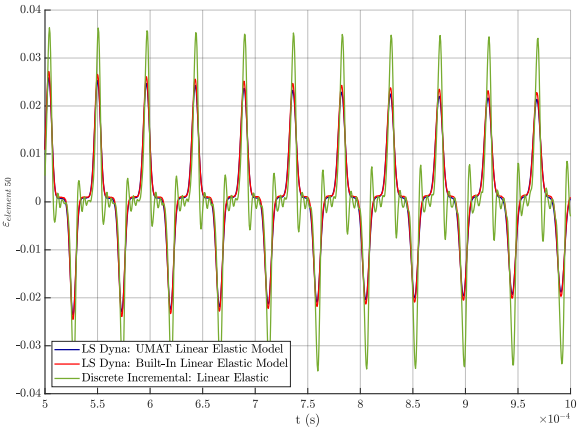
(a) No hourglassing control (Displacement-Time graph).



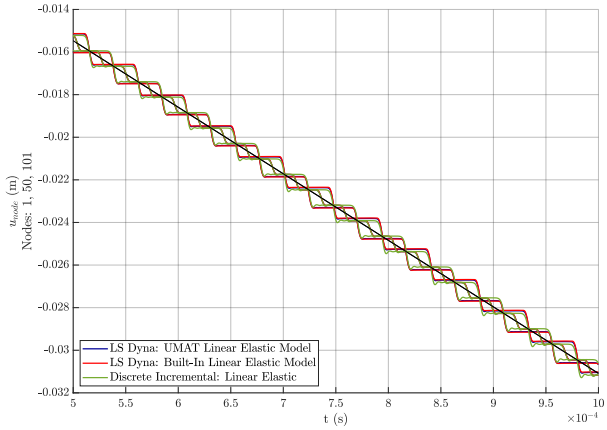
(b) No hourglassing control (Strain-Time graph).



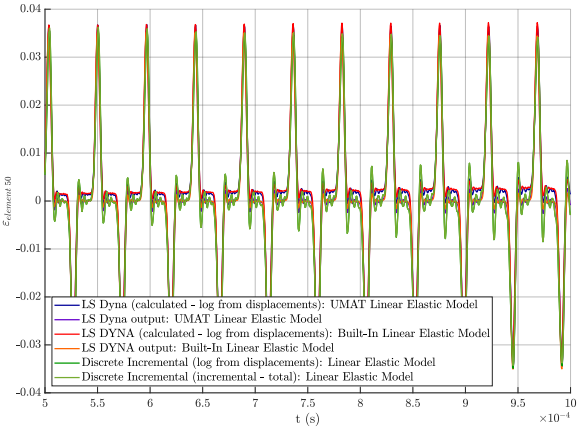
(c) Hourglassing control parameter $Q_2 = 6 \times 10^{-2}$ (Displacement-Time graph).



(d) Hourglassing control parameter $Q_2 = 6 \times 10^{-2}$ (Strain-Time graph).



(e) Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).



(f) Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).

Figure 11.4: Linear Elastic Model - LS Dyna Built-In vs UMAT vs one-dimensional spring-mass system incremental time-stepping scheme.

11.2 LS-Dyna vs Abaqus Simulations Validation

The LS Dyna simulations were then validated against Abaqus simulations. The same simulations with the same material properties and mesh parameters were carried out in Abaqus, simulating a single row of linear elastic elements using both the built-in linear elastic model and a user-defined linear elastic model (the code for which can be found in Appendix B.1). An impulse stress wave was applied to the row of elements and the stress and strain results were examined and compared to those for the LS Dyna simulation and the incremental solution of the one-dimensional spring-mass system. An example of the displacement-time, stress-time and strain-time graphs are shown in Figures 11.5 and 11.6. The full-size results are available in Appendix C.2.

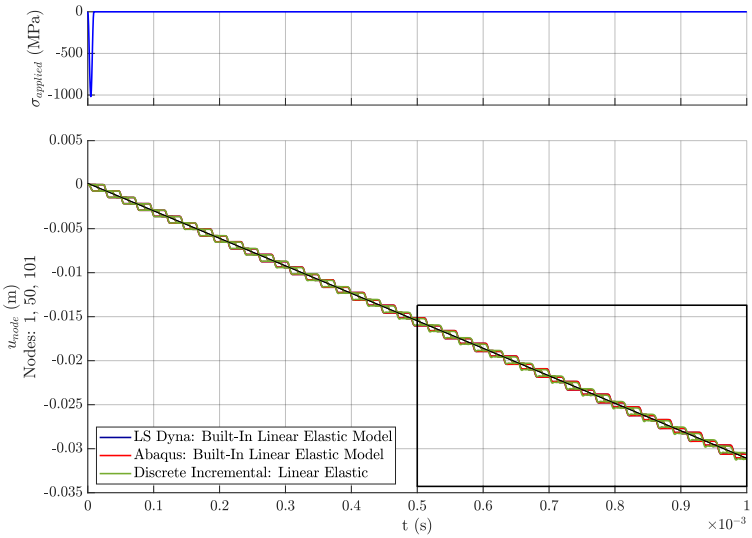


Figure 11.5: Abaqus vs LS-Dyna Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).

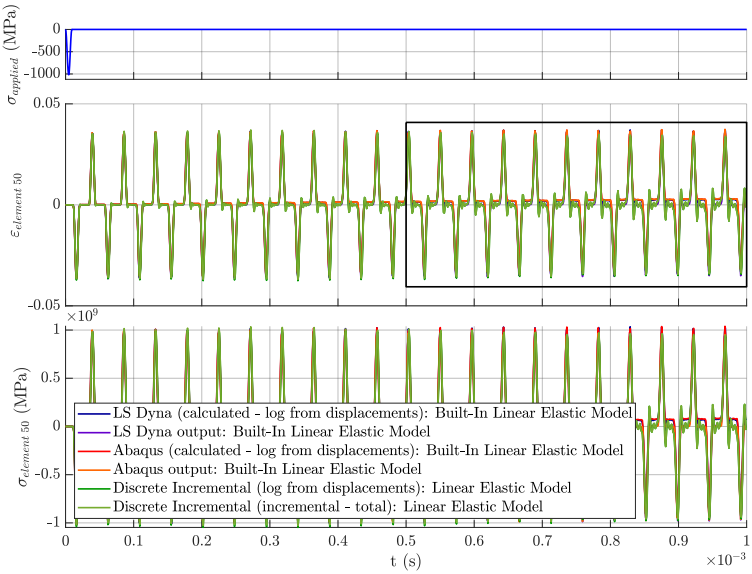
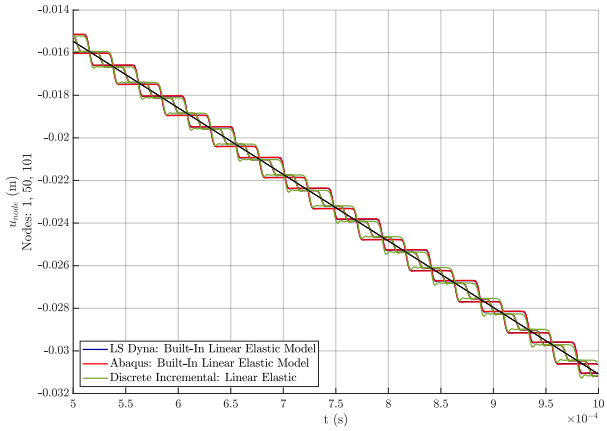


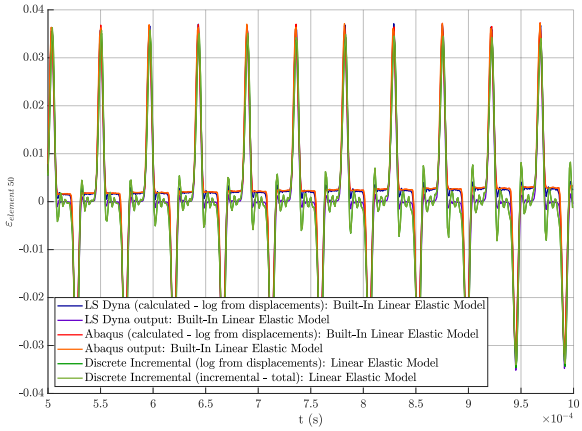
Figure 11.6: Abaqus vs LS-Dyna Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).

The simulations were compared for the built-in linear elastic model and a user-defined linear elastic model between LS Dyna and Abaqus with an hourglass control parameter of $Q_2 = 6 \times 10^{-4}$, and the incremental solution of the one-dimensional spring-mass system. The displacement of the middle and end nodes were plotted over time, and the strain was calculated logarithmically from the displacements as well as being outputted directly from the simulation results. As before, an enlarged portion of the displacement and strain graphs are shown side by side in Figure 11.7.

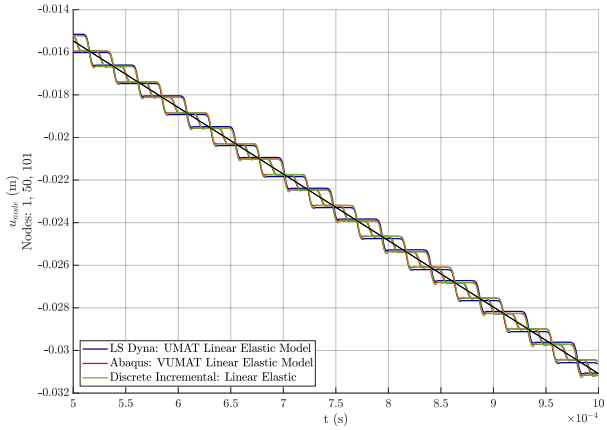
The first comparison is between LS Dyna and Abaqus using the built-in linear elastic model, shown in Figures 11.7a and 11.7b. The displacements and logarithmic strains of both the LS Dyna and Abaqus simulations show a deviation from the one-dimensional spring-mass system results, with the strain drift still prominent. The outputted strain from the LS Dyna built-in linear elastic model tracks the one-dimensional spring-mass system, however, the outputted strain from the Abaqus built-in linear elastic model shows some drift.



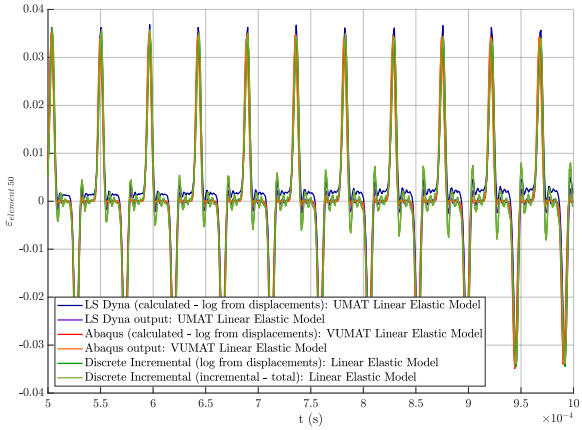
(a) Abaqus vs LS-Dyna Built-In Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).



(b) Abaqus vs LS-Dyna Built-In Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).



(c) Abaqus vs LS-Dyna User-Defined Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).



(d) Abaqus vs LS-Dyna User-Defined Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).

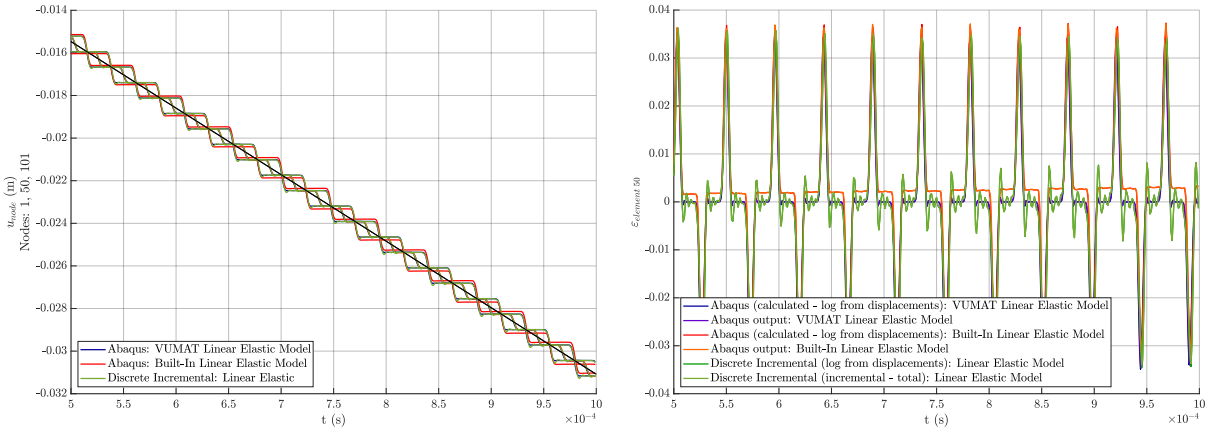
Figure 11.7: Linear Elastic Model - LS Dyna Built-In vs UMAT vs one-dimensional spring-mass system incremental time-stepping scheme.

A user-defined VUMAT was then implemented in Abaqus and compared to the results from LS Dyna. The displacement and strain graphs are shown in Figures 11.7c and 11.7d where the Abaqus user-defined material model matches the one-dimensional spring-mass system for both the displacements and strains. The LS Dyna displacements and strains deviate from the Abaqus and one-dimensional spring-mass system results.

11.3 Abaqus Simulations Validation

The Abaqus simulation results in Figure 11.7, for both the built-in model and a user-defined linear elastic model using an hourglassing control parameter of $Q_2 = 6 \times 10^{-4}$, were compared to the one-dimensional spring-mass system incremental solution, in Figure 11.8.

Figure 11.8a shows that the displacement of the built-in Abaqus linear elastic model deviates from that of the user-defined linear elastic model and the one-dimensional spring-mass system. The strains from the built-in linear elastic model also deviate from that of the user-defined material model and the one-dimensional spring-mass system, see Figure 11.8b.



(a) Abaqus Built-In vs VUMAT Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).

(b) Abaqus Built-In vs VUMAT Linear Elastic Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).

Figure 11.8: Linear Elastic Model - LS Dyna Built-In vs UMAT vs one-dimensional spring-mass system incremental time-stepping scheme.

11.4 Hereditary Integral vs FEA Simulations

The hereditary integral implementation for a linear Maxwell element in three dimensions was validated by simulating a single row of elements, as defined before, using a user-defined standard linear model (the code for which can be found in Appendix B.2). An impulse stress wave was applied to the row of elements and the stress and strain results were examined. An hourglassing control parameter of $Q_2 = 6 \times 10^{-4}$ was used and the displacement of the middle and end nodes was plotted. The material model consisted of a Kelvin-Voigt spring with an elastic modulus of $10.35GPa$, a Maxwell spring with an elastic modulus of $10.35GPa$ and a Maxwell dashpot with a viscosity of $1.7595MPa\ s$.

The displacement-time, stress-time and strain-time graphs are shown in Figures 11.9 and 11.10, where the applied stress is shown in the first graph of each of these figures. The displacements plotted follow a similar path to those from the linear elastic material model with noticeable damping over time. The stress and strain graphs also look similar to previous results with a damping effect of the stress and strain waves which is expected due to the Maxwell dashpot.

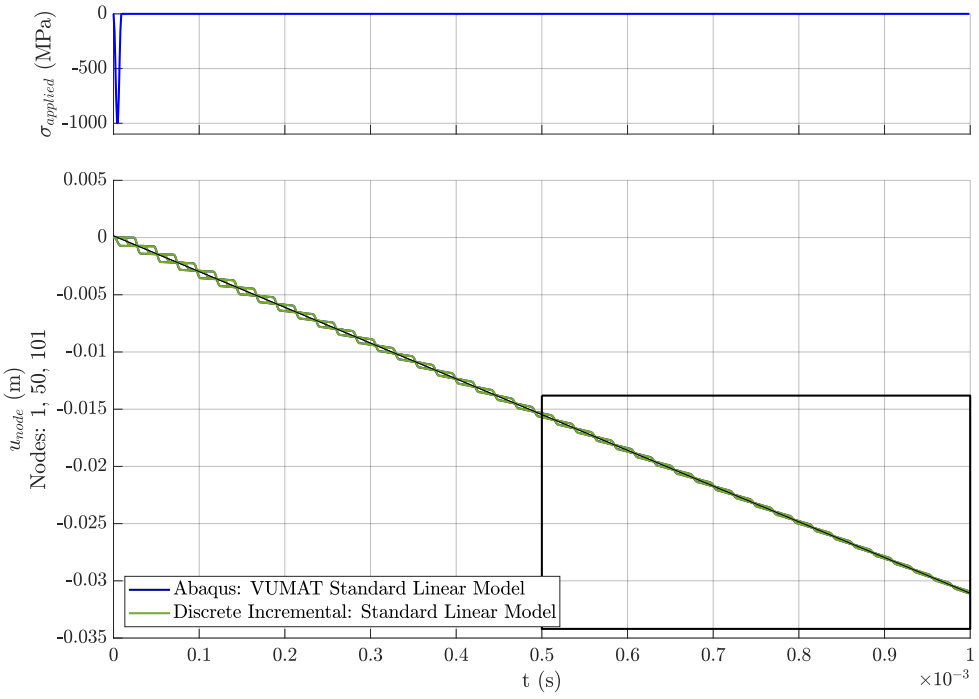


Figure 11.9: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing parameter $= 6 \times 10^{-4}$.

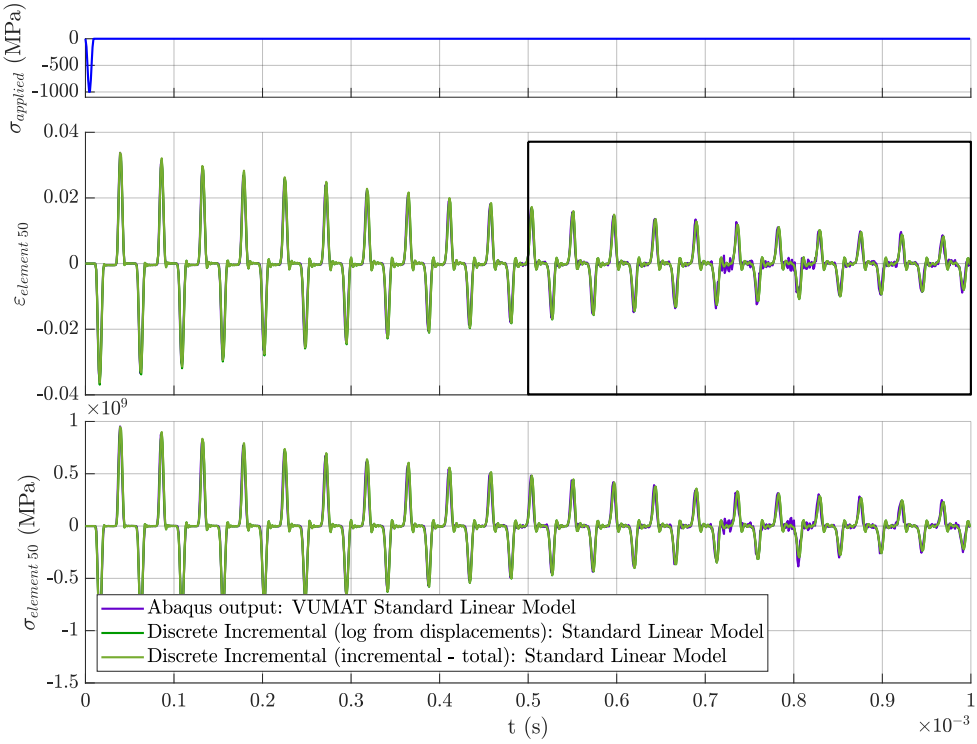
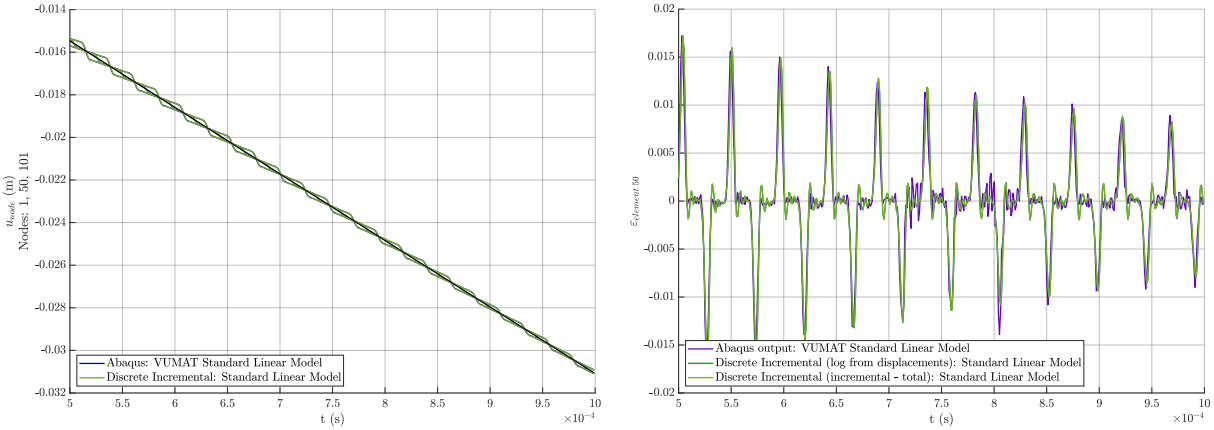


Figure 11.10: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing parameter $= 6 \times 10^{-4}$.

An enlarged portion of the displacement and strain graphs is shown in Figure 11.11. Figure 11.11a shows the displacement of the hereditary integral solution compared to the displacements outputted by the Abaqus simulation. These match each other as expected. The strain graph is shown in Figure 11.11b where the hereditary integral solution and the Abaqus result track closely. There is no strain drift, as was seen in some of the linear elastic results, and the amplitude of the strain wave is damped by the same amount.



(a) Abaqus VUMAT Standard Linear Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Displacement-Time graph).

(b) Abaqus VUMAT Standard Linear Model - Hourglassing control parameter $Q_2 = 6 \times 10^{-4}$ (Strain-Time graph).

Figure 11.11: Linear Elastic Model - LS Dyna Built-In vs UMAT vs one-dimensional spring-mass system incremental time-stepping scheme.

11.5 Abaqus vs Experimental Data

Based on the findings of Cloete *et al.* [3] and the investigation into the one-dimensional non-linear Maxwell in Chapter 6.3, values of $p > 1$ will give us the gradient of the ISR regime transition slope. For this reason we will focus on using functions with $p > 1$. Given that a value of $p < 1$ will affect the stress - strain rate curves at low strain rates, and the scarcity of data points in this region, functions for $p < 1$ will not be considered. The final simulations were also run only for strain rates in the ISR regime ($1s^{-1}$ to $120s^{-1}$).

After discussing the various ways of implementing a non-linear Maxwell element in one dimension, the use of a dashpot coefficient function was chosen for the implementation in the cortical bone model. There are several different implementations of the non-linear dashpot coefficient function in three dimensions. Three of these implementations were outlined in Chapter 10.1.4, namely as a function of the strain rate in each direction, of the dominant strain rate and of the equivalent strain rate. These implementations will be investigated in this section.

11.5.1 Simulation Parameters

For each simulation at a strain rate, the total run time was adjusted such that the total strain reached by the simulation was the same. The sampling time was also adjusted such that there are one thousand output points. The mass scaling factor was also chosen for slower strain rates to enable the simulation to run efficiently. All these parameters are outlined in Table C.1 in Appendix C.5.2 for the various strain rates that were used.

11.5.2 Simplified Material Model for Cortical Bone

A simplified material model for cortical bone is proposed which consists of a single non-linear Maxwell element and a spring in parallel, see Figure 11.12, which is a standard linear model with a non-linear dashpot. The parameters used in the simplified model are outlined in Table 11.1 in comparison to the parameters used in the material model proposed in Chapter 9.

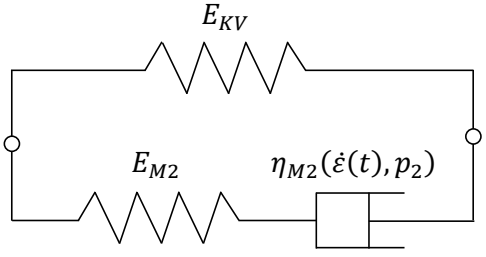


Figure 11.12: Simplified material model for cortical bone.

Table 11.1: Parameters of the cortical bone material model compared to the simplified version.

		Simplified Material Model	Proposed Material Model
E_{KV}	(MPa)	9500	9000
$\dot{\varepsilon}_{ref\ E_{KV}}$	(s^{-1})	1	1
η_{KV}	$(MPa.s)$	-	0.2
$\dot{\varepsilon}_{ref\ \eta_{KV}}$	(s^{-1})	-	1
p		-	0.7
E_{M1}	(MPa)	-	1000
η_{M1}	$(MPa.s)$	-	40
p_1		-	0.5
$\dot{\varepsilon}_{ref\ peg\ M1}$	(s^{-1})	-	1
$p_{1,peg\ M1}$		-	0.5
$\dot{\varepsilon}_{tuning\ M1}$	(s^{-1})	-	6.5
E_{M2}	(MPa)	6400	5500
η_{M2}	$(MPa.s)$	0.1	0.1
p_2		2.5	3
$\dot{\varepsilon}_{ref\ peg\ M2}$	(s^{-1})	0.35	0.85
$p_{2,peg\ M2}$		2.5	3
$\dot{\varepsilon}_{tuning\ M2}$	(s^{-1})	6.5	6.5

The stress - strain rate curves were plotted for this simplified model and compared to the experimental data and theoretical implementation by Cloete *et al.* [3], shown in Figure 11.13.

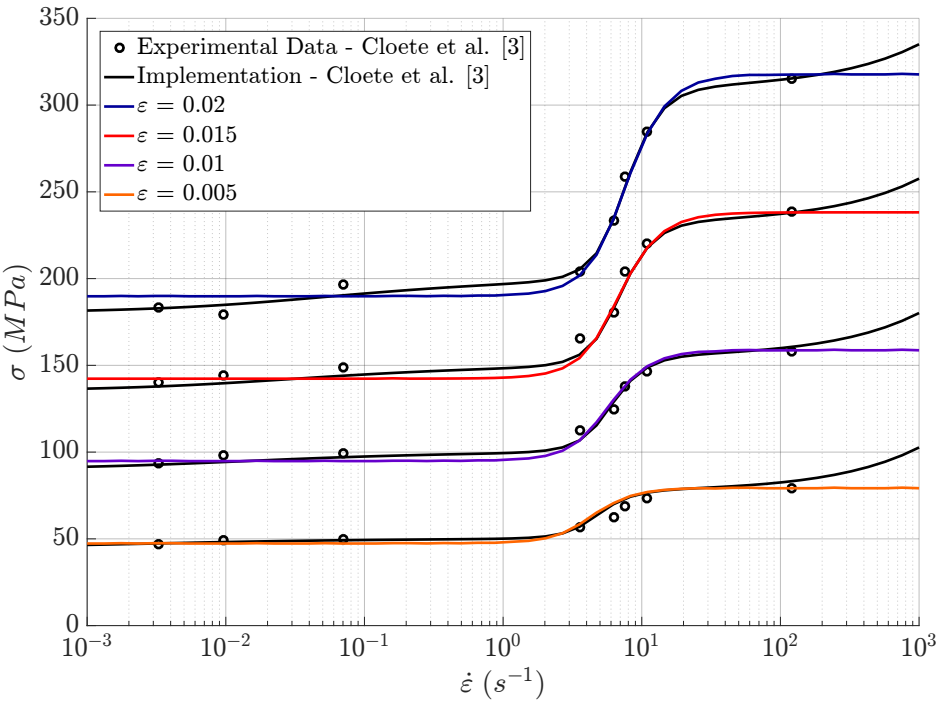


Figure 11.13: Cortical Bone Model - Baseline - $p = 2.5$ - Stress - Strain Rate graph.

11.5.3 Dashpot Coefficient Functions

Varying values of p and $\dot{\varepsilon}_{ref}$ were investigated for the dashpot coefficient function $\eta(\dot{\varepsilon})$. Each scenario was plotted in Figure 11.14.

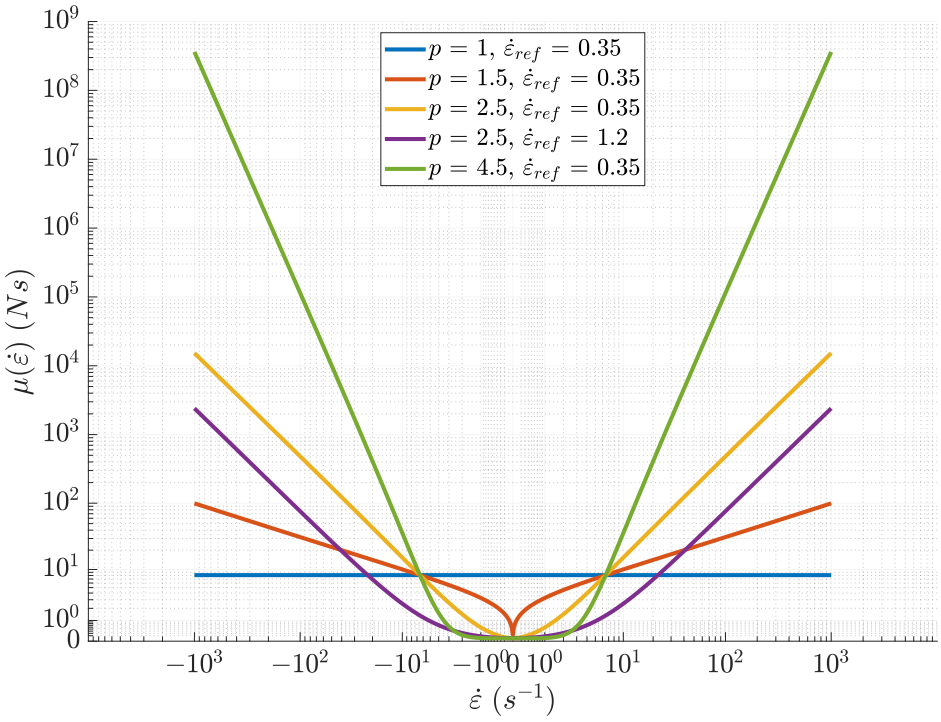


Figure 11.14: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon})$ - Dashpot coefficient functions.

A linear dashpot with $p = 1$ shows a flat line which means that the viscosity of the dashpot is constant for all strain rates. This is expected for a linear dashpot which could also be represented by a constant value. As the value of p increases, the gradient of the dashpot coefficient function on either side of zero increases and the range of strain rates around zero over which the function is minimum increases. For a value of $p = 2.5$ with different values of $\dot{\varepsilon}_{ref}$, the higher the $\dot{\varepsilon}_{ref}$, the less steep the curve.

11.5.4 Cortical Bone Model - Strain Rate in Each Direction

The simplified cortical bone model was first implemented using a dashpot coefficient as a function of the strain rate in each direction, $\eta(\dot{\varepsilon}_{ij})$. The code used to implement this user-defined material model can be found in Appendix B.3.

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

The specimens used to obtain the experimental results by Cloete *et al.* [3] were cylindrical with a nominal diameter of 0.004m and a length of 0.005m. First, a single 8 node brick element mesh was used with a cross-sectional area of $1.2566 \times 10^{-5} m^2$, which is the equivalent of a circular profile of diameter $0.004m$, and a length of $0.005m$. The element was constrained to be fixed at one end and the other end was loaded with a constant strain rate, see Figure 11.15. This simulation was carried out for various strain rates, making up a single set of stress - strain and stress - strain rate curves.

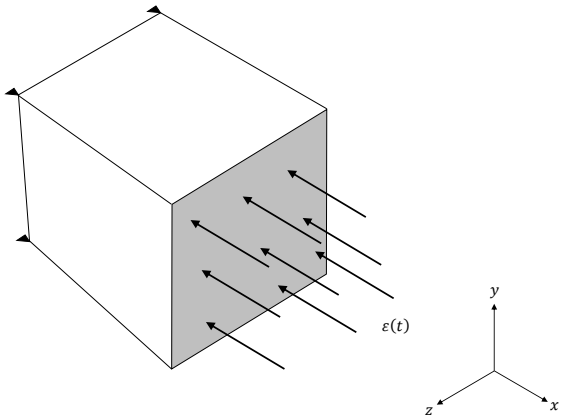
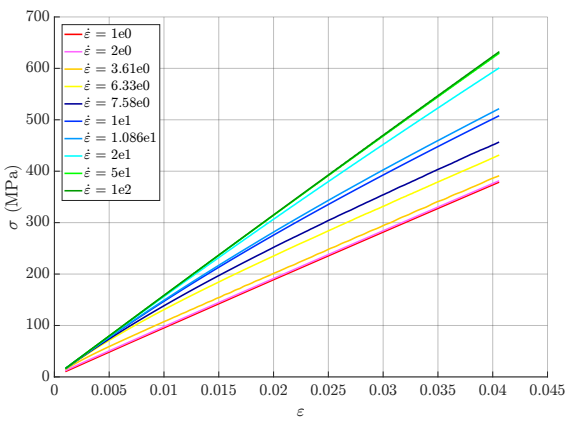
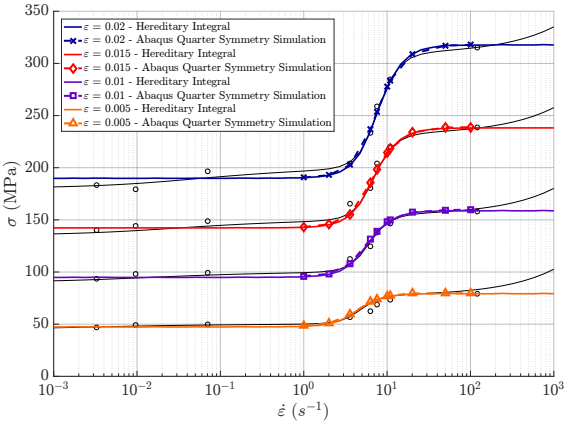


Figure 11.15: Single element mesh is constrained along the lines of symmetry and constrained at one end not to move x-direction and an applied stress $\sigma(t)$ in the negative x-direction.

The value of p used was 2.5 and the reference strain rate was 0.35. The stress - strain curves for each strain rate simulation were plotted, shown in Figure 11.16a. The stress - strain rate curves sampled at four strains were plotted in Figure 11.16b and compared to the one-dimensional hereditary integral as well as the experimental results by Cloete *et al.* [3].



(a) Abaqus Stress - Strain graph.



(b) Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph.

Figure 11.16: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{ij})$ - $p = 2.5$.

The stress - strain curves show that the modulus of the material changes with varying strain rates. The stress - strain rate curves show exactly how the stress in the element changes as the strain rate changes for varying strains. Figure 11.16b shows that, for a single element, the Abaqus simulation matches the one-dimensional hereditary integral.

Quarter Symmetry ($\dot{\varepsilon}_{ref} = 0.35$)

The same domain size of diameter $0.004m$ and length of $0.005m$ was then meshed in a quarter symmetry representation of a cylindrical specimen using 8 noded bricks. Figure 11.17 shows the mesh which has 91 elements through the cross-section and 25 elements along its length as this should provide relatively accurate results at acceptable computational expense. It is constrained along the lines of symmetry such that it can only move in the x-direction. One end is fixed not to move in the x-direction and a compressive constant strain rate is applied to the other face.

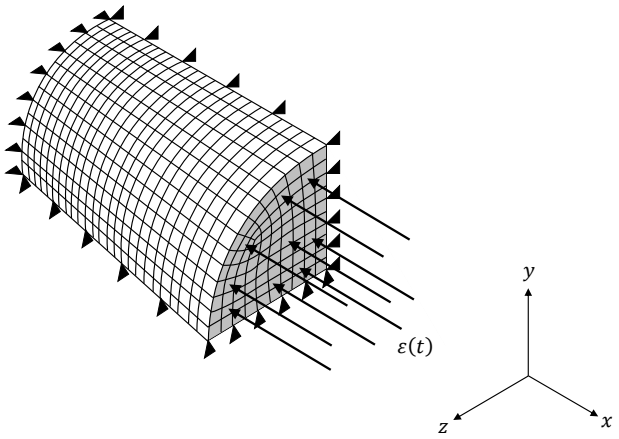
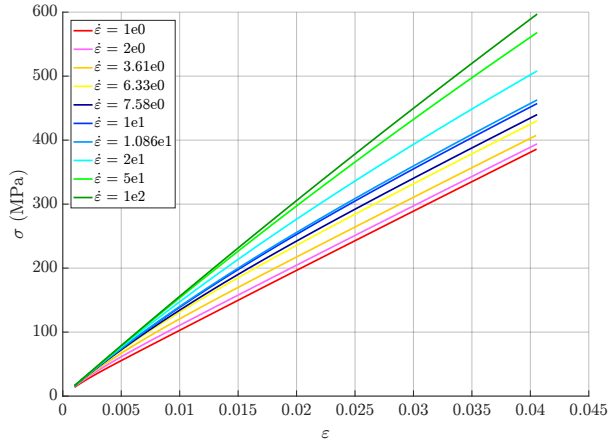
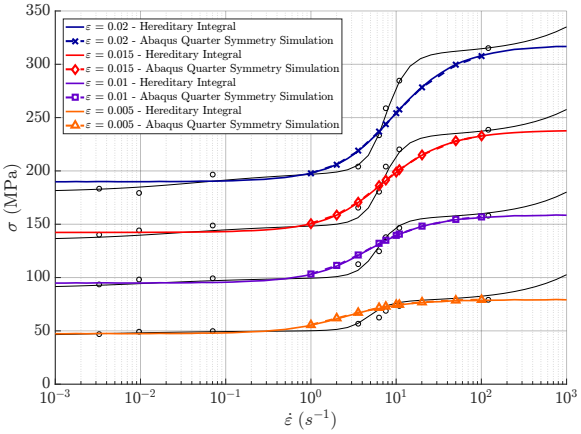


Figure 11.17: Quarter symmetry cylindrical specimen (91 elements on the face and 25 elements along the length). Constrained along lines of symmetry and constrained at one end not to move x-direction, an applied strain $\varepsilon(t)$ in the negative x-direction.

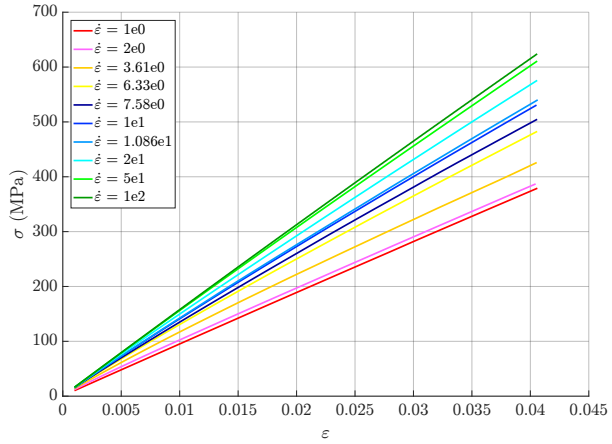
The function as a strain rate in each direction was implemented in this quarter symmetry simulation with varying values of p and the stress - strain and stress - strain rate graphs were plotted for each scenario in Figure 11.18. Firstly, a value of p equal to 1 was investigated, with a reference strain rate of 0.35. The stress - strain curves in Figure 11.18a show that the modulus of the material changes at varying strain rates. The stress - strain rate curves, sampled at four strains, were plotted in Figure 11.18b and compared to the one-dimensional hereditary integral as well as the experimental results by Cloete *et al.* [3]. The one-dimensional hereditary integral and the Abaqus simulation results for $p = 1$ matched each other, however, the experimental results exhibit a steeper gradient in the ISR regime.



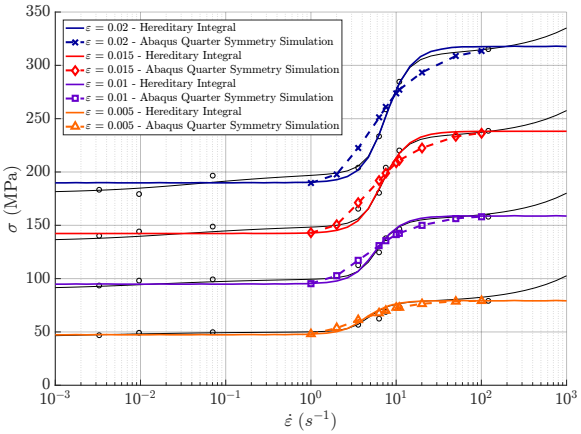
(a) Abaqus Stress - Strain graph - $p = 1$.



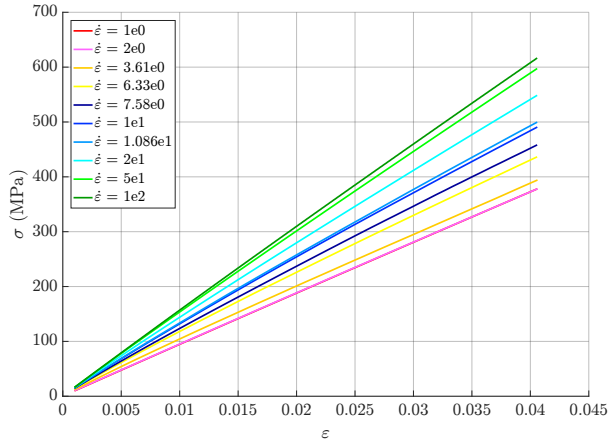
(b) Experimental vs Abaqus vs multi element hereditary integral - $p = 1$ - Stress - Strain Rate graph.



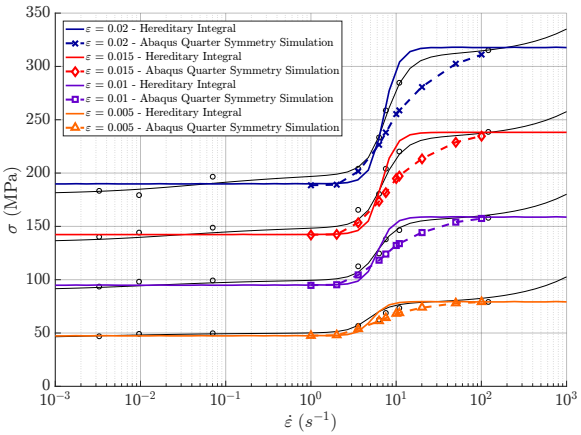
(c) Abaqus Stress - Strain graph - $p = 2.5$.



(d) Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph.



(e) Abaqus Stress - Strain graph - $p = 4.5$.



(f) Experimental vs Abaqus vs multi element hereditary integral - $p = 4.5$ - Stress - Strain Rate graph.

Figure 11.18: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$.

The value of p was increased to 2.5 while keeping the reference strain rate at 0.35. The stress - strain curves for each strain rate simulation were plotted in Figure 11.18c. The stress - strain rate curves in Figure 11.18d show that for $p = 2.5$ the one-dimensional hereditary integral and the experimental data match in the ISR regime and that the simulation has deviated from the desired result. The value of p was then increased further to 4.5 while keeping the reference strain rate at 0.35. This showed a slightly sharper uptake at the beginning of the ISR regime, however, the rest of the curve looks similar to the previous results.

These stress - strain rate curves were compared for the values of $p = 1$, 2.5 and 4.5 for the implementation of the strain rate in each direction, shown in Figure 11.19. These were also compared to the one-dimensional hereditary integral for a value of $p = 2.5$ and the experimental results by Cloete *et al.* [3]. The curve for a value of $p = 4.5$ has the steepest uptake at the start of the ISR regime, however, it does not match the gradient of the ISR regime entirely.

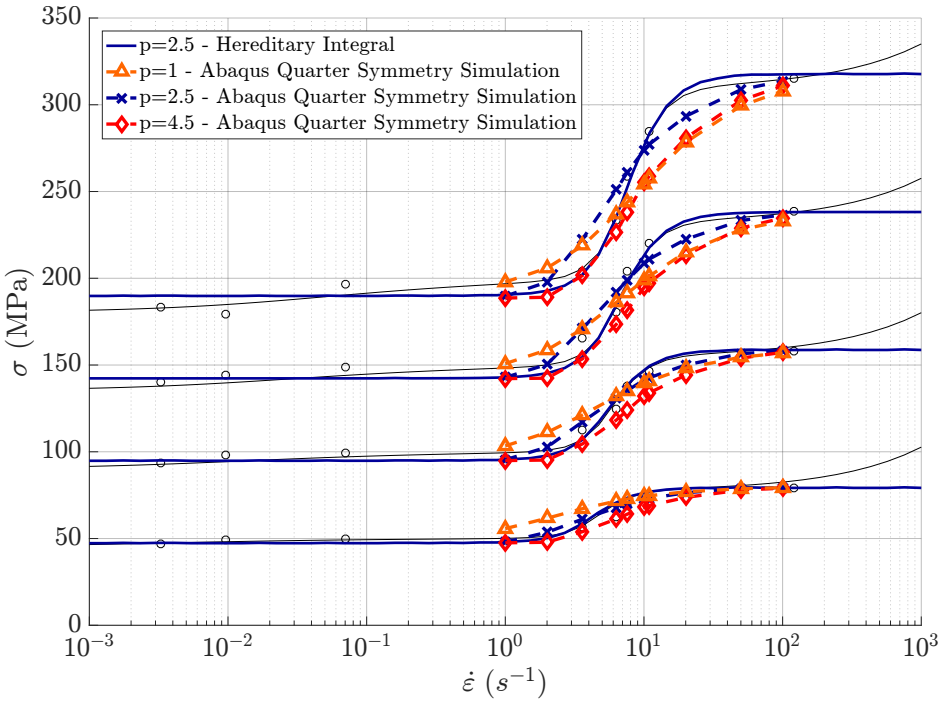
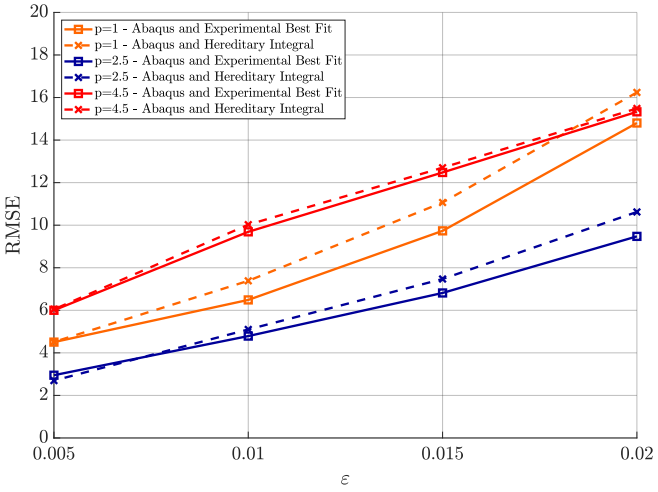
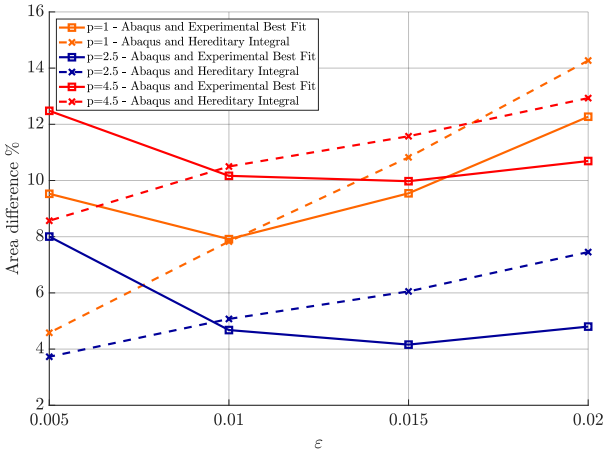


Figure 11.19: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 1.1$ vs 2.5 vs 4.5 - Stress - Strain Rate graph.

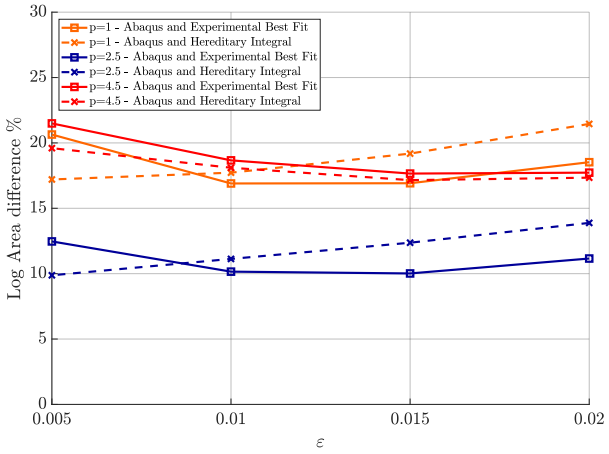
An error analysis between the simulations in Abaqus and a reference curve (either the best fit curve through the experimental results or the hereditary integral for $p = 2.5$) was carried out. There were two different methods of error analysis. The first of which was the root mean square error which was calculated for each strain curve and each value of p and shown in Figure 11.20a.



(a) Root mean square.



(b) Area difference.



(c) Log Area difference.

Figure 11.20: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Error Analysis - Abaqus vs Experimental Best Fit - Abaqus vs Hereditary Integral - $p = 1.1$ vs 2.5 vs 4.5.

The root mean square calculation was carried out for the portion of the curve between strain rates of 1s^{-1} and 100s^{-1} as follows

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}(\sigma_i - \hat{\sigma}_i)^2}{N}} \tag{11.1}$$

where N is the number of points and is equal to 10, σ_i is the i^{th} point of the Abaqus simulation curve for the strain rates specified above and $\hat{\sigma}_i$ is the i^{th} point of the reference simulation curve for the strain rates specified above.

The second error measure compares the area between each Abaqus curve and the reference curve, referred to as the area difference percentage and is shown in Figure 11.20b. This was calculated as the absolute difference in areas between the two curves as a percentage of the normalised area

under the reference curve

$$\text{Area difference } \% = \frac{\int_1^{100} |\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}| d\dot{\boldsymbol{\varepsilon}}}{\int_1^{100} \hat{\boldsymbol{\sigma}}_{norm} d\dot{\boldsymbol{\varepsilon}}} \times 100 \tag{11.2}$$

where $\dot{\boldsymbol{\varepsilon}}$ is the strain rates at chosen points, $\boldsymbol{\sigma}$ is the Abaqus simulation curve, $\hat{\boldsymbol{\sigma}}$ is the reference curve and $\hat{\boldsymbol{\sigma}}_{norm}$ is the normalised reference curve calculated as

$$\hat{\boldsymbol{\sigma}}_{norm} = \hat{\boldsymbol{\sigma}} - \min(\min(\boldsymbol{\sigma}), \min(\hat{\boldsymbol{\sigma}})). \tag{11.3}$$

The area used in equation (11.2) was calculated with respect to the linear strain rate, however, due to the logarithmic scale of the x-axis, the difference in area between the curves at the higher strain rates will contribute more to the error calculated. For this reason, the area was also calculated with respect to the log of the strain rate, shown in Figure 11.20c, in order to better represent the difference in area observed in the graphs, and is referred to as the logarithmic area difference percentage as follows

$$\text{Log Area difference } \% = \frac{\int_1^{100} |\boldsymbol{\sigma} - \hat{\boldsymbol{\sigma}}| dx}{\int_1^{100} \hat{\boldsymbol{\sigma}}_{norm} dx} \times 100 \tag{11.4}$$

where $x = \log \dot{\boldsymbol{\varepsilon}}$, $\boldsymbol{\sigma}$ is the Abaqus simulation curve, $\hat{\boldsymbol{\sigma}}$ is the reference curve and $\hat{\boldsymbol{\sigma}}_{norm}$ is the normalised reference curve shown in equation (11.3).

11.5.5 Cortical Bone Model - Dominant Strain Rate

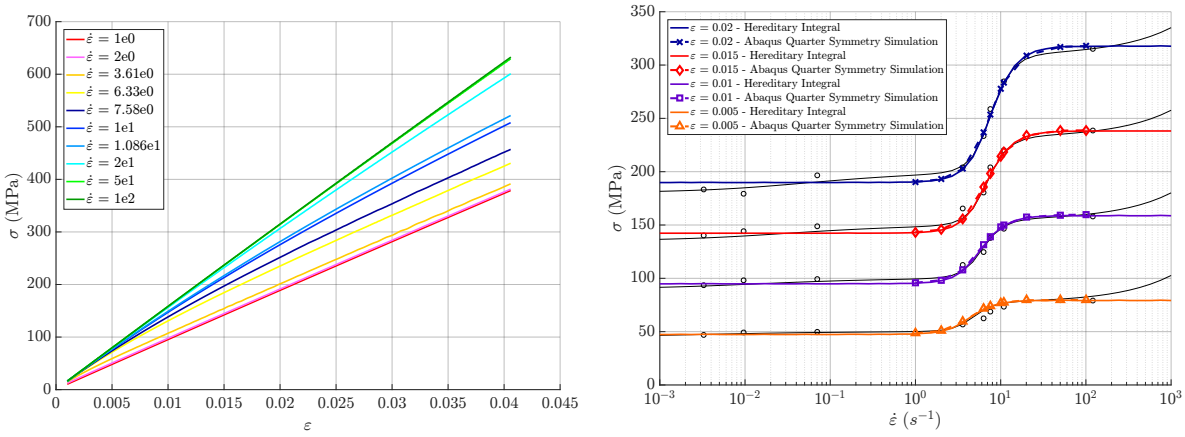
The simplified cortical bone model was then implemented using a dashpot coefficient as a function of the dominant strain rate, $\eta(\dot{\varepsilon}_{11})$. The code used to implement this user-defined material model can be found in Appendix B.4.

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

The same single element mesh, described previously, was used to investigate a dashpot coefficient as a function of the dominant strain rate $\eta(\dot{\varepsilon}_{11})$, which is the x-direction, as shown in Figure 11.15. The element was again constrained to be fixed at one end and the other end was loaded with a

constant strain rate curve. The value of p was 2.5 and the reference strain rate was 0.35.

The simulation was run again for the various strain rates, and the stress - strain curves for each strain rate simulation were plotted in Figure 11.21a. For the varying strain rates the modulus of the material changes. The stress - strain rate curves sampled at four strains were plotted and compared to the one-dimensional hereditary integral as well as the experimental results by Cloete *et al.* [3], as shown in Figure 11.21b. The Abaqus simulation results match the one-dimensional hereditary integral results.



(a) Abaqus Stress - Strain graph.

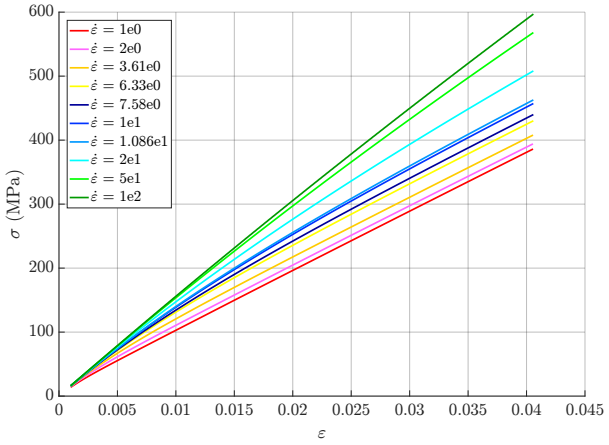
(b) Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph.

Figure 11.21: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{11})$ - $p = 2.5$.

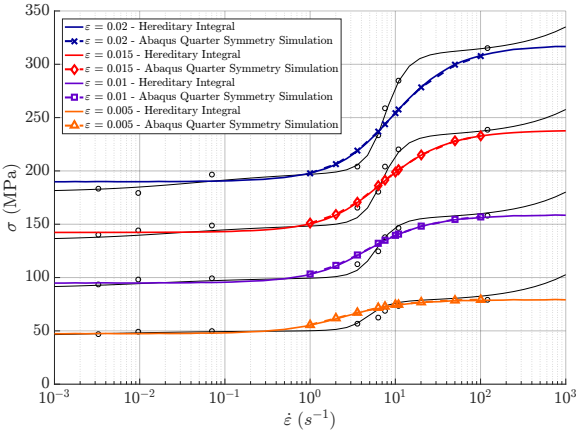
Quarter Symmetry ($\dot{\varepsilon}_{ref} = 0.35$)

The dominant strain rate function was then implemented using the same quarter symmetry simulation outlined in Figure 11.17. The mesh and simulation parameters used were the same as previously described and various simulation sets were run with varying values of p.

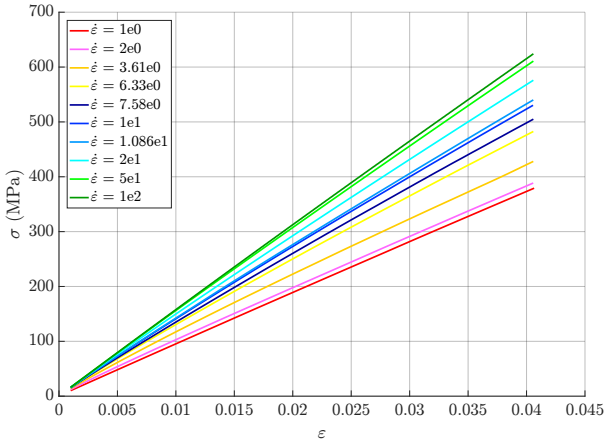
The first scenario was run with a value of p equal to 1 and a reference strain rate of 0.35 with the stress - strain curves shown in Figure 11.22a and the stress - strain rate curves shown in Figure 11.22b. This shows the same results as the strain rate in each direction for $p = 1$. The one-dimensional hereditary integral and simulation results match each other but not the experimental results. Figures 11.22c and 11.22d show the stress - strain and stress - strain rate curves for $p = 2.5$ where again, the hereditary integral and the experimental results are aligned, however, the simulation results do not exhibit a steep enough gradient in the ISR regime. The value of p was increased further to 4.5, shown in Figures 11.22e and 11.22f, which show that the gradient of the stress - strain rate curves for the simulation results do not match the desired experimental results.



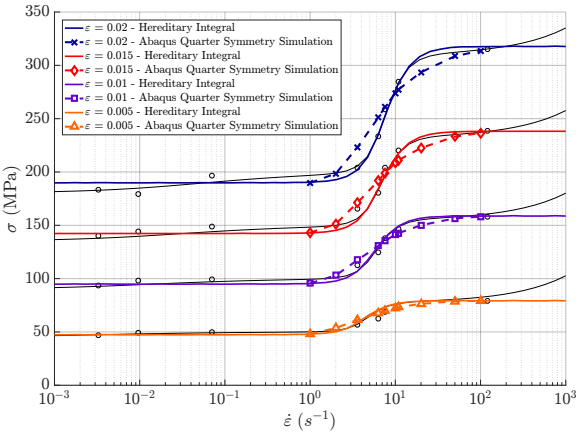
(a) Abaqus Stress - Strain graph - $p = 1$.



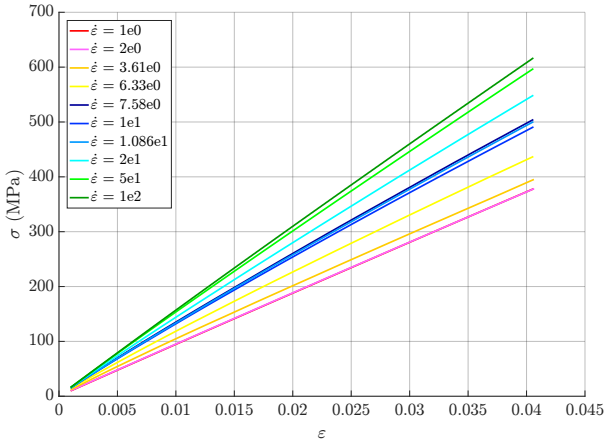
(b) Experimental vs Abaqus vs multi element hereditary integral - $p = 1$ - Stress - Strain Rate graph.



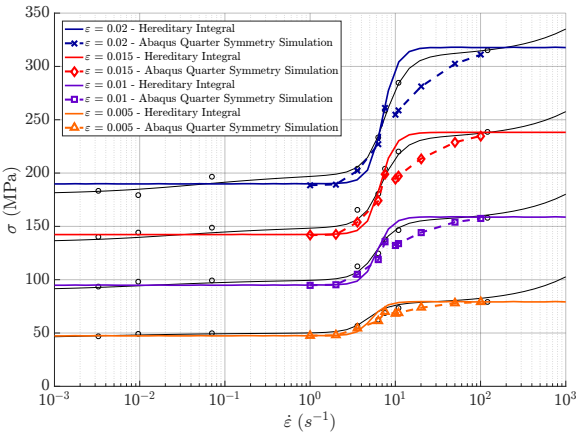
(c) Abaqus Stress - Strain graph - $p = 2.5$.



(d) Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph.



(e) Abaqus Stress - Strain graph - $p = 4.5$.



(f) Experimental vs Abaqus vs multi element hereditary integral - $p = 4.5$ - Stress - Strain Rate graph.

Figure 11.22: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$.

The stress - strain rate curves were compared for the values of p equal to 1, 2.5 and 4.5 for the implementation of the dominant strain rate, shown in Figure 11.23. These were also compared to the one-dimensional hereditary integral for a value of p equal to 2.5 and the experimental results by Cloete *et al.* [3]. The set of simulations that used a value of $p = 4.5$, again have the steepest uptake, however, the results do not match those of the experimental results.

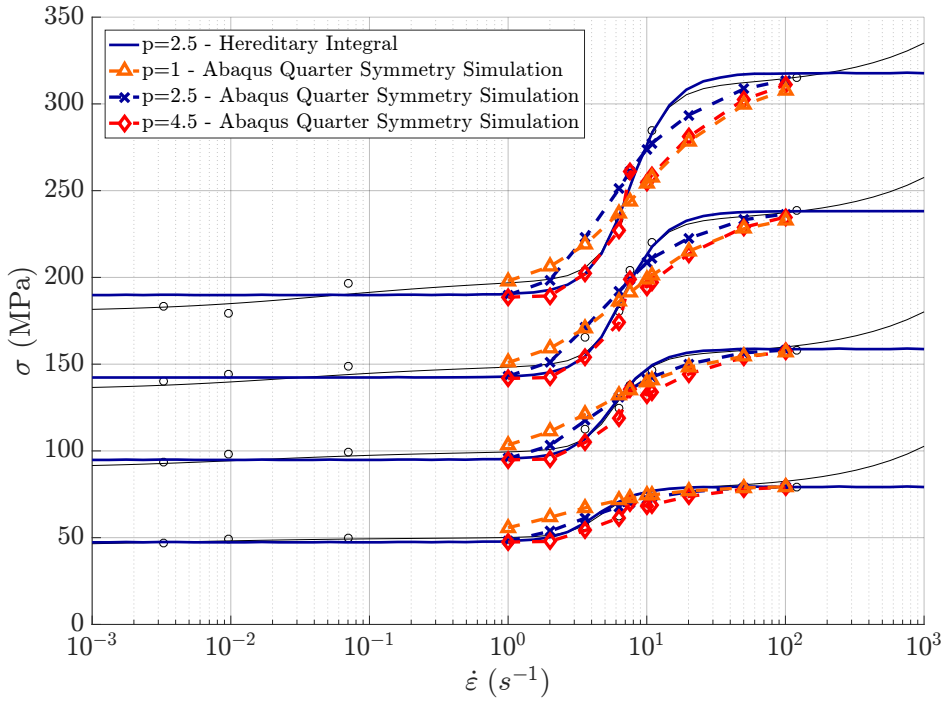
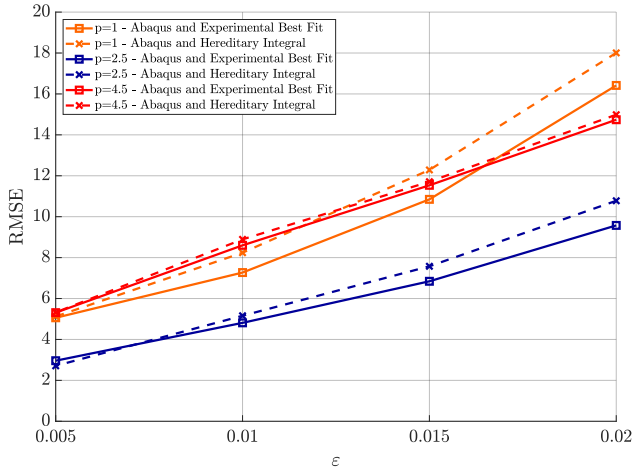
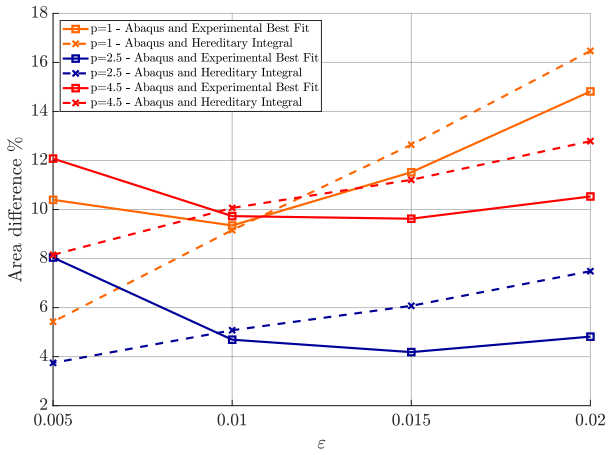


Figure 11.23: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ vs 4.5 - Stress - Strain Rate graph.

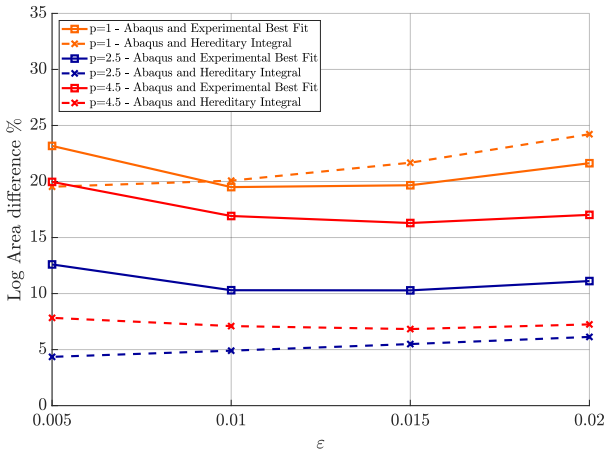
An error analysis between the simulations in Abaqus and a reference curve (either the best fit curve through the experimental results or the hereditary integral for $p = 2.5$) was carried out using the same methods as in section 11.5.4. These are shown in Figure 11.24.



(a) Root mean square.



(b) Area difference.



(c) Log Area difference.

Figure 11.24: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Error Analysis - Abaqus vs Experimental Best Fit - Abaqus vs Hereditary Integral - $p = 1$ vs 2.5 vs 4.5.

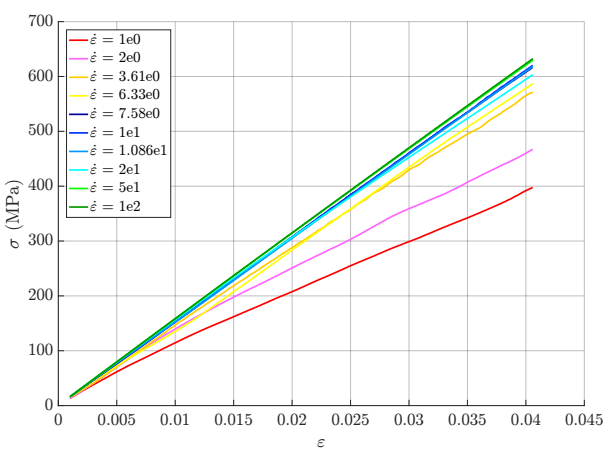
11.5.6 Cortical Bone Model - Equivalent Strain Rate

The simplified cortical bone model was then implemented using a dashpot coefficient as a function of the equivalent strain rate, $\eta(\dot{\varepsilon}_{eq})$. The code used to implement this user-defined material model can be found in Appendix B.5.

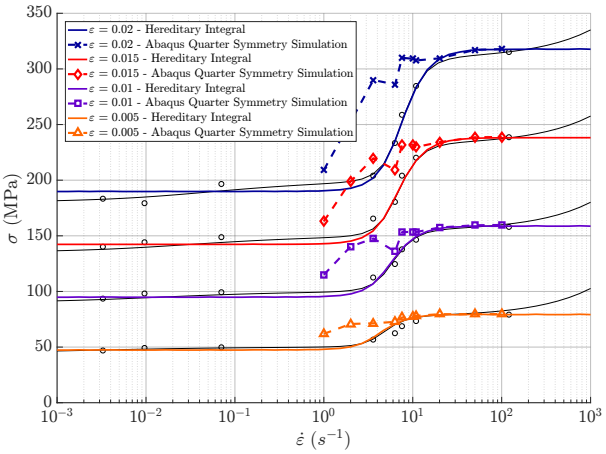
Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$ and 1.2)

The same single element simulation was carried out for a dashpot coefficient as a function of an equivalent strain rate. The mesh was again constrained to be fixed at one end and the other end was loaded with a constant strain rate. The value of p was 2.5 and the reference strain rate was 0.35 and 1.2.

The stress - strain curves for each strain rate simulation were plotted in Figures 11.25a and 11.26a. For a reference strain rate of 0.35 the stress - strain curves are concentrated at a higher modulus for the majority of the strain rate simulations. Increasing the reference strain rate to 1.2 shows that the stress - strain curves are more evenly spread out across the various strain rates. This is shown by the stress - strain rates curves shifting to the right with an increase in the reference strain rate, shown in Figures 11.25b and 11.26b.

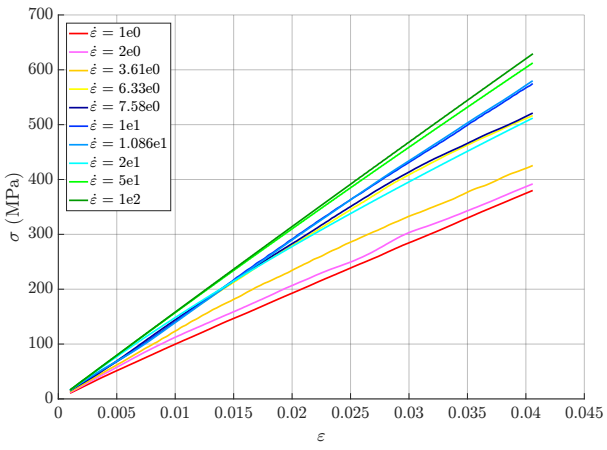


(a) Abaqus Stress - Strain graph.

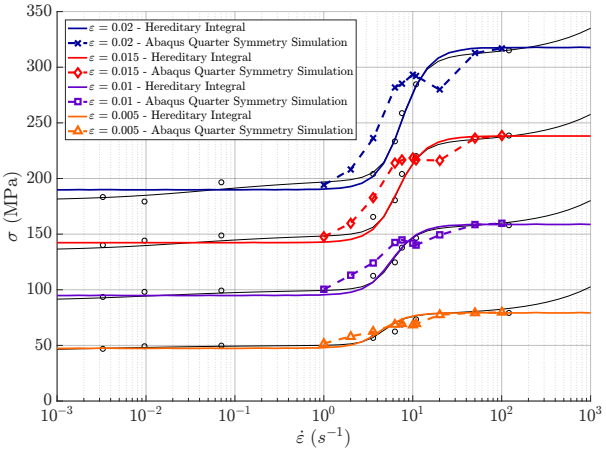


(b) Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph.

Figure 11.25: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - $\dot{\varepsilon}_{ref} = 0.35$.



(a) Abaqus Stress - Strain graph.

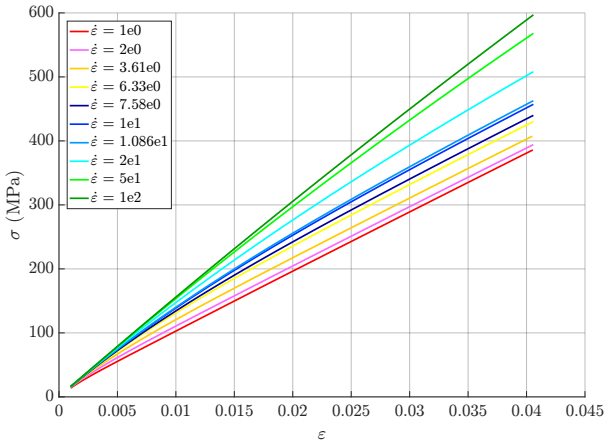


(b) Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph.

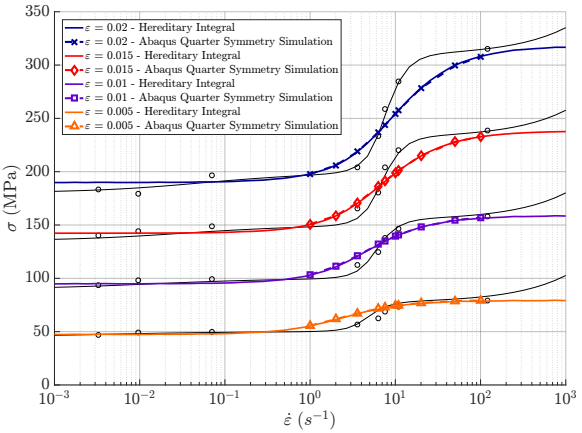
Figure 11.26: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - $\dot{\varepsilon}_{ref} = 1.2$.

The single element simulation using the equivalent strain rate does not align with either the one-dimensional hereditary integral or the experimental results. Although the gradient of the stress - strain rate curves is steep enough in the ISR regime, there is also an introduction of behaviour that resembles numerical instability but requires further investigation.

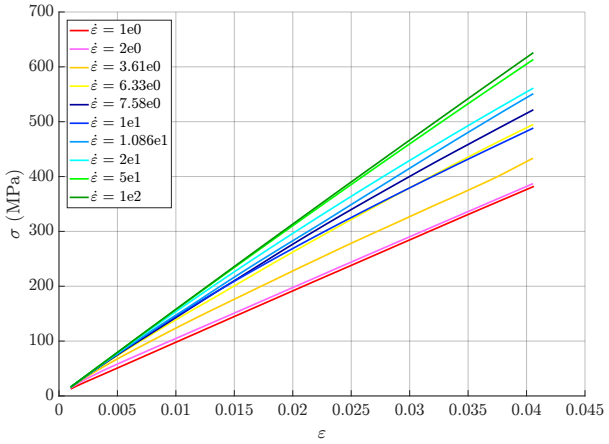
Quarter Symmetry ($\dot{\varepsilon}_{ref} = 0.35$)



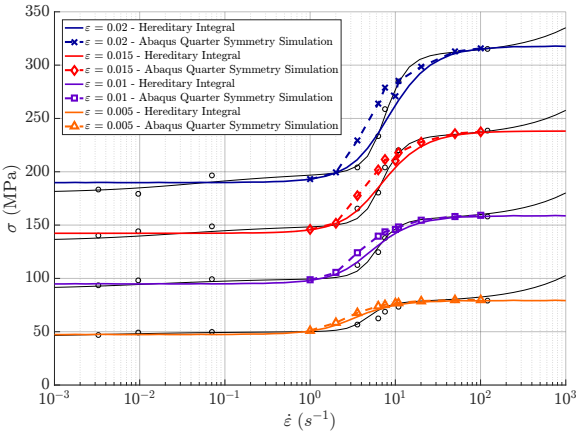
(a) Abaqus Stress - Strain graph - $p = 1$.



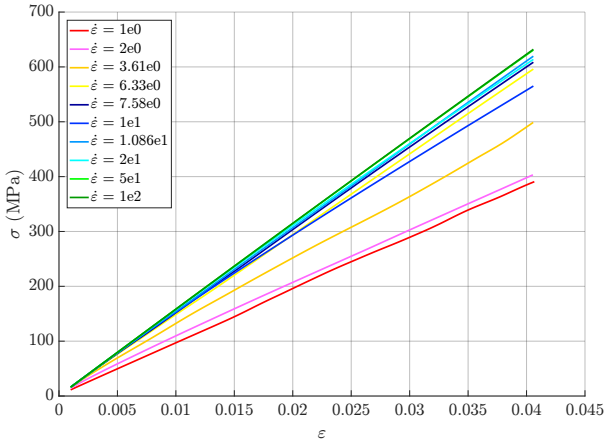
(b) Experimental vs Abaqus vs multi element hereditary integral - $p = 1$ - Stress - Strain Rate graph.



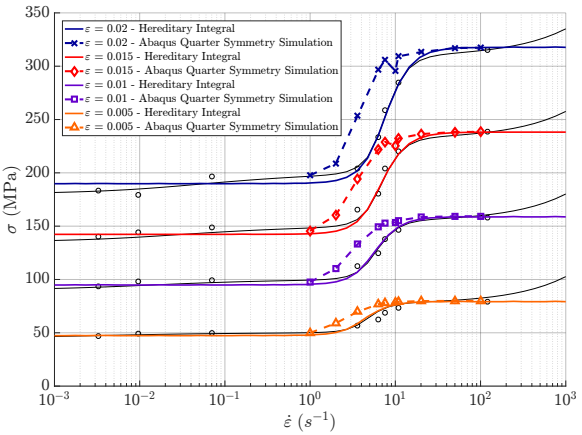
(c) Abaqus Stress - Strain graph - $p = 1.5$.



(d) Experimental vs Abaqus vs multi element hereditary integral - $p = 1.5$ - Stress - Strain Rate graph.



(e) Abaqus Stress - Strain graph - $p = 2.5$.



(f) Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph.

Figure 11.27: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $\dot{\varepsilon}_{ref} = 0.35$.

The equivalent strain rate was then implemented in a quarter symmetry simulation with the same mesh and parameters as shown in Figure 11.17, with a reference strain rate of 0.35. An initial set of simulations with $p = 1$ was run and the stress - strain and stress - strain rate curves plotted in Figures 11.27a and 11.27b. These show that the simulation results align with the one-dimensional hereditary integral but not with the experimental results.

After increasing the value of p to 1.5, while keeping the reference strain rate at 0.35, the stress - strain curves for each strain rate simulation were plotted in Figure 11.27c. The stress - strain rate curves in Figure 11.27d show that the gradient in the ISR regime matches the one-dimensional hereditary integral but not the experimental results, and there is an introduction of some anomalies.

The value of p was then increased further to 2.5 and the stress - strain curves are plotted in Figure 11.27e. The stress - strain rate curves were sampled at four strains, plotted and compared to the one-dimensional hereditary integral as well as the experimental results by Cloete *et al.* [3], shown in Figure 11.27f. This shows that the one-dimensional hereditary integral and experimental results are aligned. The simulation results show a similar gradient in the transition region, however, it is shifted to the left.

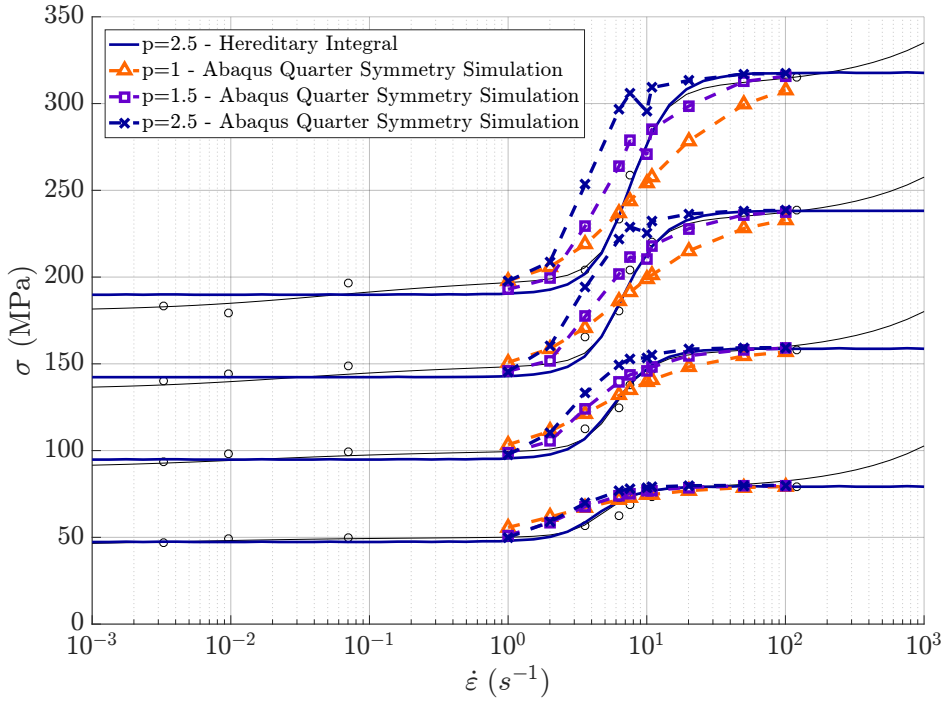


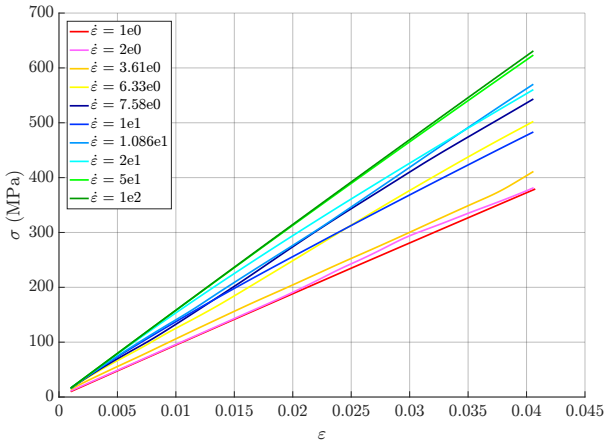
Figure 11.28: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1$ vs 1.5 vs 2.5 - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$.

For a reference strain rate of 0.35, the stress - strain rate curves were compared for a value of p equal to 1, 2.5 and 4.5 for the implementation of the equivalent strain rate. These were also compared to the one-dimensional hereditary integral for a value of p equal to 2.5 and the experimental results by Cloete *et al.* [3]. This shows that the gradient of the curves gets steeper with an increasing value

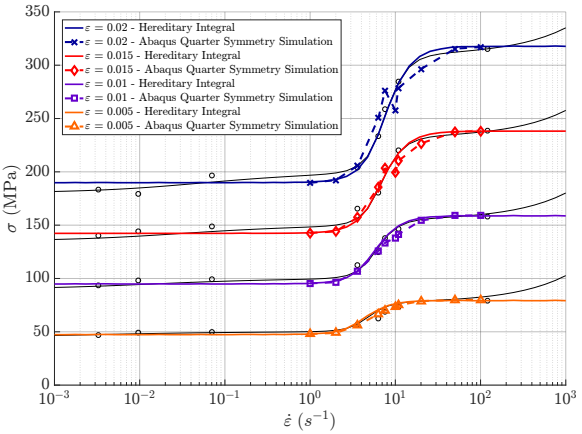
of p. The pick up point of the transition region remains the same and therefore the curves need to be shifted to the right as the value of p increases.

Quarter Symmetry ($p = 2.5$, $\dot{\varepsilon}_{ref} = 1.2$)

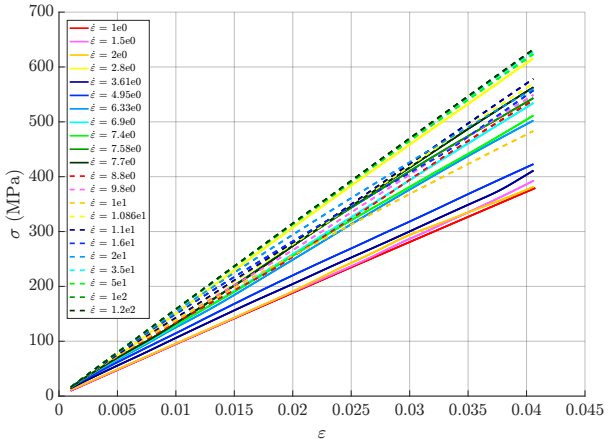
For a value of p equal to 2.5 the gradient of the stress - strain rate curves in the transition region matched that of the experimental results. The curve needed to be shifted to the right and this was done by increasing the reference strain rate to a value of 1.2. The stress - strain and stress - strain rate curves are shown in Figures 11.29a and 11.29b. The increased reference strain rate shifted the curve to the right such that the gradient and position of the simulation results now align with both the one-dimensional hereditary integral and experimental results.



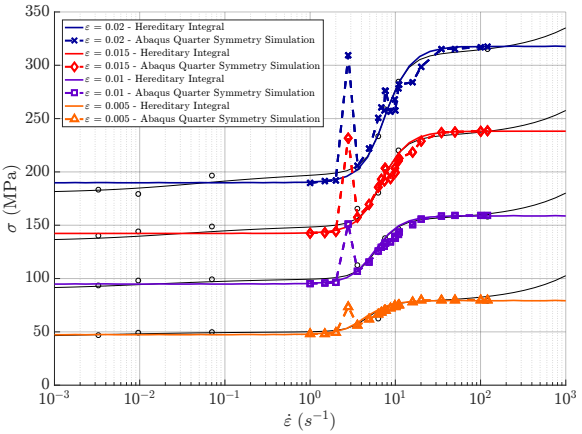
(a) Abaqus Stress - Strain graph - $p = 2.5$.



(b) Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph.



(c) Abaqus Stress - Strain graph - $p = 2.5$.

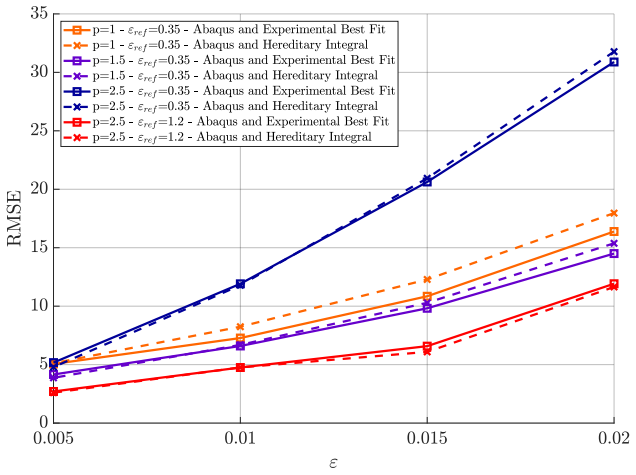


(d) Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph.

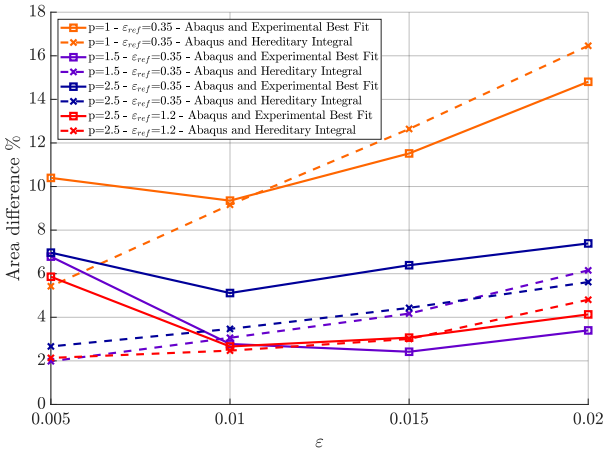
Figure 11.29: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $\dot{\varepsilon}_{ref} = 1.2$.

In order to fill in the curve, further simulations at strain rates between the existing points were carried out. The stress - strain curves for the full set of strain rates were plotted and shown in Figure 11.29c. The stress - strain rate curves are in Figure 11.29d show that there is some instability in the results at certain strain rates.

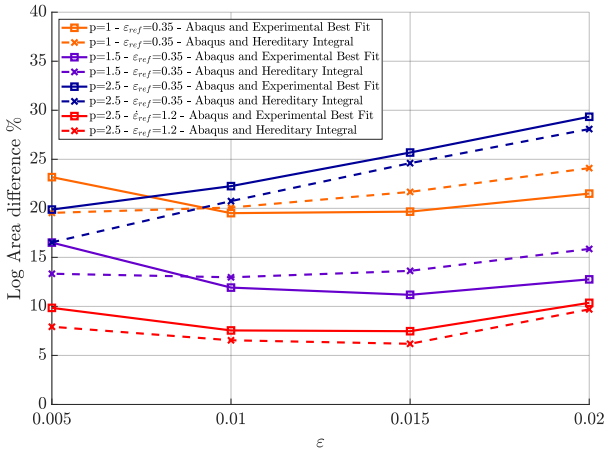
An error analysis between the simulations in Abaqus and a reference curve (either the best fit curve through the experimental results or the hereditary integral for $p = 2.5$) was carried out using the same methods as in section 11.5.4 and 11.5.5. This error analysis included the result of using a different reference strain rate to shift the curve to the right. These results are shown in Figure 11.30.



(a) Root mean square.



(b) Area difference.



(c) Log Area difference.

Figure 11.30: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - Error Analysis - Abaqus vs Experimental Best Fit - Abaqus vs Hereditary Integral - $p = 1$ vs 1.5 vs 2.5 ($\varepsilon_{ref} = 0.35$ vs 1.2).

Chapter 12

Discussion

This chapter discusses the LS Dyna and Abaqus results presented in Chapter 11. The LS Dyna linear elastic models were investigated using varying values of the hourglassing control parameter. These results were then compared to the same simulations performed in Abaqus. A user-defined linear Maxwell material model in Abaqus was validated against the one-dimensional hereditary integral, followed by the implementation of a non-linear Maxwell element in a cortical bone model using a dashpot coefficient function, which was investigated for different strain rates and values of the power p.

12.1 Validation of LS Dyna Simulations

The simulations carried out in LS Dyna were first validated for a linear elastic model against a multi-element, one-dimensional, spring-mass system, with a stress impulse applied to one end of a row of free, unconstrained elements. Initially, with the hourglassing control setting turned off in LS Dyna, the displacement of the end and middles nodes were mostly representative of the expected behaviour, with a slight drift from the one-dimensional solution over time. The displacement of the two end nodes of the row of one hundred elements oscillates between being over and under that of the one-dimensional solution, shown in Figure 11.4a. Figure 11.4b shows the strain response and a distinct strain drift over time that clarifies these discrepancies in the nodal displacements.

The simulation was run using both the built-in linear elastic material model in LS Dyna and a user-defined linear elastic model, and the amplitude of the stress and strain graphs decreased over time in both sets of results. This showed that there is a default damping being applied to the simulation. Investigation into LS Dyna's documentation in Chapter 10.3 revealed that a small amount of damping is applied in order to avoid hourglassing, the effects of which are apparent in these graphs.

Even with the hourglass control turned off damping was still being applied. The hourglass control was then adjusted by changing the parameter Q_2 to use the default value of 6×10^{-2}. This still showed the discrepancy in the displacements in Figure 11.4c, as well as a decreasing amplitude of the stress and strain graphs and the same strain drift in Figure 11.4d. The hourglassing control value Q_2 was decreased even further to 6×10^{-4} and the discrepancy in the displacements was still seen in Figure 11.4e. The amplitude of the stress and strain graphs was representative of the one-dimensional model and the behaviour that was expected. However, a strain drift in the strain graph remained in Figure 11.4f. These simulations used a single row of 8-node brick elements and the displacement of each edge was plotted. The displacements of each edge were the same and therefore the possibility of hourglassing being the cause of the strain drift was ruled out.

The strain showing the strain drift was calculated as nominal strain which is the change in length of the element divided by the original length. In Figure 11.4f the strain directly outputted by LS Dyna showed no strain drift. This lead to further investigation of the strain calculation and a different strain calculation was carried out using true strain, which is the incremental addition of strains over time, otherwise known as logarithmic strain. This calculation also shows a strain drift which means the discrepancies in the nodal displacements from LS Dyna affected the calculated strain results and that the strain outputted by LS Dyna is not calculated from the nodal displacements.

12.2 LS-Dyna vs Abaqus Simulations Validation

Inconsistencies in the strain response and nodal displacements of the LS Dyna simulations lead to additional investigation into a different software, Abaqus. The same stress impulse free bar simulations were run using the built-in and user-defined linear elastic models in Abaqus and were compared to the LS Dyna results and the one-dimensional spring-mass system solution.

The built-in material model shows the same discrepancy in the displacements for Abaqus, shown in Figure 11.7a, which matches the LS Dyna results. Using an hourglass control parameter value of $Q_2 = 6 \times 10^{-4}$ shows that the effect of damping is reduced in the Abaqus results similar to the LS Dyna results, see Figure 11.6. The strain drift is present in the built-in linear elastic model for Abaqus as it was with LS Dyna, see Figure 11.7b.

The user-defined linear elastic model in Abaqus, however, does not exhibit the discrepancy in displacements as was seen in the LS Dyna results in Figure 11.7c. The effect of damping on the stress and strain responses from Abaqus are not seen in Figure 11.7d. Only the user-defined model in LS Dyna shows the strain drift. The results from the Abaqus simulation do not exhibit the strain drift in the user-defined model, however, the strain drift was seen with the built-in model.

12.3 Abaqus Simulations Validation

In order to verify that Abaqus would not produce the same discrepancies in the displacement and strain results as LS Dyna, the results of the Abaqus built-in linear elastic model and the user-defined linear elastic model were compared. This showed that the displacement discrepancy was present in the built-in linear elastic model but not in the user-defined linear elastic model in Abaqus, shown in Figure 11.8a. The effect of damping in Abaqus was not present when using the hourglass control parameter of $Q_2 = 6 \times 10^{-4}$. The strain drift can also only be seen in the built-in model in Abaqus, Figure 11.8b, showing that the user defined linear elastic material model in Abaqus is the only simulation producing the expected results.

12.4 Hereditary Integral vs FEA Simulations

Since all results of the user-defined linear elastic model in Abaqus were shown to behave as expected, this was chosen as the finite element analysis software to use going forward. The hereditary integral implementation in Abaqus was tested using a standard linear model and the same simulation as before was carried out using a row of standard linear elements. The nodal displacements were representative as with the linear elastic tests, shown in Figure 11.11a, and there is no strain rate drift seen in the strain graph in Figure 11.11b. The amplitude of the stress and strain graphs are also aligned with the one-dimensional hereditary integral results. There are a few spikes of instability in the strain results, however the general behaviour is aligned with the one-dimensional results and the hereditary integral implemented of a linear Maxwell element in the user-defined model in Abaqus produced the expected results. The instability could be attributed to the size of the time step that is calculated and used by Abaqus.

12.5 Abaqus vs Experimental Data

In Section 10.1.4, the dashpot coefficient could either be a function of the strain rate in each direction, the dominant strain rate, or the equivalent strain rate. For each implementation a set of tests was carried out: a specimen was simulated using both a single element and quarter symmetry representation, using the same dimensions (cross sectional area and length) as those used to obtain the experimental data by Cloete *et al.* [3]. A constant strain rate displacement was applied to one end of the specimen in a compressive direction while the other end was fixed in the direction along the axis of the specimen, and the specimen was free to move perpendicular to the axis (unconstrained). The stress - strain response was recorded for each strain rate in the ISR regime. The stress at each strain rate was sampled at various strains (determined by the tests conducted by Cloete *et al.* [3]) and plotted on a stress - strain rate curve. Each set of tests was also compared

to the experimental (black circles) and simulated (solid black line) results from Cloete *et al.* [3].

The simplified version of the material model required some adjustment of the model parameters, as outlined in Table 11.1, and the results that will be used as a baseline are shown in Figure 11.13. This shows that the stress - strain rate graph passes through the middle of the low strain rate points, and through the high strain rate data point, as well as capturing the slope of the ISR regime. These parameters were also used in the user-defined models, with a varying value of the power p_2.

Each simulation was run at a different strain rate, carried out using parameters such that the specimen reached the same strain and the same number of data points was outputted. This involved adjusting the total run time of the simulation as well as the sampling time of the results. These parameters are outlined in Table C.1 where the mass scaling factor is also defined. The mass scaling factor was adjusted for the applied strain rate in order for the simulation to run more efficiently, especially for the lower strain rates which had to run for a longer time in order to achieve the same strain.

The same set of dashpot coefficient functions were investigated for each implementation. The values of p that were investigated were 1, 1.5, 2.5 and 4.5 all with a reference strain rate value of 0.35. The reference strain rate was only adjusted to 1.2 for when $p = 2.5$ in order to shift the curve along the x-axis after reasonable results were obtained for the gradient of the ISR regime. The dashpot coefficients were plotted in Figure 11.14 for each scenario.

For a value of $p = 1$ and $\dot{\varepsilon}_{ref} = 0.35$ the dashpot coefficient function plotted in Figure 11.14 shows that the value of the coefficient is a constant value for all strain rates. This is exactly what is expected as a linear Maxwell would simply have a constant dashpot coefficient value instead of a function. When the value of p is increased to 1.5, the gradient on either side of zero strain rate is slightly increased from the constant value of the linear solution. Increasing the value of p further to 2.5 shows that at small strain rates the coefficient will have a small effect on the response of the element, while at larger negative and positive strain rates the coefficient will have a larger effect on the response of the element. Changing the reference strain rate to 1.2 for $p = 2.5$ shows a dashpot coefficient that at intermediate strain rates will have a more gradual effect, as opposed to when the reference strain rate is equal to 0.35 where the transition between low and high strain rates was more severe. For a value of $p = 4.5$ and a reference strain rate of 0.35, the dashpot coefficient function in Figure 11.14, shows a much steeper gradient either side of $0s^{-1}$ with a much flatter region around zero strain rate.

12.5.1 Cortical Bone Model - Strain Rate in Each Direction

Using the strain rate in each direction, first a single element simulation was carried out using a value of p equal to 2.5 in order to assess if the three-dimensional hereditary implementation matched that of the analytical one-dimensional hereditary integral solution. The stress - strain curves in Figure 11.16a show expected linear behaviour, representing varying moduli for varying strain rates. The stress - strain rate curves, shown in Figure 11.16b, show that the Abaqus simulations and the analytical one-dimensional hereditary integral formulation track closely for a single element. Thus, for the strain rate in each direction, the implementation in Abaqus is working as expected for a one-dimensional case.

The simulation was then extended to a full three-dimensional arrangement, a quarter symmetry representation with the same global boundary conditions. Multiple mesh sizes were considered, and the finest mesh size was presented, however the scale of this mesh would not be sufficient to capture behaviour on a micro or meso scale. In order to verify that the implementation in a full three-dimensional model for a linear Maxwell was behaving as expected, the function of the strain rate in each direction was investigated with a value of p equal to 1, to establish a baseline of stress - strain rate curves. The stress - strain curves for the varying strain rates, see Figure 11.18a, shows expected linear behaviour. The Abaqus simulation and analytical one-dimensional hereditary integral for each curve track closely for the linear solution in Figure 11.18b, however, the linear solution ($p = 1$) does not capture the desired behaviour of the experimental results as expected.

Next, the value of p was adjusted to 2.5, which is the expected value of p as per the analytical one-dimensional hereditary integral results. The stress - strain responses of each simulation in Figure 11.18c look similar to the responses for the linear solution. Likewise, the stress - strain rate results from the Abaqus simulation in Figure 11.18d confirm this and they also do not capture the curve in the ISR regime as well as the analytical one-dimensional hereditary integral results.

The value of p was then increased to 4.5, above the expected value in order to further steepen the curve, however, the stress - strain results in Figure 11.18e are relatively unchanged. The uptake in Figure 11.18f is steeper, however, the rest of the ISR regime is not aligned with either the analytical one-dimensional hereditary integral results for p equal to 4.5 or the experimental results.

A comparison of all p values in Figure 11.19 shows that the initial uptake is steeper for p equal to 2.5 and p equal to 4.5 in the ISR regime than a linear Maxwell, however, the rest of the ISR regime is unchanged especially between p equal to 1 and p equal to 4.5. The reference strain rate used for the analytical one-dimensional hereditary integral results and the three-dimensional simulations was the same, and the anomalies that were observed in other result sets (which will be discussed in a later section) are not present in these results. These results have therefore shown that the strain rate in each direction for the dashpot coefficient function does not capture the expected behaviour

of the element once it is extended to three dimensions.

The root mean square error analysis in Figure 11.20a shows an increase in error as the strain increases. A value of $p = 2.5$ gives the lowest error at the highest strain of 0.02, while a value of $p = 1$ gives the highest error when comparing the Abaqus results with the hereditary integral for $p = 2.5$. The strain rates chosen for the simulations presented were chosen to be equally spaced on the logarithmic scale in Figure 11.19, but they are therefore not linearly spaced. This may result in a biasing of the error analysis using the root mean square. The error measure does still show the relative errors across values of p and the various strain curves.

An analysis of the area between the two curves as a percentage of the normalised area under the reference curve in Figure 11.20b shows varying percentage errors across the strain curves, however, the normalised area under the reference curve increases as the strain increases and therefore the relative error across strains is not accurately captured. The relative error between values of p at a given strain can, however, be examined. At lower strains, the highest error is for $p = 4.5$ and the lowest is for $p = 2.5$. As the strain increases, this starts to change with the highest error being for $p = 1$ at the highest strain and the lowest error being for $p = 2.5$. The logarithmic area percentage difference percentage plotted in Figure 11.20c shows that a value of $p = 2.5$ has the lowest error across all strains and between the Abaqus simulation and both the experimental best fit curve and the hereditary integral. The value of the error percentage is slightly higher than that previously calculated as the area difference percentage.

12.5.2 Cortical Bone Model - Dominant Strain Rate

The use of the dominant strain rate in the dashpot coefficient function was then investigated, where the dominant strain rate is in the direction of the applied load, along the axis of the specimen. A single element simulation was again carried out using a value of p equal to 2.5 in order to assess that the implementation of the dominant strain rate is behaving as expected in a one-dimensional simulation. The stress - strain curves in Figure 11.21a show expected straight line behaviour representing varying moduli for the varying strain rates. The stress - strain rate curves, shown in Figure 11.21b, show that the Abaqus simulations and the analytical one-dimensional hereditary integral formulation track closely for a single element, showing that for the dominant strain rate, the implementation in Abaqus is working as expected for a one-dimensional case.

The simulation was then extended to a full three-dimensional arrangement, firstly using a value of p equal to 1 to establish that the linear solution gives the expected results. The stress - strain results, see Figure 11.22a, show the expected behaviour. The stress - strain rate curves in Figure 11.22b show that the linear solution tracks closely to the analytical one-dimensional hereditary integral results, the same as when using the strain rate in each direction.

A value of p equal to 2.5 was used to investigate if using the dominant strain rate sharpens the curve in the ISR regime. The stress - strain results, Figure 11.22c, also shows similar behaviour to when the strain rate in each direction was used. This is confirmed in Figure 11.22d which shows that the simulations are still not capturing the gradient of the ISR regime as well as the one-dimensional hereditary integral.

Again the value of p was increased to 4.5 to determine if this would capture the ISR regime better than lower p values. The stress - strain results in Figure 11.22e look similar to those when using the strain rate in each direction, with a slight difference in the line for strain rate $7.58s^{-1}$ which is out of sequence, suggesting that some numerical issue occurred at that strain rate. The stress - strain rate curves in Figure 11.22f show that the Abaqus simulations do not capture the steepness of the ISR regime and deviate substantially from the desired result seen in the experimental and one-dimensional hereditary integral results. The numerical issue mentioned before is also seen in these graphs at $7.58s^{-1}$. The reference strain rate used for the analytical one-dimensional hereditary integral results and the three-dimensional simulations was the same.

Comparing all three results in Figure 11.23, very similar behaviour was observed to that when the strain rate in each direction was used. The curve of p equal to 4.5 has a steeper curve at the beginning of the ISR regime however then it flattens out and is similar to the linear solution. Again, the dominant strain rate is not capturing the representative behaviour in three dimensions. This solution also has some numerical instabilities.

The root mean square error analysis in Figure 11.24a shows a similar increase in error as the strain increases, with a value of $p = 2.5$ giving the lowest error and a value of $p = 1$ giving the highest error. An analysis of the area difference between the two curves as a percentage of the normalised area under the reference curve in Figure 11.24b shows the same varying percentage errors across the strain curves, however, the relative error between values of p at a given strain shows that at lower strains, the highest error is for $p = 4.5$ and the lowest is for $p = 2.5$. As the strain increases, the highest error is for $p = 1$ at the highest strain and the lowest is for $p = 2.5$. For the logarithmic area difference percentage in Figure 11.24c the lowest percentage error across all strains and between Abaqus and both reference curves is for $p = 2.5$.

12.5.3 Cortical Bone Model - Equivalent Strain Rate

The equivalent strain rate was first tested using a single element simulation with a value of p equal to 2.5 and a reference strain rate of 0.35 to assess a one-dimensional representation in Abaqus. The stress - strain curves in Figure 11.25a are closely grouped in the higher strain rates with a few strain rate simulations exhibiting a lower modulus. The stress - strain rate curves in Figure 11.25b confirm that there is a straight line behaviour at the higher strain rates of the simulation set and therefore the curve is shifted further to the left than was expected. In contrast to the

single element simulations carried out using the strain rate in each direction or the dominant strain rate, the coupling between strain rate directions, when using the equivalent strain rate raised to a power, is not representative of the analytical one-dimensional hereditary integral, as might have been expected. There also appear to be some issues with the simulations carried out at several strain rates ($6.33s^{-1}$, $10.86s^{-1}$ and $20s^{-1}$). These issues could be due to numerical instabilities or the interaction between strain rates in the different directions, however, this would require further investigation.

The reference strain rate was then increased to a value of 1.2 in order to shift the curve closer to that of the expected analytical one-dimensional hereditary integral and the experimental data. The stress - strain curves in Figure 11.26a show that some of the responses do not exhibit a constant modulus at different strains, and the stress - strain rate curves in Figure 11.26b confirm that this is the case. The curve has shifted closer to the analytical one-dimensional hereditary integral, however, the result is not what is expected and still shows some serious deviations from the expected behaviour in some simulations.

The simulation was then extended to a full three-dimensional arrangement. The approach of using the equivalent strain rate is what was proposed by Bekker *et al.* [2], and to ensure that the simulation can give the expected results as those achieved by Bekker *et al.* [2], the linear solution was simulated first. The stress - strain results in Figure 11.27a show the expected results as before, and the stress - strain rate curves in Figure 11.27b also show that the Abaqus simulations with a value of p equal to 1 matches the analytical one-dimensional hereditary integral results. This uses the same reference strain rate as the analytical one-dimensional hereditary integral result. This shows that the implementation of the equivalent strain rate is behaving as expected and that the issues that were seen in the single element simulation are likely to be associated with raising the power of the equivalent strain rate in the dashpot coefficient function.

Keeping the reference strain rate the same, the value of p was then increased to 2.5 and the stress - strain results were produced. These results, shown in Figure 11.27e, look different to previous result sets and there is a sharper increase in the gradient in the ISR regime. The stress - strain rate curves in Figure 11.27f show that the gradient has increased to match both the analytical one-dimensional hereditary integral results and the experimental results, and that the curve has been shifted to the left and an adjustment to the reference strain rate was needed. There are, however, deviations from expected behaviour at strain rates of $10s^{-1}$ and $7.58s^{-1}$.

To investigate if the numerical issue is linked to increasing the power, the value of p was set to 1.5 instead and the reference strain rate was kept the same. The stress - strain results in Figure 11.27c show a change from the linear results and an anomaly at the same strain rate as before. In addition, the stress - strain rate curves in Figure 11.27d show an increase in the gradient of the ISR regime to match the analytical one-dimensional hereditary integral result, however, not the experimental results as expected. The graph is also shifted, although less than with a value of p equal to 2.5, however, there are still obvious anomalies at $10s^{-1}$ and $7.58s^{-1}$. The general trend appears to be a

increasing number of issues in the expected result as the power is increased. Comparing the stress - strain rate curves in Figure 11.28, the gradient of the ISR regime increases to match the expected analytical one-dimensional hereditary integral behaviour. Once again deviations from the expected behaviour begin to appear as the power p increases away from a value of 1.

Since the value of p equal to 2.5 gives results that best match the ISR regime behaviour, the reference strain rate was increased to shift the curve to the right. The stress - strain results, Figure 11.29a, show some inconsistent behaviour in the gradient at different strains for a single strain rate test. The stress - strain rate curves in Figure 11.29b show that the curve has been shifted to the right with an increase in the reference strain rate, as expected, with the same anomalies at $7.58s^{-1}$, $10s^{-1}$ and $20s^{-1}$. Thus, the reference strain rate needs to be adjusted differently for the three-dimensional simulation as compared with that of the analytical one-dimensional hereditary integral results when using the equivalent strain rate.

To determine if there were any other potential issues at strain rates along the curve, a further 12 simulations were carried out, with the stress - strain results in Figure 11.29c show the same inconsistent behaviour. The stress - strain rate curves in Figure 11.29d show more instabilities and deviations from the expected result, which still need to be investigated further. While the one-dimensional non-linear Maxwell behaviour might be indicative of the experimental results and the expected behaviour, extending to three dimensions cannot be done without taking into account three-dimensional effects early on in the implementation.

The root mean square analysis in Figure 11.30a shows the same increase in error as the strain increases, with a value of $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$ giving the lowest error and a value of $p = 2.5$ and $\dot{\varepsilon}_{ref} = 0.35$ giving the highest error. This is due to the lower reference strain rate giving a curve that is shifted too far to the left. An analysis of the area difference percentage in Figure 11.30b shows the same varying percentage errors across the strain curves, however, the relative error between values of p at a given strain shows that at lower strains, the highest error is for $p = 1$ and the lowest is for $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$. As the strain increases, the highest error remains for $p = 1$ at the highest strain and the lowest is for $p = 1.5$. The results of the logarithmic area difference calculation in Figure 11.30c show that the lowest error is for $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$ across all strains and for both reference curves.

12.6 Summary of Three-Dimensional Investigations

A summary of the results of the simulations carried out in LS Dyna and Abaqus, is presented in Tables 12.1 and 12.2. Table 12.1 shows the force impulse verification tests for a linear elastic model, compared to a one-dimensional spring-mass system. For an hourglass control parameter $Q_2 = 6 \times 10^{-4}$, the logarithmic strain ($\ln \varepsilon$), that outputted by the FEA software (Output ε) and

the incremental strain (Incremental ε) were compared.

For the quarter symmetry Abaqus simulations, the area difference error analyses in Figures 11.20b, 11.24b and 11.30b show that the lowest error across the three different implementations of the dashpot coefficient function between Abaqus and the experimental best fit line was the equivalent strain rate implementation with a value of $p = 1.5$, and between Abaqus and the hereditary integral was the equivalent strain rate implementation with a value of $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$. The best fit curve to the experimental data, however, could introduce its own margin of error. The logarithmic area difference error analyses in Figures 11.20c, 11.24c and 11.30c, however, show that the lowest error percentage across the three implementations between Abaqus and the experimental best fit line was the equivalent strain rate implementation with a value of $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$, and between Abaqus and the hereditary integral was the dominant strain rate implementation with a value of $p = 2.5$. Although qualitatively, the equivalent strain rate implementation with $p = 2.5$ and $\dot{\varepsilon}_{ref} = 1.2$ shows the best match with the hereditary integral, the numerical anomalies observed in the middle of the ISR regime could be contributing to the higher error percentage seen in the error analysis.

Table 12.2 shows the results matrix of the constant strain rate tests carried out for the cortical bone user-defined material model, using a single element and quarter symmetry mesh in Abaqus. These simulations were carried out using three different ways of computing the strain rate used in the dashpot coefficient function for a non-linear Maxwell element. These were the strain rate in each direction, the dominant strain rate and an equivalent strain rate.

Table 12.1: Matrix of expected behaviour for force impulse tests using a linear elastic model in finite element software and compared to a one-dimensional spring-mass system.

		LS Dyna		Abaqus		One-Dimensional Spring-Mass System
		Built-In	User-Defined	Built-In	User-Defined	
Hourglassing Control						
None		○	○	–	–	●
$Q_2 = 6 \times 10^{-2}$		○	○	–	–	●
$Q_2 = 6 \times 10^{-4}$	$\ln \varepsilon$	◑	◑	◑	●	●
	Output ε	●	●	◑	●	–
	Incremental ε	–	–	–	–	●

Key: ● Model returns expected results ○ Model does not return expected results
◑ Model partially returns expected results

Table 12.2: Matrix of expected behaviour for constant strain rate three-dimensional simulations carried out in Abaqus for a cortical bone material model, and compared to the one-dimensional hereditary integral solution.

	Strain Rate In Each Direction	Dominant Strain Rate	Equivalent Strain Rate	One-dimensional Hereditary Integral
Single Element				
$p = 2.5 \quad \dot{\varepsilon}_{ref} = 0.35$	●	●	◔	●
$p = 2.5 \quad \dot{\varepsilon}_{ref} = 1.2$	–	–	◑	●
Quarter-Symmetry Mesh				
$p = 1 \quad \dot{\varepsilon}_{ref} = 0.35$	●	●	●	●
$p = 1.5 \quad \dot{\varepsilon}_{ref} = 0.35$	–	–	◑	●
$p = 2.5 \quad \dot{\varepsilon}_{ref} = 0.35$	◔	◔	◑	●
$p = 2.5 \quad \dot{\varepsilon}_{ref} = 1.2$	–	–	◕	●
$p = 4.5 \quad \dot{\varepsilon}_{ref} = 0.35$	◔	◔	–	●

Key: ● Model returns expected results ◑ Model partially returns expected results ◕ Model mostly returns expected results ◔ Model barely returns expected results

The implementation of the non-linear hereditary integral in a three-dimensional simulation builds on the model proposed by Cloete *et al.* [3], using concepts introduced by Shim *et al.* and Bekker *et al.* [1, 2]. The results in Chapter 11 show a different response at low and high strain rates which supports the findings by Johnson *et al.* [6]. Chapter 11 directly compares one-dimensional and three-dimensional implementations of a visco-elastic model. This type of comparison is not extensively found in the literature for models of this type for bone. This work therefore adds initial verification of the hereditary integral approach for modelling of visco-elastic materials such as bone to the body of knowledge.

Chapter 13

Recommendations and Future Work

This chapter discusses possible future work relating to the finite element implementation of the model. The long-term goal is for the material models presented in this dissertation to be used in patient-specific modelling of cortical bone. Before this is possible, further work is required in several aspects of the model. Firstly, the design of the dashpot coefficient for a non-linear dashpot, secondly the implementation of a non-linear dashpot in three dimensions and the choice of strain rate used, and thirdly the reliability of the finite element software.

13.1 Dashpot Coefficient Function

The choice of the dashpot coefficient function could be investigated further, especially at near-zero strain rates. This is particularly important for cases where both tensile and compressive waves are possible. The addition of a linear term was used to address this potential issue, however, for values of $p < 1$ at a zero strain rate the dashpot coefficient would still tend towards infinity. Values of $p < 1$ were not investigated extensively in this dissertation, but for a more generic non-linear Maxwell based model they would need to be considered.

13.2 Non-linear User Defined Material Model

The use of the equivalent strain rate in the dashpot coefficient with a power above 1 has produced some promising results, but the source of the anomalies requires further investigation. Adjusting the reference strain rate for varying values of the power p could also be investigated further for a more robust model. For non-linear models where strain rates are raised to a power, a time step

sensitivity investigation could provide more insight.

The three-dimensional effects of a non-linear user defined material model were apparent in the use of the equivalent strain rate due to the coupling of the strain rate in different directions. These three-dimensional effects need to be investigated earlier on in the development of the material model. Once these issues have been resolved, patient-specific parameters such as a density function as proposed by Shim *et al.* [1], can be introduced to move towards a more representative skeletal model.

13.3 Damping in FEA Software

Finite element software uses reduced order elements, which have reduced energy modes that are prone to hourglassing. Hourglassing control is generally done through the application of a bulk viscosity which can interfere with the dissipative effects of the chosen visco-elastic model. The damping, due to the default hourglassing control that was encountered, when simulating in LS Dyna and Abaqus was dealt with by adjusting the bulk viscosity controls. However, this is only one potential method of preventing hourglassing in FEA Software. There are several hourglassing controls that could be causing the deviations from the expected behaviour that were observed and these should be investigated further to determine if they could be effective at preventing hourglassing without obscuring the dissipative effects of the Maxwell dashpot.

Chapter 14

Conclusions

This dissertation has reported on the development and implementation of a material model for cortical bovine bone with the use of non-linear visco-elastic models. Fundamental visco-elastic models were investigated and a hereditary integral was derived from first principles that captures the behaviour of a linear Maxwell element. It was shown that the linear Maxwell model was not sufficient to capture the behaviour in the ISR regime, as reported by Cloete *et al.* [3], and the material model was therefore modified to allow for non-linearity in two different ways, by raising the strain rate or velocity to a power or by making the dashpot coefficient a function of the strain rate. The strain rate across the Maxwell dashpot in the material model is not directly available in the calculation of the hereditary integral, so the effect of using the strain rate across the entire element was investigated.

The hereditary integral was then extended to a three-dimensional model following the work of Shim *et al.* [1], where a three-dimensional strain rate tensor is introduced to capture the strain rates in three dimensions. The strain rate tensor used here could be constructed in several ways: using the strain rate in each direction; using the dominant strain rate; or using the equivalent strain rate. This three-dimensional extension of the material model for cortical bone was implemented in commercial finite element analysis software. During initial verification steps, unexpected results were produced when using user-defined and built-in time-dependent linear elasticity material models in LS Dyna. These unexpected results were reported on in this dissertation and further development was carried out using an alternative software, Abaqus. The expected results were produced when using a user-defined material model for linear elasticity.

A standard linear model was implemented in Abaqus to verify that the implementation of the hereditary integral was working in the finite element software. The implementation allows for the use of the equivalent strain rate in a manner similar to that taken by Bekker *et al.* [2]. The implementation in three dimensions revealed some discrepancies between the Abaqus simulation results and the results of the one-dimensional hereditary integral, dependent on the implementation

of the dashpot coefficient and the strain rate.

In a single element simulation, the strain rate in each direction and the dominant strain rate implementations produce the expected results when setting the non-linear parameter p to that which correctly captures the steepness of the ISR transition behaviour. Multi-element quarter-symmetry meshes representing cylindrical specimens were used to validate the three-dimensional models, where linear Maxwell elements produced the expected results. The effect of changing p in the quarter-symmetry simulations was less pronounced than that observed for the single-element models. This resulted in the steep transition behaviour in the ISR regime not being sufficiently captured.

When the equivalent strain rate is used with the value of p suggested by the one-dimensional results, the material model was able to capture the transition behaviour in the ISR regime in the quarter-symmetry models, provided an adjusted reference strain rate was used. The effect of changing the value of p is not consistent across various strain rates. This issue was investigated and was found to be present in all models, even single-element ones. Despite these issues, the model developed shows promise in capturing the ISR regime behaviour in three-dimensional finite element models and should be further developed in future projects.

References

[1] V. P. W. Shim, L. M. Yang, J. F. Liu, and V. S. Lee, "Characterisation of the dynamic compressive mechanical properties of cancellous bone from the human cervical spine," *International Journal of Impact Engineering*, vol. 32, no. 1-4, pp. 525–540, 2006.

[2] A. Bekker, S. Kok, T. J. Cloete, and G. N. Nurick, "Introducing objective power law rate dependence into a visco-elastic material model of bovine cortical bone," *International Journal of Impact Engineering*, vol. 66, pp. 28–36, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.ijimpeng.2013.12.003

[3] T. J. Cloete, G. Paul, and E. B. Ismail, "Hopkinson bar techniques for the intermediate strain rate testing of bovine cortical bone." *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 372, no. 2015, p. 20130210, 2014.

[4] R. R. Adharapurapu, F. Jiang, and K. S. Vecchio, "Dynamic fracture of bovine bone," *Materials Science and Engineering C*, vol. 26, no. 8, pp. 1325–1332, 2006.

[5] T. Cloete, G. Paul, and E. Ismail, "A preliminary investigation of the dynamic viscoelastic relaxation of bovine cortical bone," *EPJ Web of Conferences*, vol. 94, p. 03004, 2015. [Online]. Available: http://www.epj-conferences.org/10.1051/epjconf/20159403004

[6] T. P. Johnson, S. Socrate, and M. C. Boyce, "A viscoelastic, viscoplastic model of cortical bone valid at low and high strain rates," *Acta Biomaterialia*, vol. 6, no. 10, pp. 4073–4080, 2010. [Online]. Available: http://dx.doi.org/10.1016/j.actbio.2010.04.017

[7] J. Aerssens, S. Boonen, G. Lowet, and J. Dequeker, "Interspecies differences in bone composition, density, and quality: Potential implications for in vivo bone research," *Endocrinology*, vol. 139, no. 2, pp. 663–670, 1998.

[8] a. van der Westhuizen, "The strain rate dependent mechanical properties and modelling of bovine cortical bone in compression," Ph.D. dissertation, 2008.

[9] B. Clarke, "Normal Bone Anatomy and Physiology," *Clin J Am Soc Nephrol*, vol. 3, no. Suppl 3, pp. 131–139, 2008.

[10] S. Firestein, *KELLEY & FIRESTEIN'S Textbook of Rheumatology*, 2017.

[11] K. Moore and A. Dalley, *Moore's Clinically Oriented Anatomy*, 1999.

[12] J. Y. Rho, L. Kuhn-Spearing, and P. Zioupos, "Mechanical properties and the hierarchical structure of bone," *Medical Engineering and Physics*, vol. 20, no. 2, pp. 92–102, 1998.

[13] N. C. Tennyson, R. Ewert, and V. Niranjan, "Dynamic viscoelastic response of bone," *Experimental Mechanics*, vol. 12, p. 502, 1972.

[14] Y. Tanabe and K. Kobayashi, "Anisotropy in the dynamic non-linear viscoelastic properties of bovine compact bone," *Journal of Materials Science: Materials in Medicine*, vol. 5, no. 6-7, pp. 397–401, 1994.

[15] J. T. Cheung and M. Zhang, "7 - Mechanics of the human skin and underlying soft tissues," in *Woodhead Publishing Series in Textiles*, Y. Li, X.-Q. B. T. B. E. o. T. Dai, and Clothing, Eds. Woodhead Publishing, 2006, pp. 111–124. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781845690526500078

[16] M. Capurro and F. Barberis, "9 - Evaluating the mechanical properties of biomaterials," P. Dubruel and S. B. T. B. f. B. R. Van Vlierberghe, Eds. Woodhead Publishing, 2014, pp. 270–323. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780857098047500091

[17] G. C. Papanicolaou and S. P. Zaoutsos, "1 - Viscoelastic constitutive modeling of creep and stress relaxation in polymers and polymer matrix composites," in *Woodhead Publishing Series in Composites Science and Engineering*, R. M. B. T. C. Guedes and F. in Polymer Matrix Composites, Eds. Woodhead Publishing, 2011, pp. 3–47. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781845696566500014

[18] P. A. Kelly, "Solid Mechanics Part I: An Introduction to Solid Mechanics," 2020, pp. 283–342.

[19] R. Lakes, *Viscoelastic Materials*. Cambridge University Press, 2009.

[20] Y. Fung, *A First Course in Continuum Mechanics for Physical and Biologocal Engineers and Scientists*, 1994.

[21] J. Bonet and R. D. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, 2nd ed. Cambridge University Press, 2008.

[22] D. L. Logan, *A First Course in the Finite Element Method Fourth Edition*. Thomson, 2007.

[23] R. D. Cook and H. Saunders, "Concepts and Applications of Finite Element Analysis (2nd Edition)," *Journal of Pressure Vessel Technology*, vol. 106, no. 1, pp. 127–127, 1984.

[24] LIVERMORE SOFTWARE TECHNOLOGYCORPORATION, *LS-Dyna Keyword User's Manual Volume I*, 2017.

[25] Dassault Systemes Simulia Inc., "Abaqus Analysis User's Manual Version 6.11."

[26] LIVERMORE SOFTWARE TECHNOLOGY CORPORATION, *LS-Dyna Theory Manual*, 2017.

[27] G. Dhondt, "The Finite Element Method for Three-Dimensional Thermomechanical Applications," dec 2004.

[28] Dassault Systemes Simulia Inc., "Abaqus Theory Manual Version 6.11."

Appendix A

LS Dyna UMAT Code

A.1 Linear Elastic UMAT

```
      subroutine umat41 (cm,eps,sig,epsp,hsv,dt1,capa,etype,tt,
     1       temper,failel,crv,nnpcrv,cma,qmat,elsiz,idele,reject)
      c
      c************************************************************************
      c|  Livermore Software Technology Corporation  (LSTC)                    |
      c|  -------------------------------------------------------------------  |
      c|  Copyright 1987-2008 Livermore Software Tech. Corp                    |
      c|  All rights reserved                                                  |
      c************************************************************************
      c
            include 'nlqparm'
            include 'bk06.inc'
            include 'iounits.inc'
            dimension cm(*),eps(*),sig(*),hsv(*),crv(lq1,2,*),cma(*),qmat(3,3)
            integer nnpcrv(*)
            logical failel,reject
            character*5 etype
            INTEGER8 idele
      c
            double precision :: k1,v,lambda1,lambda2,dtrace,trace
      c
            logical :: created
      c
            if (ncycle.eq.1) then
              if (cm(16).ne.1234567) then
                call usermsg('mat41')
              endif
            endif
      c
            if (etype.eq.'solid'.or.etype.eq.'shl_t'.or.
     1           etype.eq.'sld2d'.or.etype.eq.'tshel'.or.
     2           etype.eq.'sph  '.or.etype.eq.'sldax') then
      c
      c************************************************************************
                k1 =cm(1)
                v =cm(2)
                lambda1 =v/((1.+v)*(1.-2.*v))
                lambda2 =1./(1.+v)
      c
      c************************************************************************
                trace=hsv(1)+eps(1)+hsv(2)+eps(2)+hsv(3)+eps(3)
      c
                sig(1)=k1*(lambda1*trace+lambda2*(hsv(1)+eps(1)))
                sig(2)=k1*(lambda1*trace+lambda2*(hsv(2)+eps(2)))
                sig(3)=k1*(lambda1*trace+lambda2*(hsv(3)+eps(3)))
                sig(4)=k1*lambda2*(hsv(4)+eps(4))
                sig(5)=k1*lambda2*(hsv(5)+eps(5))
                sig(6)=k1*lambda2*(hsv(6)+eps(6))
      c
                hsv(1)=hsv(1)+eps(1)
                hsv(2)=hsv(2)+eps(2)
                hsv(3)=hsv(3)+eps(3)
                hsv(4)=hsv(4)+eps(4)
                hsv(5)=hsv(5)+eps(5)
                hsv(6)=hsv(6)+eps(6)
      c************************************************************************
      c
            endif
      c
            return
            end
```

Appendix B

Abaqus VUMAT Code

B.1 Linear Elastic - VUMAT

```
      subroutine vumat(
C Read only (unmodifiable)variables -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
C      include 'vaba_param.inc'
C
      dimension props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(*),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(*),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(*), enerInelasNew(*)
C
      double precision :: xk1,v,lambda1,lambda2,trace
      character*80 cmname
C
C Define model variables:
      xk1 = props(1)
      v = props(2)
C
C     print*, xk1
C     print*, v
      do 100 i = 1,nblock

        lambda1 = v/((1.0+v)*(1.0-2.0*v))
C       print*, lambda1
        lambda2 = 1.0/(1.0+v)
C       print*, lambda2
        trace = stateOld(i,1)+strainInc(i,1)+
     1          stateOld(i,2)+strainInc(i,2)+
     2          stateOld(i,3)+strainInc(i,3)
      if (totalTime .eq. 0) then
        stressNew(i,1) = (xk1)*strainInc(i,1)
        stressNew(i,2) = (xk1)*strainInc(i,2)
        stressNew(i,3) = (xk1)*strainInc(i,3)
        stressNew(i,4) = (xk1)*strainInc(i,4)
        stressNew(i,5) = (xk1)*strainInc(i,5)
        stressNew(i,6) = (xk1)*strainInc(i,6)
      else
        stressNew(i,1) = xk1*(lambda1*trace+lambda2*(stateOld(i,1)+strainInc(i,1)))
        stressNew(i,2) = xk1*(lambda1*trace+lambda2*(stateOld(i,2)+strainInc(i,2)))
        stressNew(i,3) = xk1*(lambda1*trace+lambda2*(stateOld(i,3)+strainInc(i,3)))
        stressNew(i,4) = xk1*(lambda2*(stateOld(i,4)+strainInc(i,4)))
        stressNew(i,5) = xk1*(lambda2*(stateOld(i,5)+strainInc(i,5)))
        stressNew(i,6) = xk1*(lambda2*(stateOld(i,6)+strainInc(i,6)))
      end if
C
        stateNew(i,1)=stateOld(i,1)+strainInc(i,1)
        stateNew(i,2)=stateOld(i,2)+strainInc(i,2)
        stateNew(i,3)=stateOld(i,3)+strainInc(i,3)
        stateNew(i,4)=stateOld(i,4)+strainInc(i,4)
        stateNew(i,5)=stateOld(i,5)+strainInc(i,5)
        stateNew(i,6)=stateOld(i,6)+strainInc(i,6)
  100 continue

      return
      end
```

B.2 Standard Linear - VUMAT

```
      subroutine vumat(
C Read only (unmodifiable)variables -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
C      include 'vaba_param.inc'
C
      dimension props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(*),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(*),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(*), enerInelasNew(*)
C
      double precision :: xk1,xk2,mu2,theta2,v,lambda1,lambda2
      double precision :: trace,decay,tracedot
      double precision :: sigE1,sigE2,sigE3,sigE4,sigE5,sigE6
      double precision :: sigM1_1,sigM1_2,sigM1_3,sigM1_4,sigM1_5,sigM1_6
      double precision :: epsdot1,epsdot2,epsdot3,epsdot4,epsdot5,epsdot6
      character*80 cmname
C
C Define model variables:
      xk1 = props(1)
      xk2 = props(2)
      theta2 = props(3)
      v = props(4)
C
C     print*, xk1
C     print*, v
      do 100 i = 1,nblock
        mu2=theta2*xk2
        lambda1 = v/((1.0+v)*(1.0-2.0*v))
C       print*, lambda1
        lambda2 = 1.0/(1.0+v)
C       print*, lambda2
        trace = stateOld(i,1)+strainInc(i,1)+
     1          stateOld(i,2)+strainInc(i,2)+
     2          stateOld(i,3)+strainInc(i,3)
C
        sigE1 = xk1*(lambda1*trace+lambda2*(stateOld(i,1)+strainInc(i,1)))
        sigE2 = xk1*(lambda1*trace+lambda2*(stateOld(i,2)+strainInc(i,2)))
        sigE3 = xk1*(lambda1*trace+lambda2*(stateOld(i,3)+strainInc(i,3)))
        sigE4 = xk1*(lambda2*(stateOld(i,4)+strainInc(i,4)))
        sigE5 = xk1*(lambda2*(stateOld(i,5)+strainInc(i,5)))
        sigE6 = xk1*(lambda2*(stateOld(i,6)+strainInc(i,6)))
C
        epsdot1=(strainInc(i,1)/dt)
        epsdot2=(strainInc(i,2)/dt)
        epsdot3=(strainInc(i,3)/dt)
        epsdot4=(strainInc(i,4)/dt)
        epsdot5=(strainInc(i,5)/dt)
        epsdot6=(strainInc(i,6)/dt)
C
        tracedot=epsdot1+epsdot2+epsdot3
C
        decay=exp(-dt/theta2)
C
        sigM1_1=decay*stateOld(i,7)+(1.-decay)*mu2*(lambda1*tracedot+
     1              +lambda2*epsdot1)
        sigM1_2=decay*stateOld(i,8)+(1.-decay)*mu2*(lambda1*tracedot+
     1              +lambda2*epsdot2)
        sigM1_3=decay*stateOld(i,9)+(1.-decay)*mu2*(lambda1*tracedot+
     1              +lambda2*epsdot3)
        sigM1_4=decay*stateOld(i,10)+(1.-decay)*mu2*((.5*lambda2)*epsdot4)
```

```
        sigM1_5=decay*stateOld(i,11)+(1.-decay)*mu2*((.5*lambda2)*epsdot5)
        sigM1_6=decay*stateOld(i,12)+(1.-decay)*mu2*((.5*lambda2)*epsdot6)
C
      if (totalTime .eq. 0) then
        stressNew(i,1) = (xk1+xk2)*strainInc(i,1)
        stressNew(i,2) = (xk1+xk2)*strainInc(i,2)
        stressNew(i,3) = (xk1+xk2)*strainInc(i,3)
        stressNew(i,4) = (xk1+xk2)*strainInc(i,4)
        stressNew(i,5) = (xk1+xk2)*strainInc(i,5)
        stressNew(i,6) = (xk1+xk2)*strainInc(i,6)
      else
        stressNew(i,1) = sigE1+sigM1_1
        stressNew(i,2) = sigE2+sigM1_2
        stressNew(i,3) = sigE3+sigM1_3
        stressNew(i,4) = sigE4+sigM1_4
        stressNew(i,5) = sigE5+sigM1_5
        stressNew(i,6) = sigE6+sigM1_6
      end if
C
        stateNew(i,1)=stateOld(i,1)+strainInc(i,1)
        stateNew(i,2)=stateOld(i,2)+strainInc(i,2)
        stateNew(i,3)=stateOld(i,3)+strainInc(i,3)
        stateNew(i,4)=stateOld(i,4)+strainInc(i,4)
        stateNew(i,5)=stateOld(i,5)+strainInc(i,5)
        stateNew(i,6)=stateOld(i,6)+strainInc(i,6)
C
        stateNew(i,7)=sigM1_1
        stateNew(i,8)=sigM1_2
        stateNew(i,9)=sigM1_3
        stateNew(i,10)=sigM1_4
        stateNew(i,11)=sigM1_5
        stateNew(i,12)=sigM1_6
  100 continue

      return
      end
```

B.3 Cortical Model - Strain Rate in Each Direction - VUMAT

```
      subroutine vumat(
      C Read only (unmodifiable)variables -
           1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
           2  stepTime, totalTime, dt, cmname, coordMp, charLength,
           3  props, density, strainInc, relSpinInc,
           4  tempOld, stretchOld, defgradOld, fieldOld,
           5  stressOld, stateOld, enerInternOld, enerInelasOld,
           6  tempNew, stretchNew, defgradNew, fieldNew,
      C Write only (modifiable) variables -
           7  stressNew, stateNew, enerInternNew, enerInelasNew )
      C
      C      include 'vaba_param.inc'
      C
            dimension props(nprops), density(nblock),
           1  coordMp(nblock,*),
           2  charLength(*), strainInc(nblock,ndir+nshr),
           3  relSpinInc(*), tempOld(*),
           4  stretchOld(*), defgradOld(*),
           5  fieldOld(*), stressOld(nblock,ndir+nshr),
           6  stateOld(nblock,nstatev), enerInternOld(nblock),
           7  enerInelasOld(nblock), tempNew(*),
           8  stretchNew(*), defgradNew(*), fieldNew(*),
           9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
           1  enerInternNew(*), enerInelasNew(*)
      C
            double precision :: xk1,xk2,emu2,v,lambda1,lambda2,n,epsdot_ref
            double precision :: trace,decay,tracedot
            double precision :: strainsumsquared,strainrate_eq,eq_power
            double precision :: sigE1,sigE2,sigE3,sigE4,sigE5,sigE6
            double precision :: sigM1_1,sigM1_2,sigM1_3,sigM1_4,sigM1_5,sigM1_6
            double precision :: epsdot1,epsdot2,epsdot3,epsdot4,epsdot5,epsdot6
            double precision :: emu_func1,emu_func2,emu_func3,emu_func4,emu_func5,emu_func6
            double precision :: eq_power1,eq_power2,eq_power3,eq_power4,eq_power5,eq_power6
            character*80 cmname
      C
      C Define model variables:
            xk1 = props(1)
            xk2 = props(2)
            emu2 = props(3)
            n = props(4)
            epsdot_ref = props(5)
            v = props(6)
      C
      C     print*, xk1
      C     print*, v
            do 100 i = 1,nblock
              lambda1 = v/((1.0+v)*(1.0-2.0*v))
      C       print*, lambda1
              lambda2 = 1.0/(1.0+v)
      C       print*, lambda2
              trace = stateOld(i,1)+strainInc(i,1)+
           1          stateOld(i,2)+strainInc(i,2)+
           2          stateOld(i,3)+strainInc(i,3)
      C
              sigE1 = xk1*(lambda1*trace+lambda2*(stateOld(i,1)+strainInc(i,1)))
              sigE2 = xk1*(lambda1*trace+lambda2*(stateOld(i,2)+strainInc(i,2)))
              sigE3 = xk1*(lambda1*trace+lambda2*(stateOld(i,3)+strainInc(i,3)))
              sigE4 = xk1*(lambda2*(stateOld(i,4)+strainInc(i,4)))
              sigE5 = xk1*(lambda2*(stateOld(i,5)+strainInc(i,5)))
              sigE6 = xk1*(lambda2*(stateOld(i,6)+strainInc(i,6)))
      C
              epsdot1=(strainInc(i,1)/dt)
              epsdot2=(strainInc(i,2)/dt)
              epsdot3=(strainInc(i,3)/dt)
              epsdot4=(strainInc(i,4)/dt)
              epsdot5=(strainInc(i,5)/dt)
              epsdot6=(strainInc(i,6)/dt)
      C
              tracedot=epsdot1+epsdot2+epsdot3
      C
              if( epsdot1 .gt. 0.0) then
                  eq_power1 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot1)**(n - 1.0))
              else
                  eq_power1 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot1)**(n - 1.0))
              end if
```

```
C
        if( epsdot2 .gt. 0.0) then
            eq_power2 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot2)**(n - 1.0))
        else
            eq_power2 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot2)**(n - 1.0))
        end if
C
        if( epsdot3 .gt. 0.0) then
            eq_power3 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot3)**(n - 1.0))
        else
            eq_power3 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot3)**(n - 1.0))
        end if
C
        if( epsdot4 .gt. 0.0) then
            eq_power4 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot4)**(n - 1.0))
        else
            eq_power4 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot4)**(n - 1.0))
        end if
C
        if( epsdot5 .gt. 0.0) then
            eq_power5 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot5)**(n - 1.0))
        else
            eq_power5 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot5)**(n - 1.0))
        end if
C
        if( epsdot6 .gt. 0.0) then
            eq_power6 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot6)**(n - 1.0))
        else
            eq_power6 = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot6)**(n - 1.0))
        end if
C
        emu_func1 = emu2*(1.0+eq_power1)
        emu_func2 = emu2*(1.0+eq_power2)
        emu_func3 = emu2*(1.0+eq_power3)
        emu_func4 = emu2*(1.0+eq_power4)
        emu_func5 = emu2*(1.0+eq_power5)
        emu_func6 = emu2*(1.0+eq_power6)

        sigM1_1=exp(-(dt*xk2)/emu_func1)*stateOld(i,7)+
     1              +(1.-exp(-(dt*xk2)/emu_func1))*emu_func1*(lambda1*tracedot+lambda2*epsdot1)
        sigM1_2=exp(-(dt*xk2)/emu_func2)*stateOld(i,8)+
     1              +(1.-exp(-(dt*xk2)/emu_func2))*emu_func2*(lambda1*tracedot+lambda2*epsdot2)
        sigM1_3=exp(-(dt*xk2)/emu_func3)*stateOld(i,9)+
     1              +(1.-exp(-(dt*xk2)/emu_func3))*emu_func3*(lambda1*tracedot+lambda2*epsdot3)
        sigM1_4=exp(-(dt*xk2)/emu_func4)*stateOld(i,10)+
     1              +(1.-exp(-(dt*xk2)/emu_func4))*emu_func4*((.5*lambda2)*epsdot4)
        sigM1_5=exp(-(dt*xk2)/emu_func5)*stateOld(i,11)+
     1              +(1.-exp(-(dt*xk2)/emu_func5))*emu_func5*((.5*lambda2)*epsdot5)
        sigM1_6=exp(-(dt*xk2)/emu_func6)*stateOld(i,12)+
     1              +(1.-exp(-(dt*xk2)/emu_func6))*emu_func6*((.5*lambda2)*epsdot6)
C
       if (totalTime .eq. 0) then
          stressNew(i,1) = (xk1+xk2)*strainInc(i,1)
          stressNew(i,2) = (xk1+xk2)*strainInc(i,2)
          stressNew(i,3) = (xk1+xk2)*strainInc(i,3)
          stressNew(i,4) = (xk1+xk2)*strainInc(i,4)
          stressNew(i,5) = (xk1+xk2)*strainInc(i,5)
          stressNew(i,6) = (xk1+xk2)*strainInc(i,6)
       else
          stressNew(i,1) = sigE1+sigM1_1
          stressNew(i,2) = sigE2+sigM1_2
          stressNew(i,3) = sigE3+sigM1_3
          stressNew(i,4) = sigE4+sigM1_4
          stressNew(i,5) = sigE5+sigM1_5
          stressNew(i,6) = sigE6+sigM1_6
       end if
C
        stateNew(i,1)=stateOld(i,1)+strainInc(i,1)
        stateNew(i,2)=stateOld(i,2)+strainInc(i,2)
        stateNew(i,3)=stateOld(i,3)+strainInc(i,3)
        stateNew(i,4)=stateOld(i,4)+strainInc(i,4)
        stateNew(i,5)=stateOld(i,5)+strainInc(i,5)
        stateNew(i,6)=stateOld(i,6)+strainInc(i,6)
C
        stateNew(i,7)=sigM1_1
```

```
      stateNew(i,8)=sigM1_2
      stateNew(i,9)=sigM1_3
      stateNew(i,10)=sigM1_4
      stateNew(i,11)=sigM1_5
      stateNew(i,12)=sigM1_6
100   continue

      return
      end
```

B.4 Cortical Model - Dominant Strain Rate - VUMAT

```
      subroutine vumat(
      C Read only (unmodifiable)variables -
           1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
           2  stepTime, totalTime, dt, cmname, coordMp, charLength,
           3  props, density, strainInc, relSpinInc,
           4  tempOld, stretchOld, defgradOld, fieldOld,
           5  stressOld, stateOld, enerInternOld, enerInelasOld,
           6  tempNew, stretchNew, defgradNew, fieldNew,
      C Write only (modifiable) variables -
           7  stressNew, stateNew, enerInternNew, enerInelasNew )
      C
      C      include 'vaba_param.inc'
      C
            dimension props(nprops), density(nblock),
           1  coordMp(nblock,*),
           2  charLength(*), strainInc(nblock,ndir+nshr),
           3  relSpinInc(*), tempOld(*),
           4  stretchOld(*), defgradOld(*),
           5  fieldOld(*), stressOld(nblock,ndir+nshr),
           6  stateOld(nblock,nstatev), enerInternOld(nblock),
           7  enerInelasOld(nblock), tempNew(*),
           8  stretchNew(*), defgradNew(*), fieldNew(*),
           9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
           1  enerInternNew(*), enerInelasNew(*)
      C
            double precision :: xk1,xk2,emu2,v,lambda1,lambda2,n,epsdot_ref,emu_func
            double precision :: trace,decay,tracedot
            double precision :: strainsumsquared,strainrate_eq,eq_power
            double precision :: sigE1,sigE2,sigE3,sigE4,sigE5,sigE6
            double precision :: sigM1_1,sigM1_2,sigM1_3,sigM1_4,sigM1_5,sigM1_6
            double precision :: epsdot1,epsdot2,epsdot3,epsdot4,epsdot5,epsdot6
            character*80 cmname
      C
      C Define model variables:
            xk1 = props(1)
            xk2 = props(2)
            emu2 = props(3)
            n = props(4)
            epsdot_ref = props(5)
            v = props(6)
      C
      C     print*, xk1
      C     print*, v
            do 100 i = 1,nblock
              lambda1 = v/((1.0+v)*(1.0-2.0*v))
      C       print*, lambda1
              lambda2 = 1.0/(1.0+v)
      C       print*, lambda2
              trace = stateOld(i,1)+strainInc(i,1)+
           1          stateOld(i,2)+strainInc(i,2)+
           2          stateOld(i,3)+strainInc(i,3)
      C
              sigE1 = xk1*(lambda1*trace+lambda2*(stateOld(i,1)+strainInc(i,1)))
              sigE2 = xk1*(lambda1*trace+lambda2*(stateOld(i,2)+strainInc(i,2)))
              sigE3 = xk1*(lambda1*trace+lambda2*(stateOld(i,3)+strainInc(i,3)))
              sigE4 = xk1*(lambda2*(stateOld(i,4)+strainInc(i,4)))
              sigE5 = xk1*(lambda2*(stateOld(i,5)+strainInc(i,5)))
              sigE6 = xk1*(lambda2*(stateOld(i,6)+strainInc(i,6)))
      C
              epsdot1=(strainInc(i,1)/dt)
              epsdot2=(strainInc(i,2)/dt)
              epsdot3=(strainInc(i,3)/dt)
              epsdot4=(strainInc(i,4)/dt)
              epsdot5=(strainInc(i,5)/dt)
              epsdot6=(strainInc(i,6)/dt)
      C
              tracedot=epsdot1+epsdot2+epsdot3
      C
              if( epsdot3 .gt. 0.0) then
                  eq_power = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((epsdot3)**(n - 1.0))
              else
                  eq_power = (6.5**(1.0 - n))*((6.5/epsdot_ref)**(2.5 - 1.0))*((-epsdot3)**(n - 1.0))
              end if
      C
              emu_func = emu2*(1.0+eq_power)
```

```
        sigM1_1=exp(-(dt*xk2)/emu_func)*stateOld(i,7)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot1)
        sigM1_2=exp(-(dt*xk2)/emu_func)*stateOld(i,8)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot2)
        sigM1_3=exp(-(dt*xk2)/emu_func)*stateOld(i,9)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot3)
        sigM1_4=exp(-(dt*xk2)/emu_func)*stateOld(i,10)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot4)
        sigM1_5=exp(-(dt*xk2)/emu_func)*stateOld(i,11)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot5)
        sigM1_6=exp(-(dt*xk2)/emu_func)*stateOld(i,12)+
     1              +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot6)
C
        if (totalTime .eq. 0) then
          stressNew(i,1) = (xk1+xk2)*strainInc(i,1)
          stressNew(i,2) = (xk1+xk2)*strainInc(i,2)
          stressNew(i,3) = (xk1+xk2)*strainInc(i,3)
          stressNew(i,4) = (xk1+xk2)*strainInc(i,4)
          stressNew(i,5) = (xk1+xk2)*strainInc(i,5)
          stressNew(i,6) = (xk1+xk2)*strainInc(i,6)
        else
          stressNew(i,1) = sigE1+sigM1_1
          stressNew(i,2) = sigE2+sigM1_2
          stressNew(i,3) = sigE3+sigM1_3
          stressNew(i,4) = sigE4+sigM1_4
          stressNew(i,5) = sigE5+sigM1_5
          stressNew(i,6) = sigE6+sigM1_6
        end if
C
          stateNew(i,1)=stateOld(i,1)+strainInc(i,1)
          stateNew(i,2)=stateOld(i,2)+strainInc(i,2)
          stateNew(i,3)=stateOld(i,3)+strainInc(i,3)
          stateNew(i,4)=stateOld(i,4)+strainInc(i,4)
          stateNew(i,5)=stateOld(i,5)+strainInc(i,5)
          stateNew(i,6)=stateOld(i,6)+strainInc(i,6)
C
          stateNew(i,7)=sigM1_1
          stateNew(i,8)=sigM1_2
          stateNew(i,9)=sigM1_3
          stateNew(i,10)=sigM1_4
          stateNew(i,11)=sigM1_5
          stateNew(i,12)=sigM1_6
  100   continue

        return
        end
```

B.5 Cortical Model - Equivalent Strain Rate - VUMAT

```
      subroutine vumat(
C Read only (unmodifiable)variables -
     1  nblock, ndir, nshr, nstatev, nfieldv, nprops, lanneal,
     2  stepTime, totalTime, dt, cmname, coordMp, charLength,
     3  props, density, strainInc, relSpinInc,
     4  tempOld, stretchOld, defgradOld, fieldOld,
     5  stressOld, stateOld, enerInternOld, enerInelasOld,
     6  tempNew, stretchNew, defgradNew, fieldNew,
C Write only (modifiable) variables -
     7  stressNew, stateNew, enerInternNew, enerInelasNew )
C
C      include 'vaba_param.inc'
C
      dimension props(nprops), density(nblock),
     1  coordMp(nblock,*),
     2  charLength(*), strainInc(nblock,ndir+nshr),
     3  relSpinInc(*), tempOld(*),
     4  stretchOld(*), defgradOld(*),
     5  fieldOld(*), stressOld(nblock,ndir+nshr),
     6  stateOld(nblock,nstatev), enerInternOld(nblock),
     7  enerInelasOld(nblock), tempNew(*),
     8  stretchNew(*), defgradNew(*), fieldNew(*),
     9  stressNew(nblock,ndir+nshr), stateNew(nblock,nstatev),
     1  enerInternNew(*), enerInelasNew(*)
C
      double precision :: xk1,xk2,emu2,v,lambda1,lambda2,n,epsdot_ref,emu_func
      double precision :: trace,decay,tracedot
      double precision :: strainsumsquared,strainrate_eq,eq_power
      double precision :: sigE1,sigE2,sigE3,sigE4,sigE5,sigE6
      double precision :: sigM1_1,sigM1_2,sigM1_3,sigM1_4,sigM1_5,sigM1_6
      double precision :: epsdot1,epsdot2,epsdot3,epsdot4,epsdot5,epsdot6
      double precision :: epsdot1_tot,epsdot2_tot,epsdot3_tot,epsdot4_tot,epsdot5_tot,epsdot6_tot
      character*80 cmname
C
C Define model variables:
      xk1 = props(1)
      xk2 = props(2)
      emu2 = props(3)
      n = props(4)
      epsdot_ref = props(5)
      v = props(6)
C
C     print*, xk1
C     print*, v
      do 100 i = 1,nblock
        lambda1 = v/((1.0+v)*(1.0-2.0*v))
C       print*, lambda1
        lambda2 = 1.0/(1.0+v)
C       print*, lambda2
        trace = stateOld(i,1)+strainInc(i,1)+
     1          stateOld(i,2)+strainInc(i,2)+
     2          stateOld(i,3)+strainInc(i,3)
C
        sigE1 = xk1*(lambda1*trace+lambda2*(stateOld(i,1)+strainInc(i,1)))
        sigE2 = xk1*(lambda1*trace+lambda2*(stateOld(i,2)+strainInc(i,2)))
        sigE3 = xk1*(lambda1*trace+lambda2*(stateOld(i,3)+strainInc(i,3)))
        sigE4 = xk1*(lambda2*(stateOld(i,4)+strainInc(i,4)))
        sigE5 = xk1*(lambda2*(stateOld(i,5)+strainInc(i,5)))
        sigE6 = xk1*(lambda2*(stateOld(i,6)+strainInc(i,6)))
C
        epsdot1=(strainInc(i,1)/dt)
        epsdot2=(strainInc(i,2)/dt)
        epsdot3=(strainInc(i,3)/dt)
        epsdot4=(strainInc(i,4)/dt)
        epsdot5=(strainInc(i,5)/dt)
        epsdot6=(strainInc(i,6)/dt)
C
        epsdot1_tot=((strainInc(i,1)+stateOld(i,1))/dt)
        epsdot2_tot=((strainInc(i,2)+stateOld(i,1))/dt)
        epsdot3_tot=((strainInc(i,3)+stateOld(i,1))/dt)
        epsdot4_tot=((strainInc(i,4)+stateOld(i,1))/dt)
        epsdot5_tot=((strainInc(i,5)+stateOld(i,1))/dt)
        epsdot6_tot=((strainInc(i,6)+stateOld(i,1))/dt)
C
        tracedot=epsdot1+epsdot2+epsdot3
```

```
C
      strainsumsquared = strainInc(i,1)**2.0
     1                 + strainInc(i,2)**2.0
     2                 + strainInc(i,3)**2.0
     3                 + 2.0 * strainInc(i,4)**2.0
     4                 + 2.0 * strainInc(i,5)**2.0
     5                 + 2.0 * strainInc(i,6)**2.0
      strainrate_eq = sqrt(strainsumsquared/(1.0+2.0 * v**2.0))/dt
C
      if( strainrate_eq .gt. 0.0) then
          eq_power = (6.5**(1.0 - n))*
                                ((6.5/epsdot_ref)**(2.5 - 1.0))*((strainrate_eq)**(n - 1.0))
      else
          eq_power = (6.5**(1.0 - n))*
                                ((6.5/epsdot_ref)**(2.5 - 1.0))*((-strainrate_eq)**(n - 1.0))
      end if
C
      emu_func = emu2*(1.0+eq_power)

      sigM1_1=exp(-(dt*xk2)/emu_func)*stateOld(i,7)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot1)
      sigM1_2=exp(-(dt*xk2)/emu_func)*stateOld(i,8)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot2)
      sigM1_3=exp(-(dt*xk2)/emu_func)*stateOld(i,9)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*(lambda1*tracedot+lambda2*epsdot3)
      sigM1_4=exp(-(dt*xk2)/emu_func)*stateOld(i,10)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot4)
      sigM1_5=exp(-(dt*xk2)/emu_func)*stateOld(i,11)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot5)
      sigM1_6=exp(-(dt*xk2)/emu_func)*stateOld(i,12)+
     1             +(1.-exp(-(dt*xk2)/emu_func))*emu_func*((.5*lambda2)*epsdot6)
C
     if (totalTime .eq. 0) then
       stressNew(i,1) = (xk1+xk2)*strainInc(i,1)
       stressNew(i,2) = (xk1+xk2)*strainInc(i,2)
       stressNew(i,3) = (xk1+xk2)*strainInc(i,3)
       stressNew(i,4) = (xk1+xk2)*strainInc(i,4)
       stressNew(i,5) = (xk1+xk2)*strainInc(i,5)
       stressNew(i,6) = (xk1+xk2)*strainInc(i,6)
     else
       stressNew(i,1) = sigE1+sigM1_1
       stressNew(i,2) = sigE2+sigM1_2
       stressNew(i,3) = sigE3+sigM1_3
       stressNew(i,4) = sigE4+sigM1_4
       stressNew(i,5) = sigE5+sigM1_5
       stressNew(i,6) = sigE6+sigM1_6
     end if
C
       stateNew(i,1)=stateOld(i,1)+strainInc(i,1)
       stateNew(i,2)=stateOld(i,2)+strainInc(i,2)
       stateNew(i,3)=stateOld(i,3)+strainInc(i,3)
       stateNew(i,4)=stateOld(i,4)+strainInc(i,4)
       stateNew(i,5)=stateOld(i,5)+strainInc(i,5)
       stateNew(i,6)=stateOld(i,6)+strainInc(i,6)
C
       stateNew(i,7)=sigM1_1
       stateNew(i,8)=sigM1_2
       stateNew(i,9)=sigM1_3
       stateNew(i,10)=sigM1_4
       stateNew(i,11)=sigM1_5
       stateNew(i,12)=sigM1_6
   100 continue

     return
     end
```

Appendix C

Full Results of Simulations

This appendix shows the full size graphs that were presented in Chapter 11.

C.1 LS Dyna Simulations Verification

C.1.1 No Hourglassing control

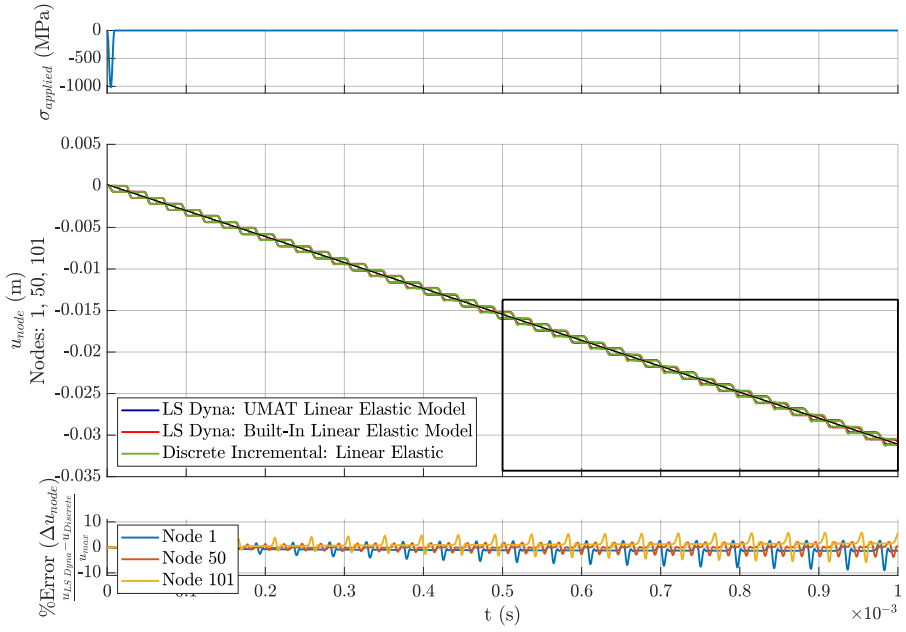


Figure C.1: Linear Elastic Model - LS Dyna vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Off

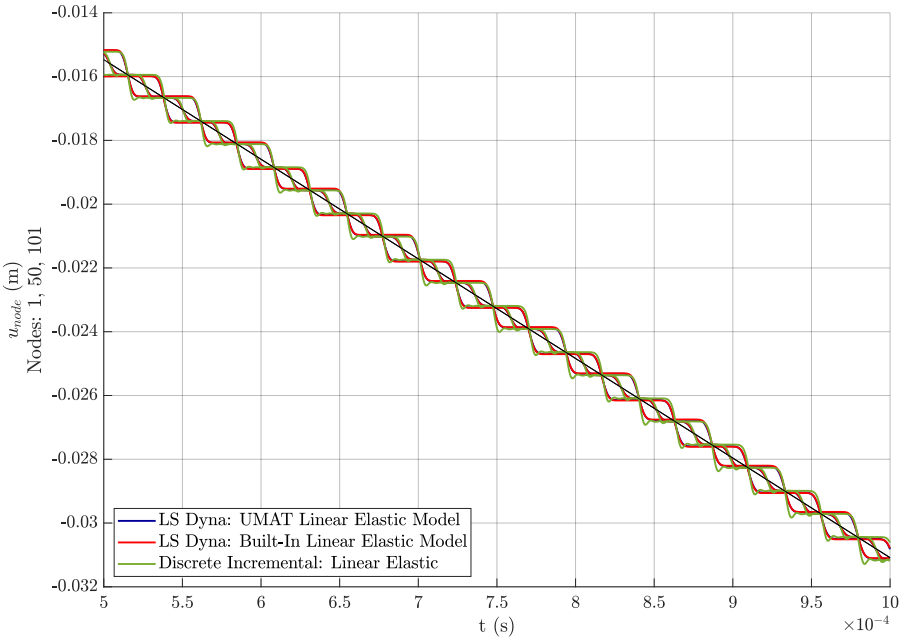


Figure C.2: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Off

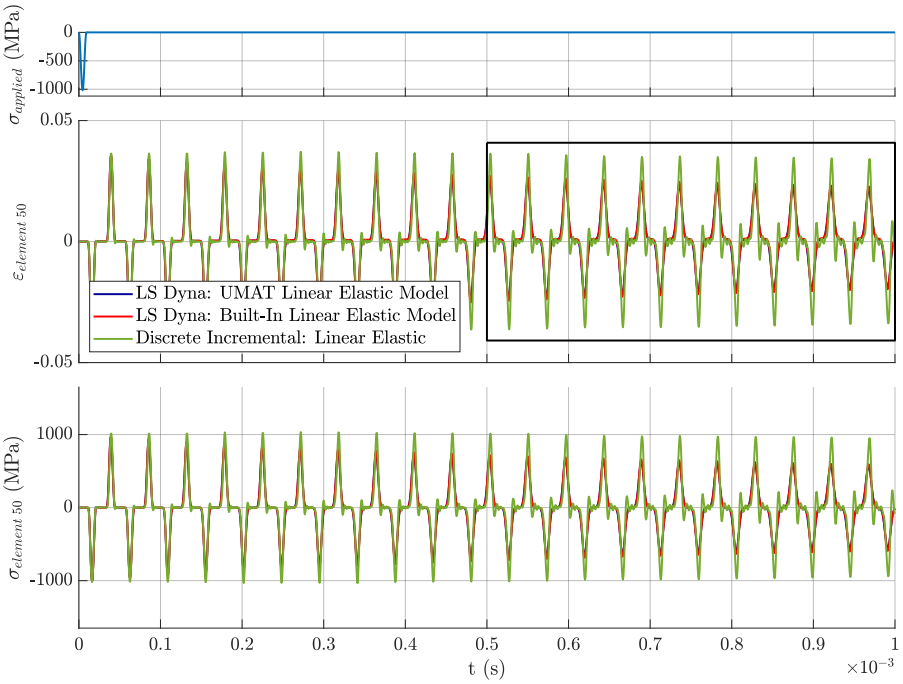


Figure C.3: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Off

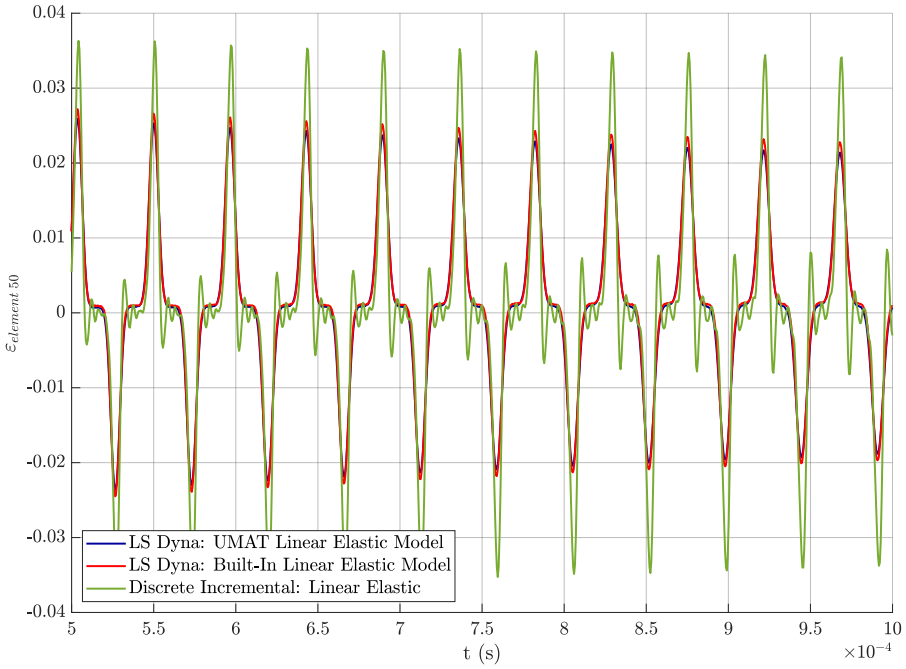


Figure C.4: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Off

C.1.2 Default Hourglassing Control $Q_2 = 6 \times 10^{-2}$

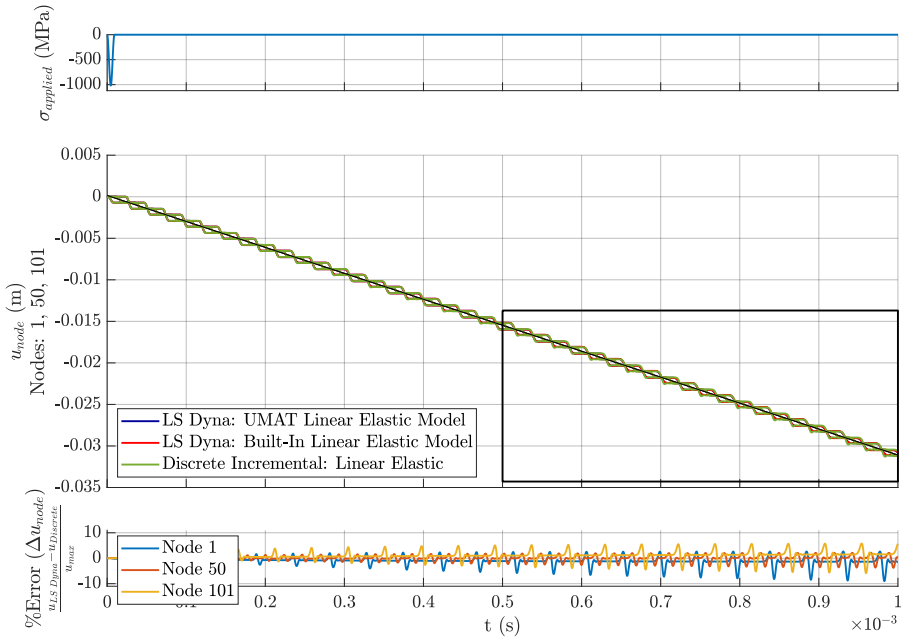


Figure C.5: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-2}$

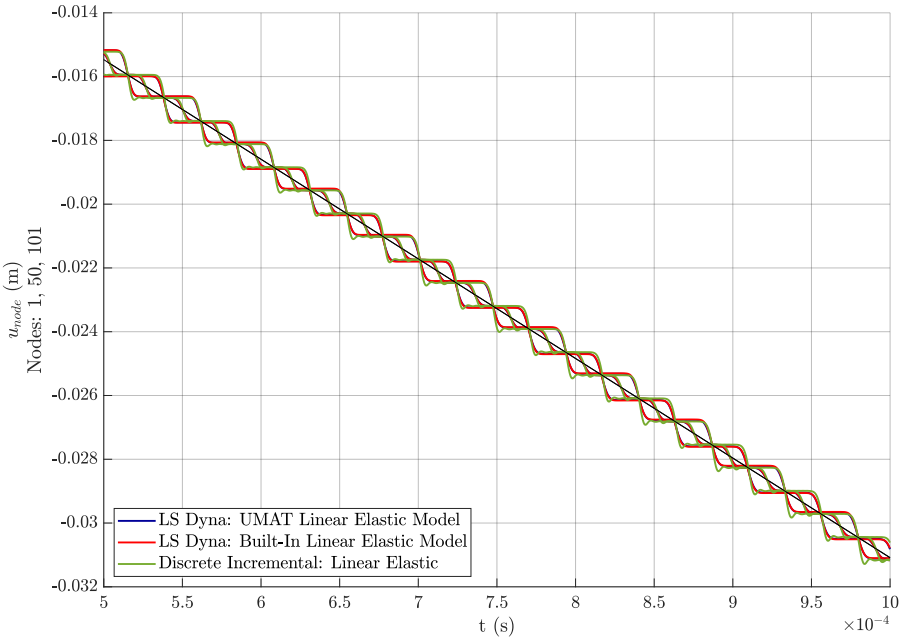


Figure C.6: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-2}$

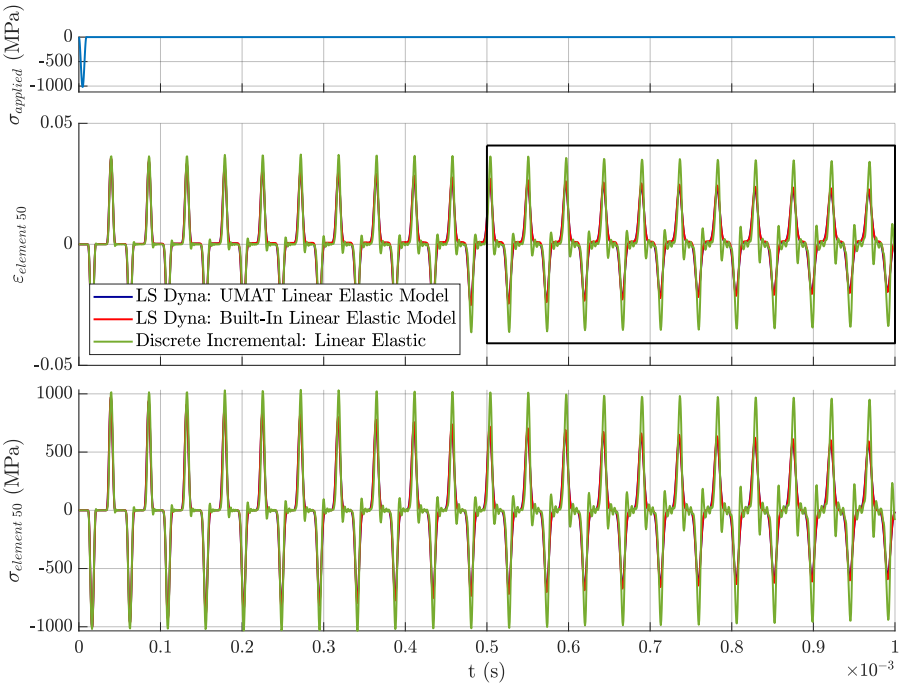


Figure C.7: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-2}$

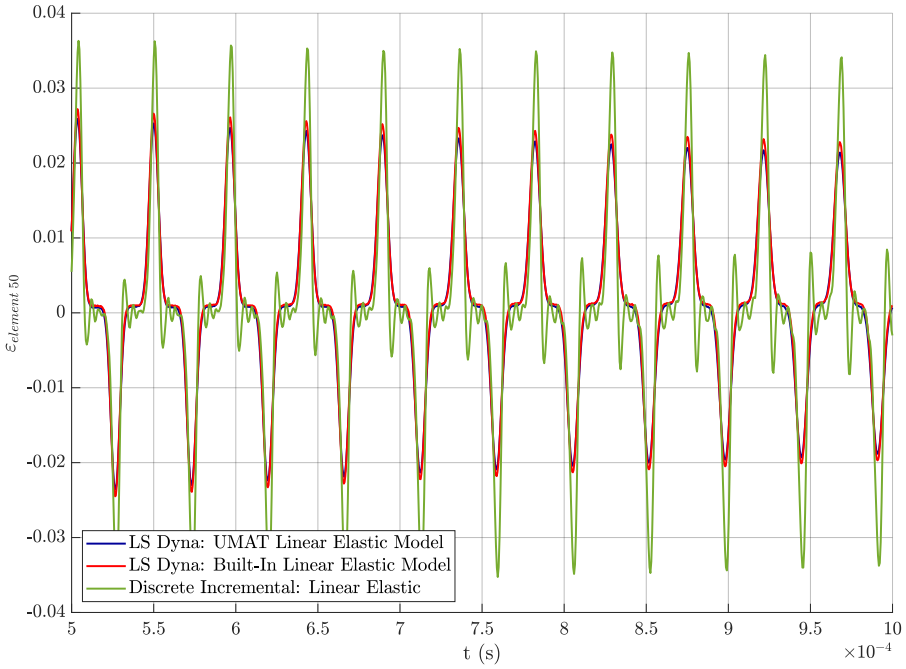


Figure C.8: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-2}$

C.1.3 Hourglassing Control $Q_2 = 6 \times 10^{-4}$

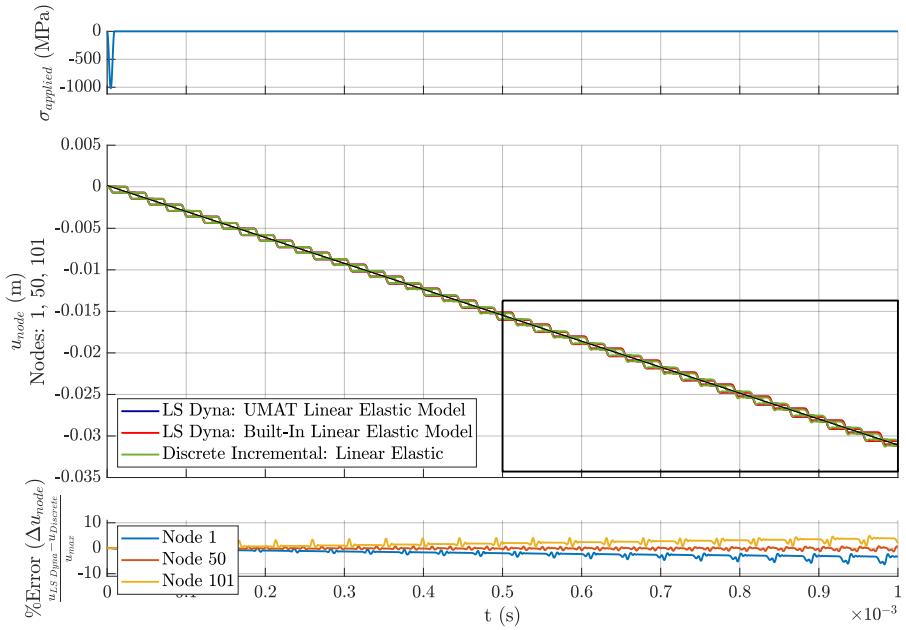


Figure C.9: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

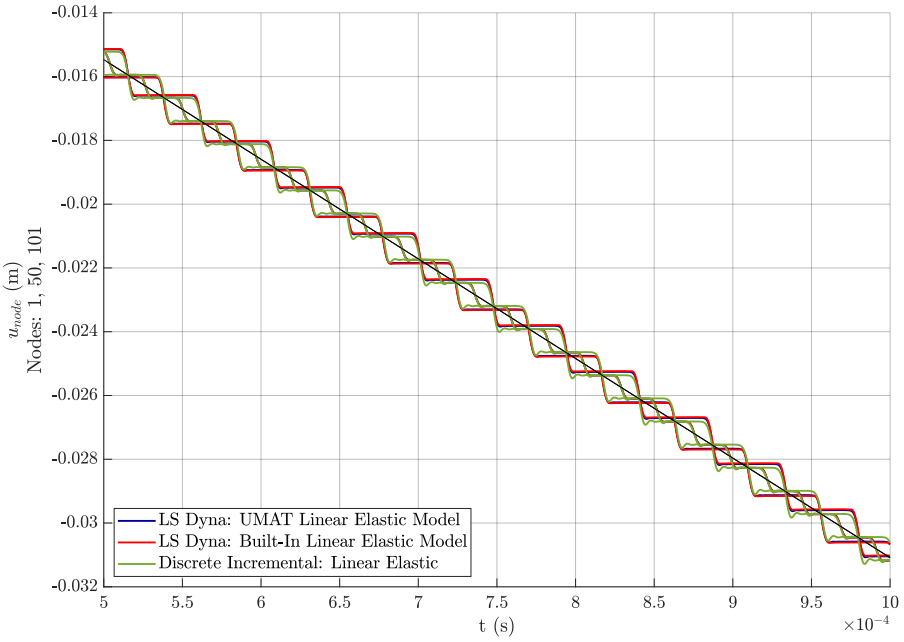


Figure C.10: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

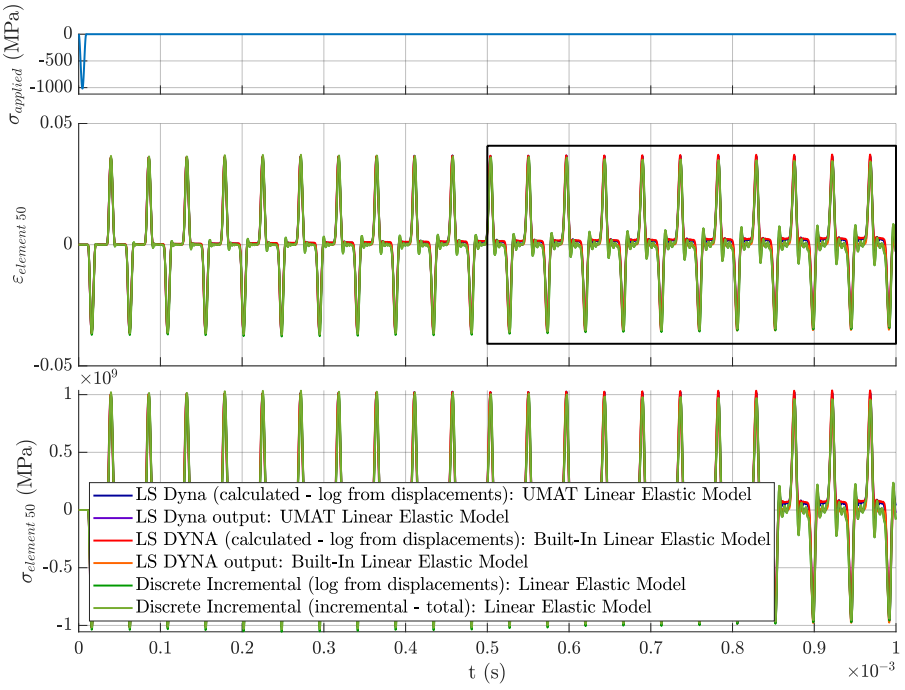


Figure C.11: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

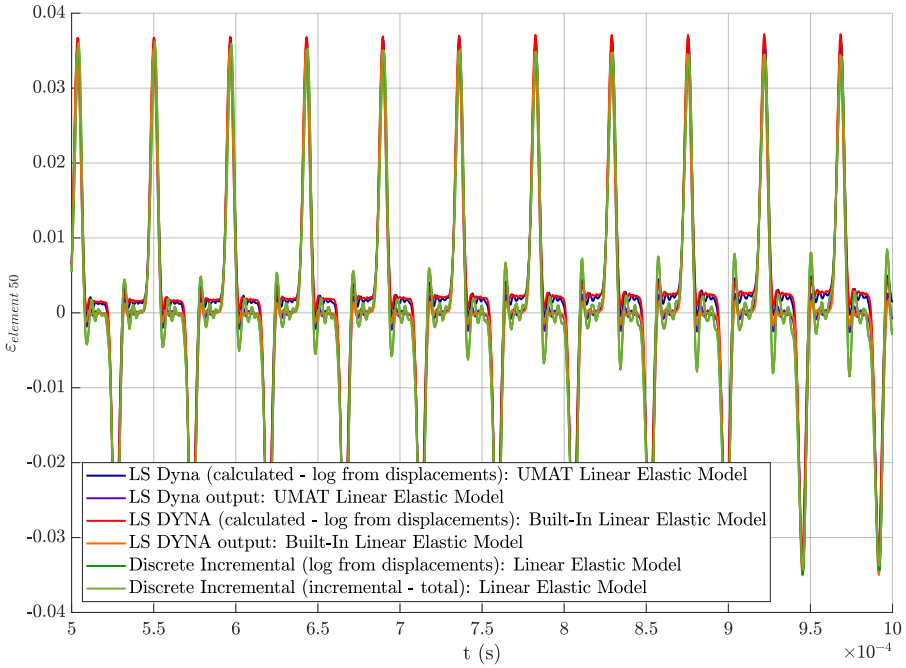


Figure C.12: Linear Elastic Model - LS Dyna Built-In vs UMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

C.2 LS-Dyna vs Abaqus Simulations Validation

C.2.1 Built-In Linear Elastic Models, Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

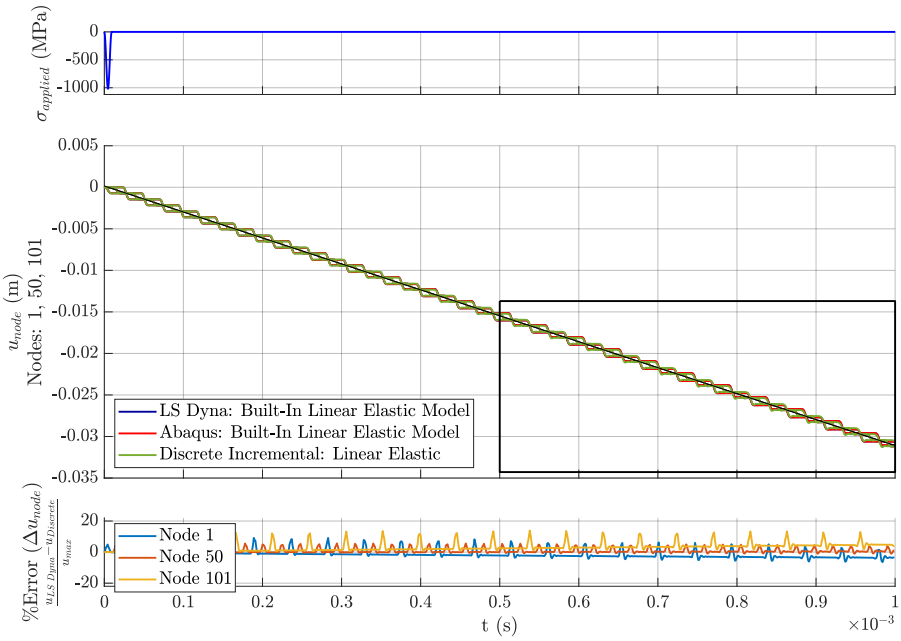


Figure C.13: Linear Elastic Model - LS Dyna vs Abaqus vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

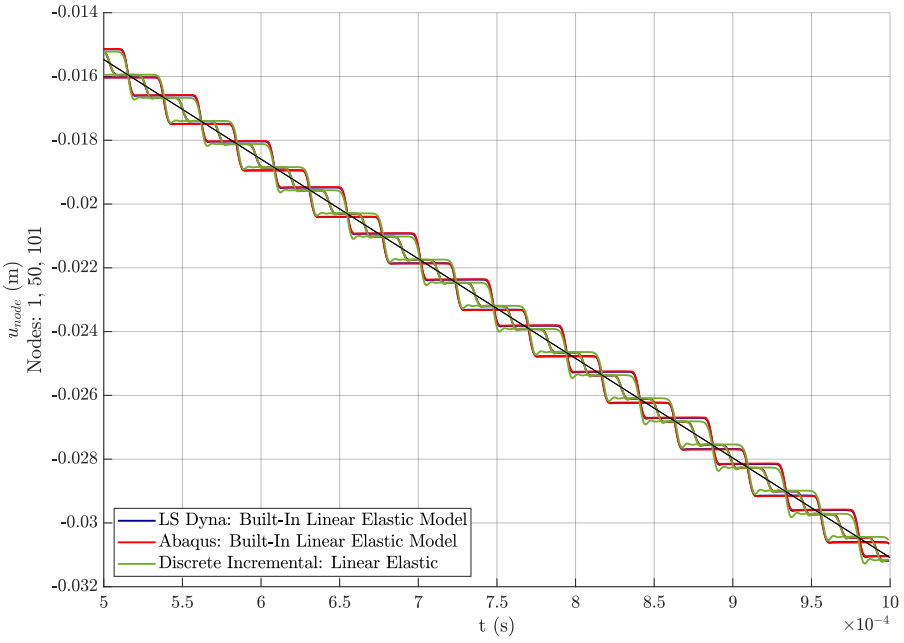


Figure C.14: Linear Elastic Model - LS Dyna Built-In vs Abaqus Built-In vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

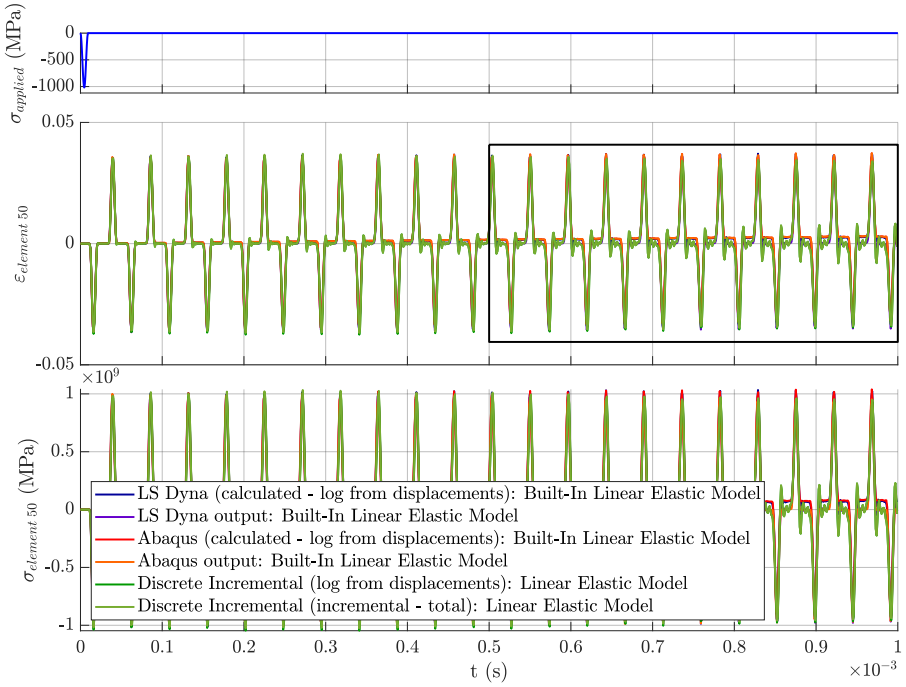


Figure C.15: Linear Elastic Model - LS Dyna Built-In vs Abaqus Built-In vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

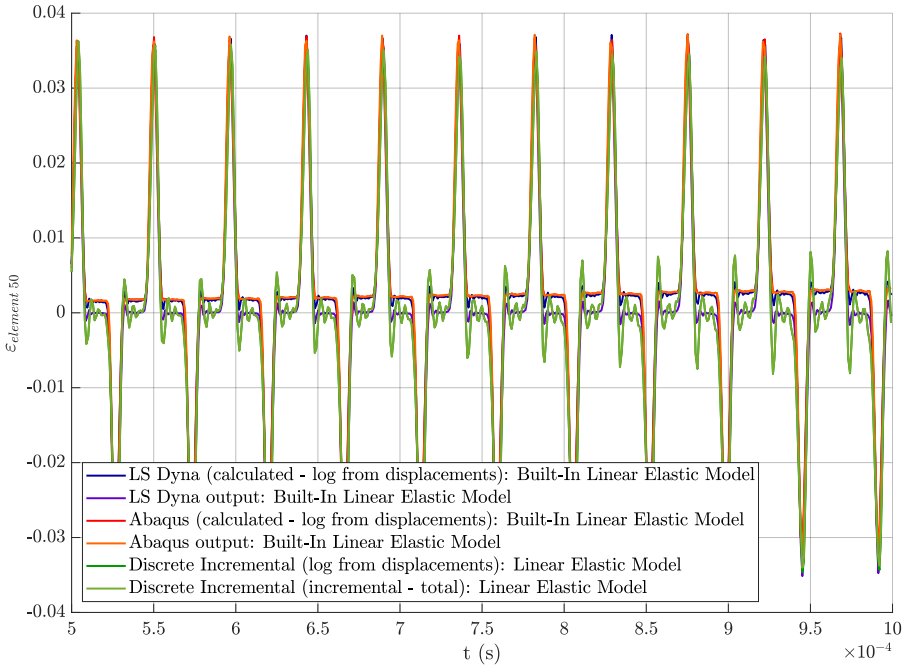


Figure C.16: Linear Elastic Model - LS Dyna Built-In vs Abaqus Built-In vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

C.2.2 User-Defined Linear Elastic Models, Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

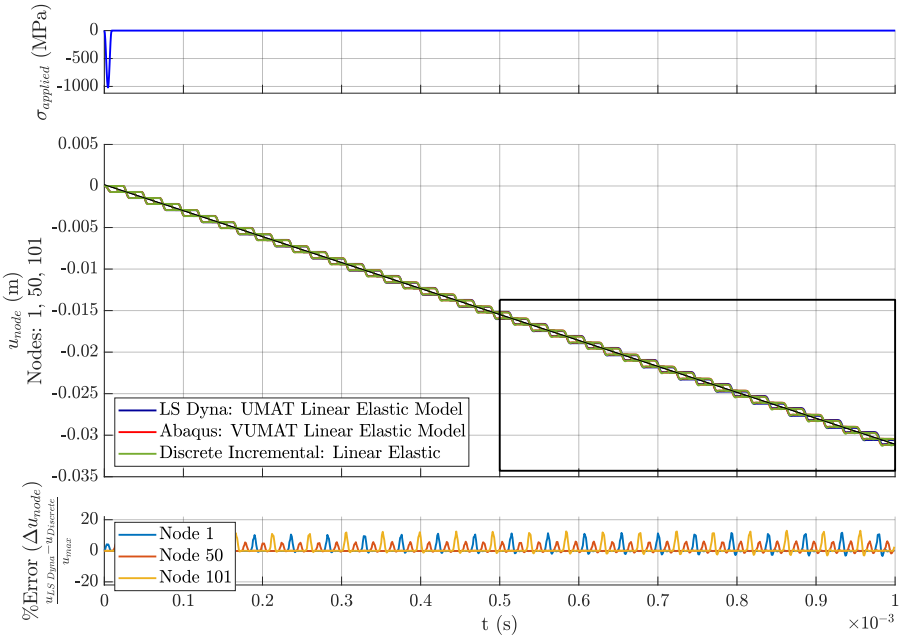


Figure C.17: Linear Elastic Model - LS Dyna UMAT vs Abaqus VUMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

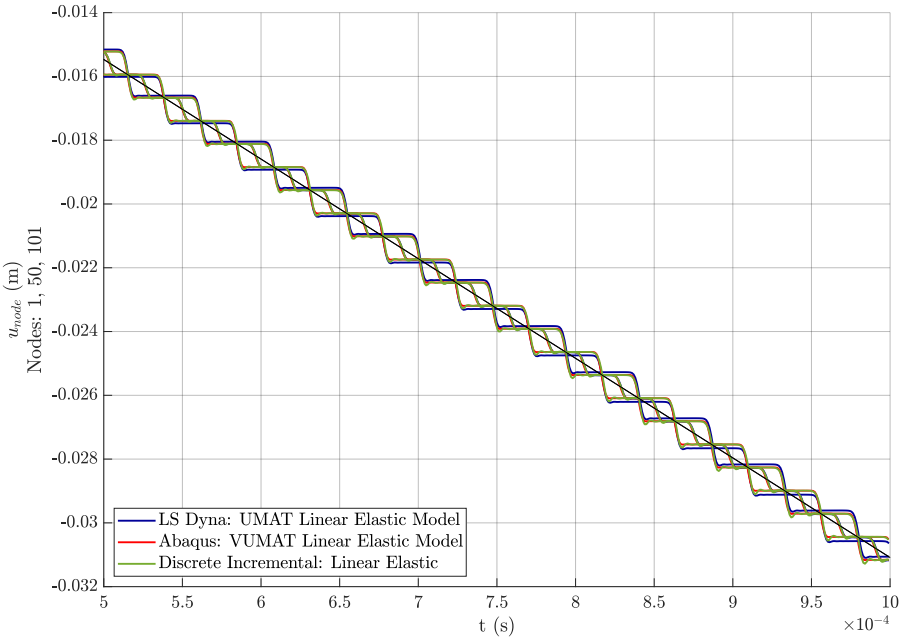


Figure C.18: Linear Elastic Model - LS Dyna UMAT vs Abaqus VUMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

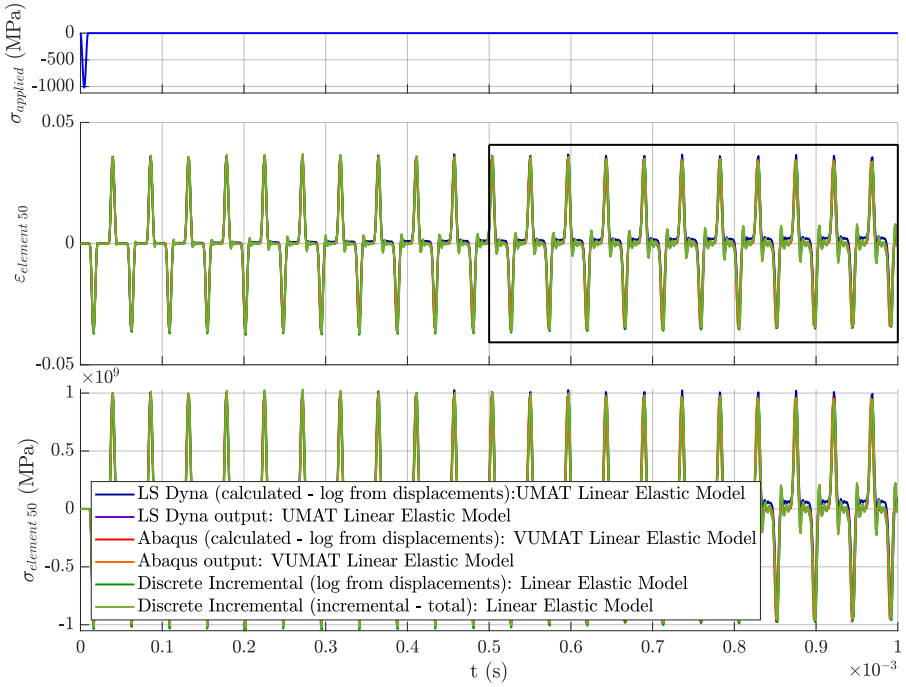


Figure C.19: Linear Elastic Model - LS Dyna UMAT vs Abaqus VUMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

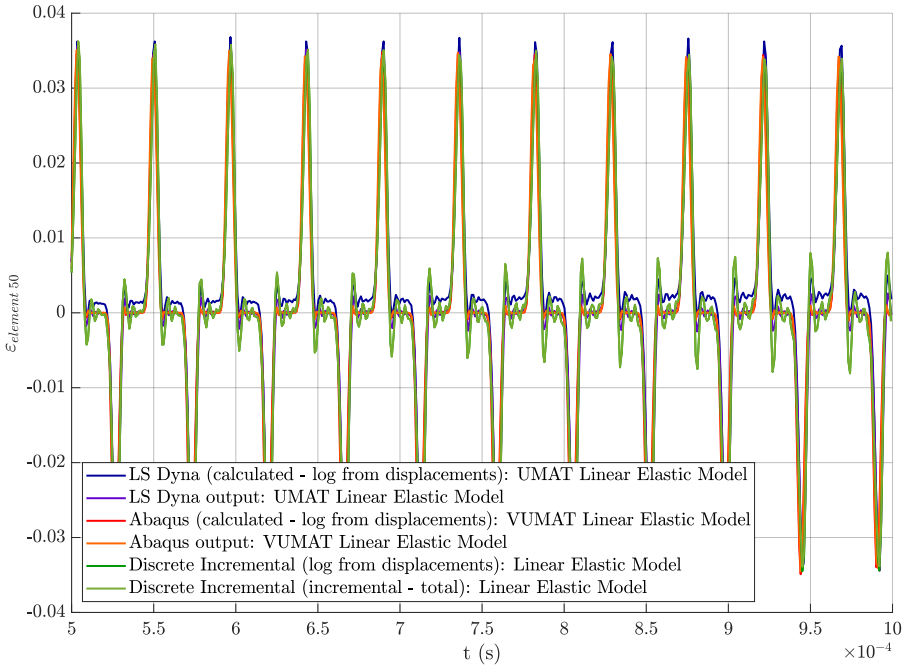


Figure C.20: Linear Elastic Model - LS Dyna UMAT vs Abaqus VUMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

C.3 Abaqus Simulations Validation

C.3.1 Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

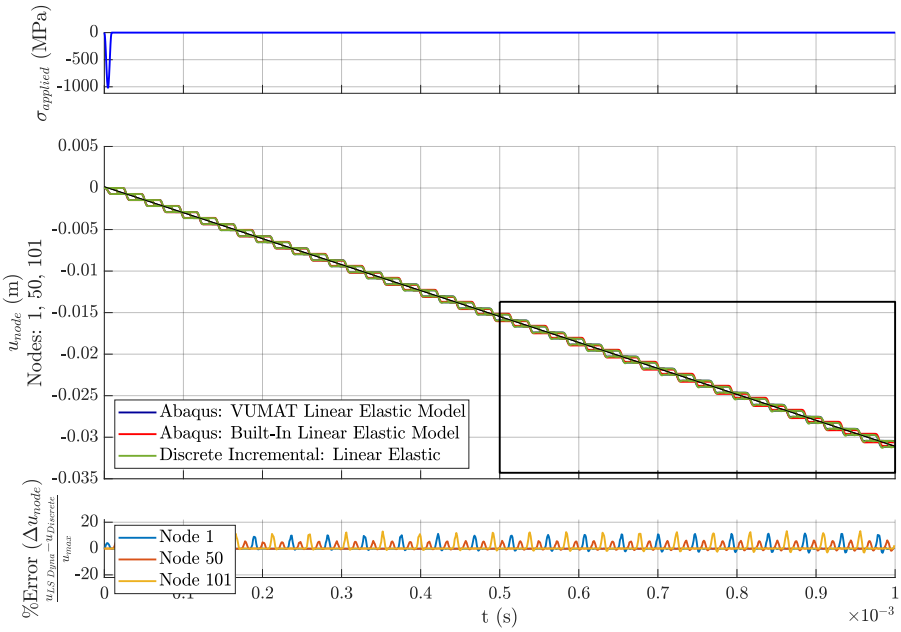


Figure C.21: Linear Elastic Model - Abaqus Built-In vs VUMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

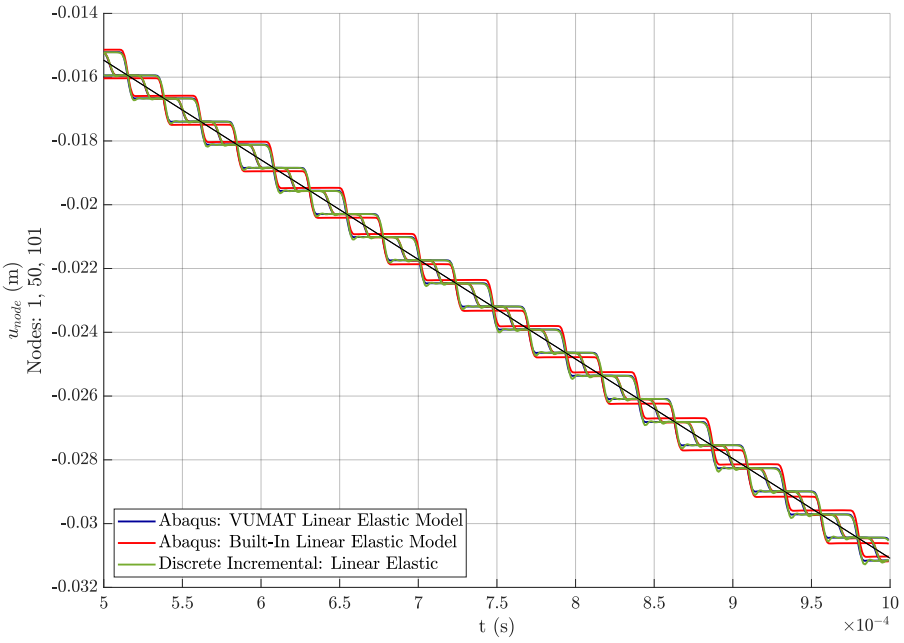


Figure C.22: Linear Elastic Model - Abaqus Built-In vs VUMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

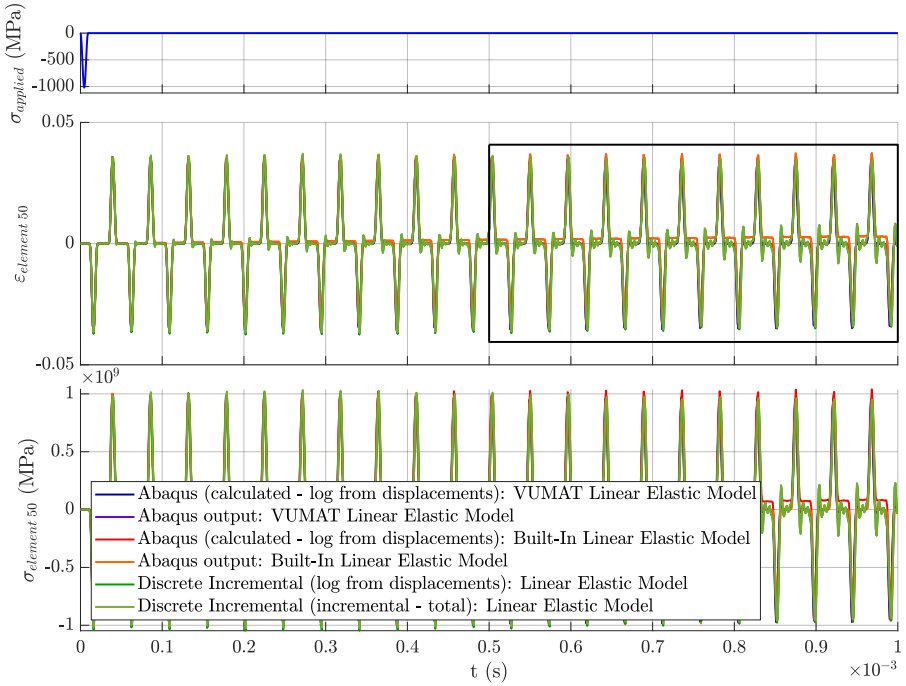


Figure C.23: Linear Elastic Model - Abaqus Built-In vs VUMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

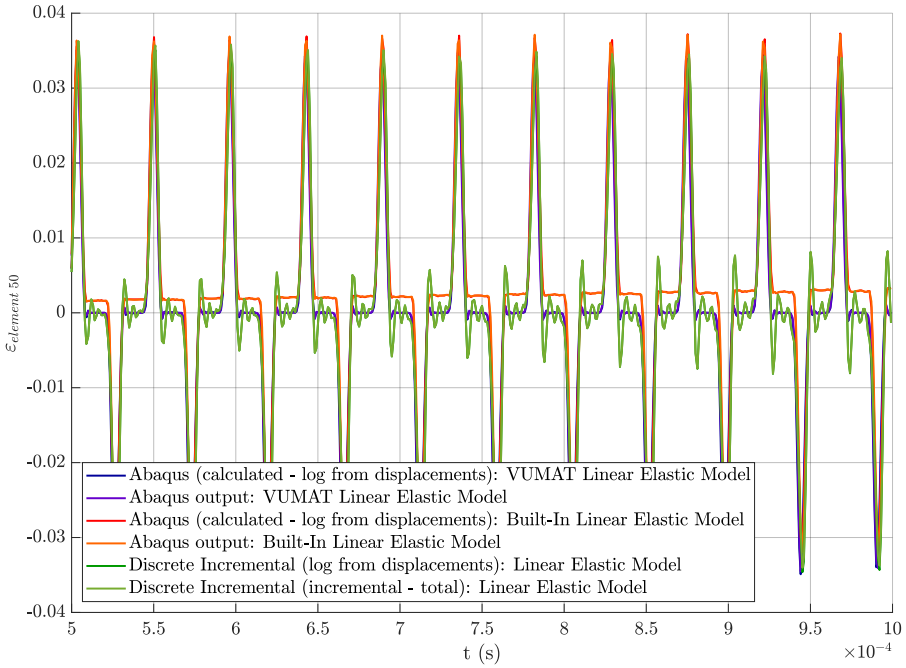


Figure C.24: Linear Elastic Model - Abaqus Built-In vs VUMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

C.4 Hereditary Integral vs FEA Simulations

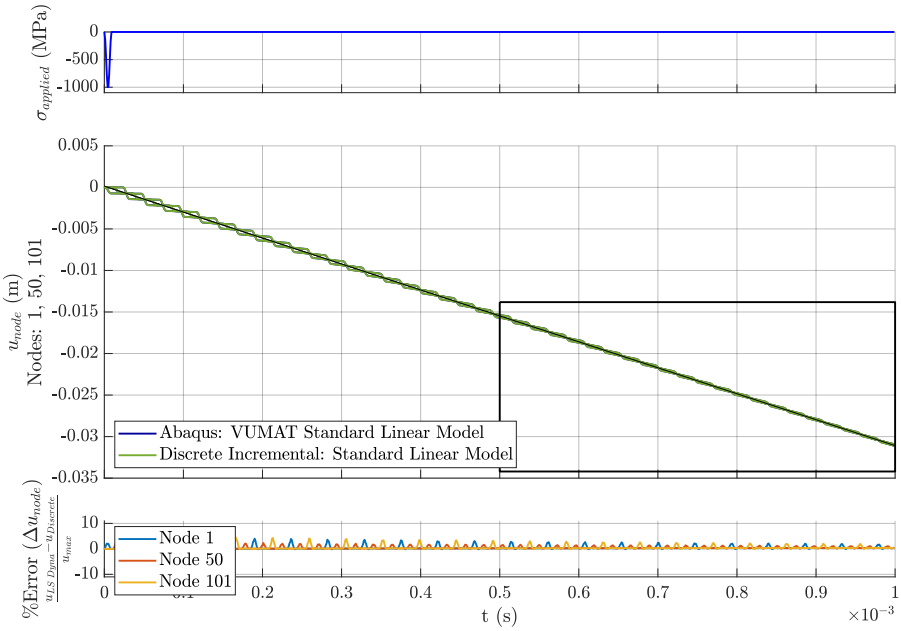


Figure C.25: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Displacement-Time graph - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

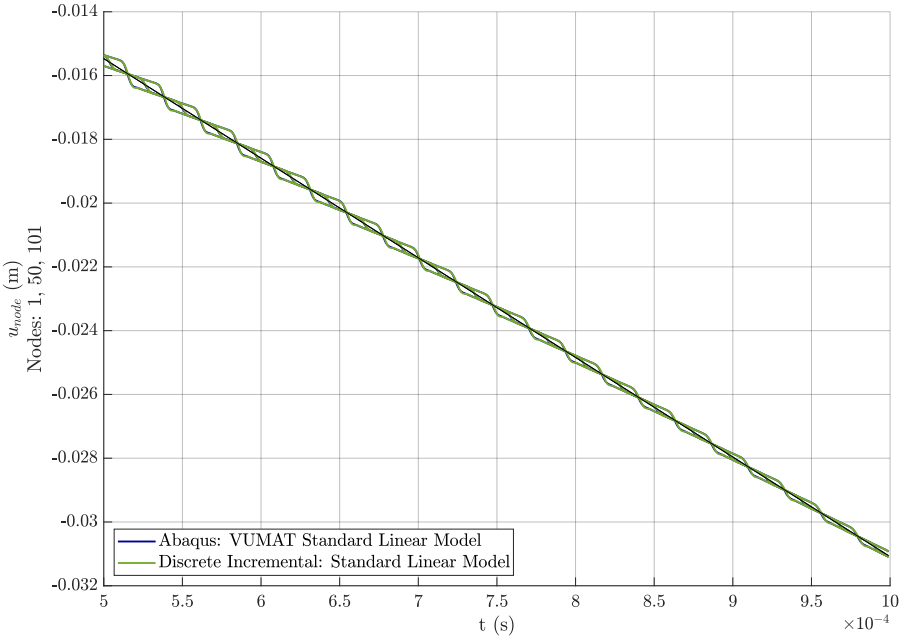


Figure C.26: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Displacement-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

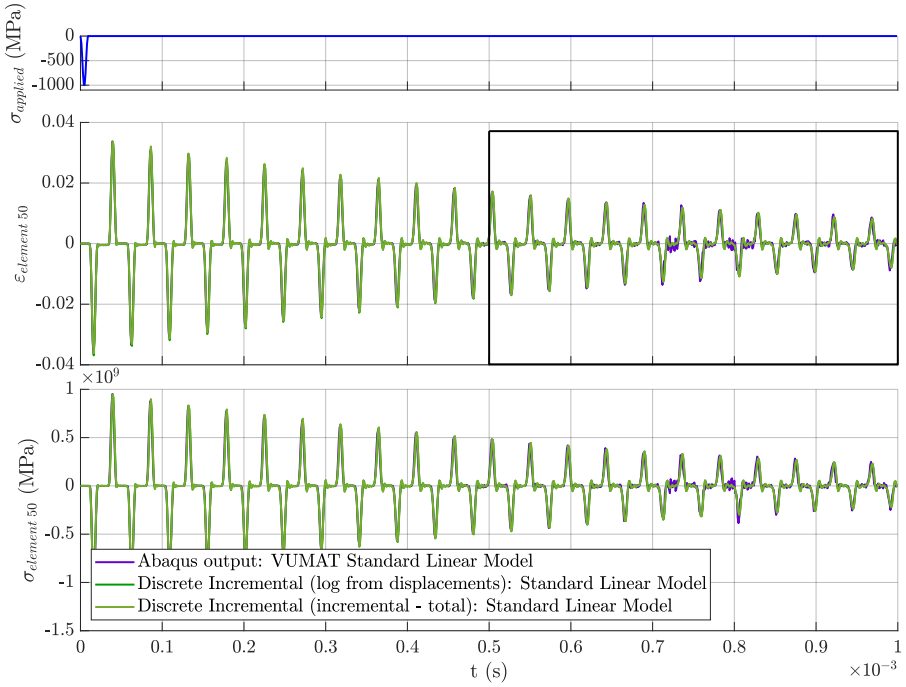


Figure C.27: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Stress-Time and Strain-Time graphs - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

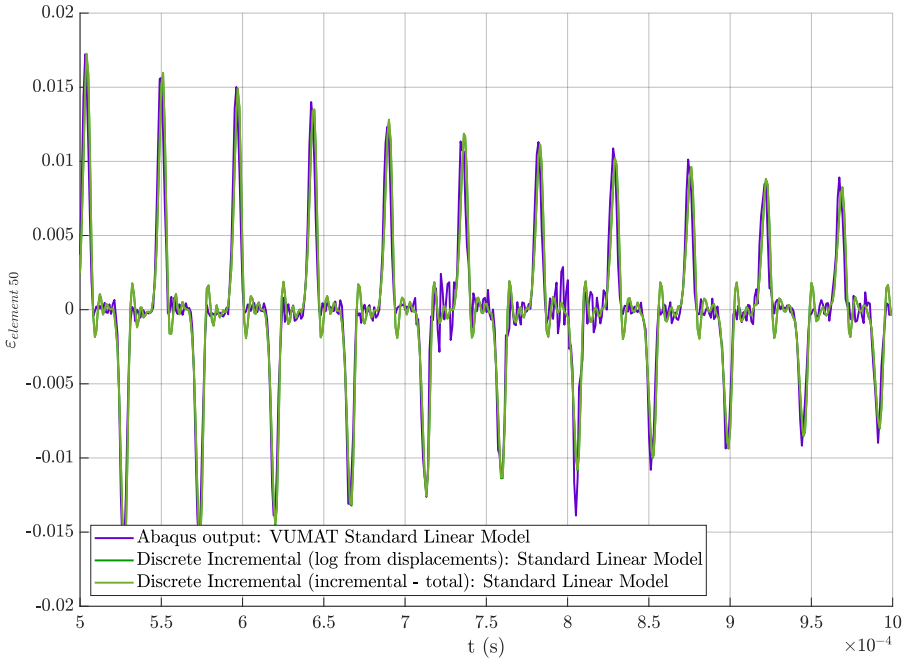


Figure C.28: Standard Linear Model - Abaqus VUMAT vs multi element hereditary integral - Strain-Time graph (zoomed in) - Hourglassing Control Parameter $Q_2 = 6 \times 10^{-4}$

C.5 Abaqus vs Experimental Data

C.5.1 Simplified Material Model for Cortical Bone

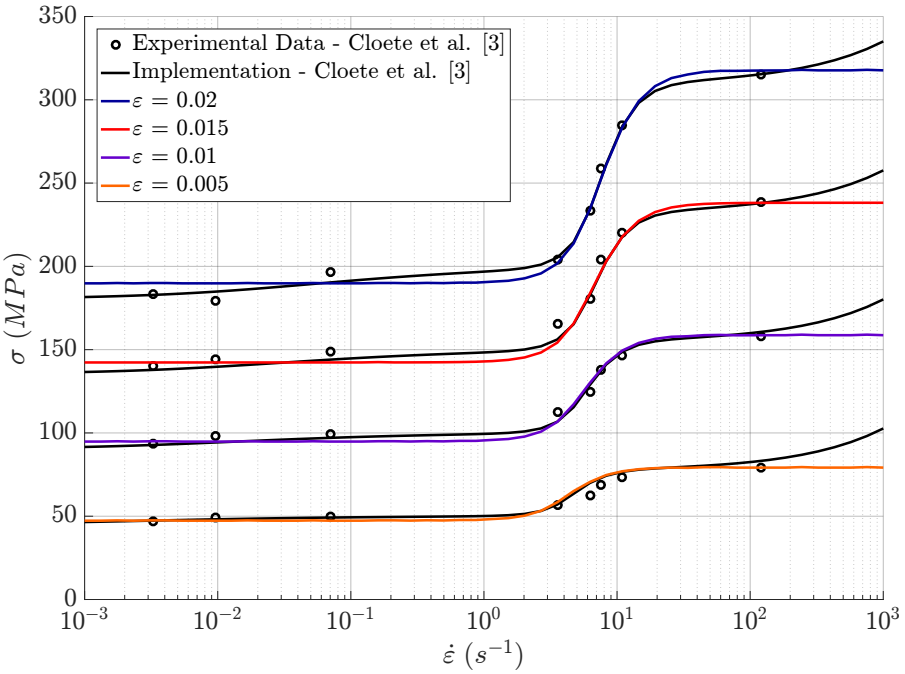


Figure C.29: Cortical Bone Model - Baseline - $p = 2.5$ - Stress - Strain Rate graph

C.5.2 Simulation Parameters

The simulation parameters are outlined in Table C.1 for the various strain rates that were used.

Table C.1: Abaqus Simulations Parameters

Strain rate $\dot{\varepsilon}$ (s^{-1})	Mass Scaling Factor	Run Time t_{tot} (s)	Sampling Time t_{samp} (s)
1.000×10^{0}	10	0.04000000	4.000×10^{-5}
1.500×10^{0}	10	0.02666670	2.667×10^{-5}
2.000×10^{0}	8	0.02000000	2.000×10^{-5}
2.800×10^{0}	8	0.01428550	1.429×10^{-5}
3.610×10^{0}	7	0.01108000	1.108×10^{-5}
4.950×10^{0}	7	0.00808100	8.081×10^{-6}
6.330×10^{0}	5	0.00631900	6.319×10^{-6}
6.900×10^{0}	5	0.00579700	5.797×10^{-6}
7.400×10^{0}	5	0.00540550	5.406×10^{-6}
7.580×10^{0}	3	0.00527700	5.277×10^{-6}
7.700×10^{0}	3	0.00519500	5.195×10^{-6}
8.800×10^{0}	3	0.00454545	4.545×10^{-6}
9.800×10^{0}	3	0.00408165	4.082×10^{-6}
1.000×10^{1}	1	0.00400000	4.000×10^{-6}
1.086×10^{1}	1	0.00368300	3.683×10^{-6}
1.100×10^{1}	1	0.00363635	3.636×10^{-6}
1.600×10^{1}	1	0.00250000	2.500×10^{-6}
2.000×10^{1}	1	0.00200000	2.000×10^{-6}
3.500×10^{1}	1	0.00114285	1.143×10^{-6}
5.000×10^{1}	1	0.00080000	8.000×10^{-7}
1.000×10^{2}	1	0.00040000	4.000×10^{-7}
1.200×10^{2}	1	0.00033330	3.333×10^{-7}

C.5.3 Cortical Bone Model - Strain Rate in Each Direction

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

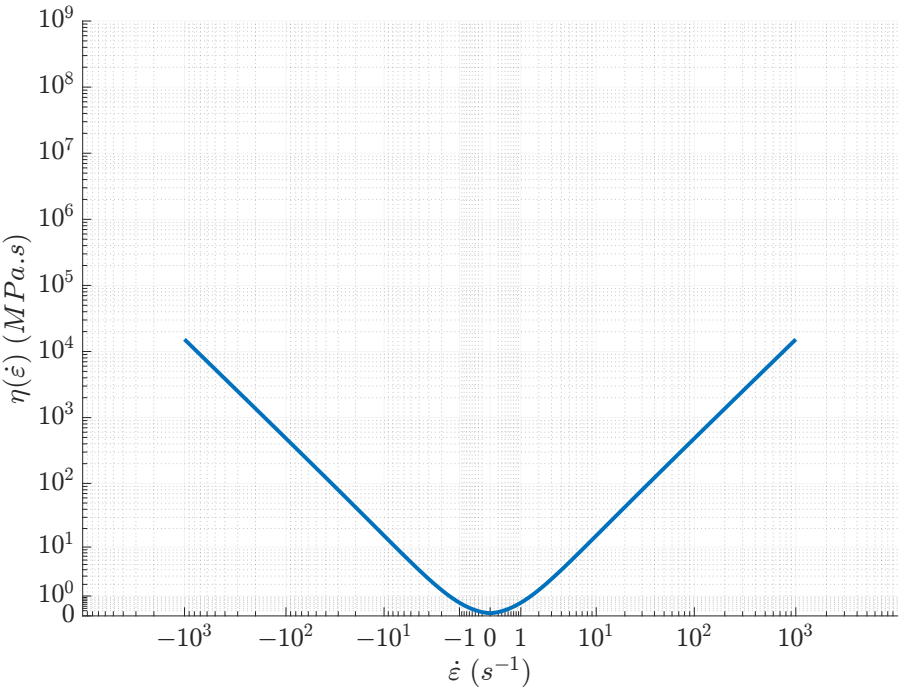


Figure C.30: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{ij})$ - $p = 2.5$ - Dashpot coefficient function

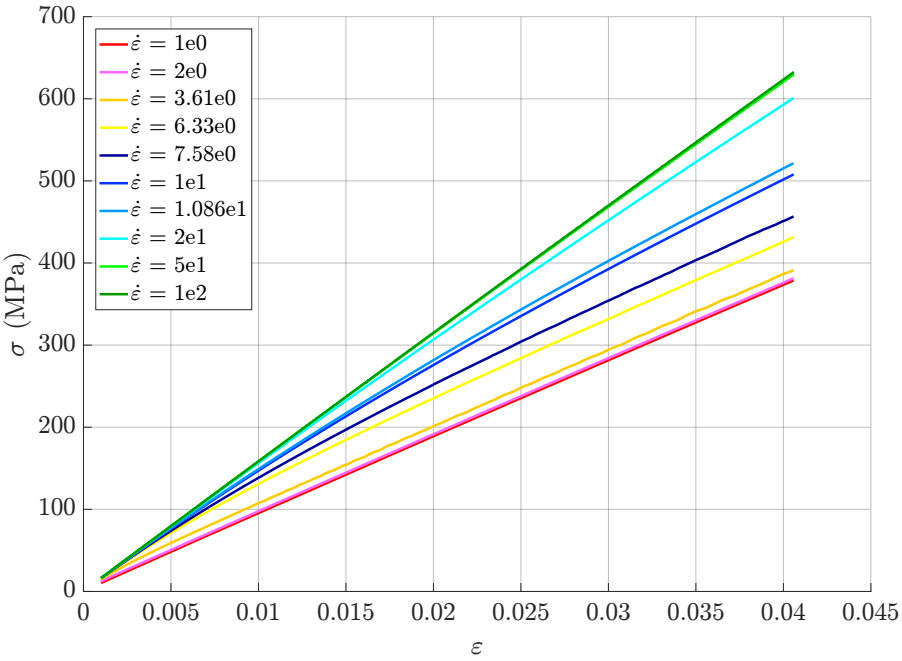


Figure C.31: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{ij})$ - Abaqus - $p = 2.5$ - Stress - Strain graph

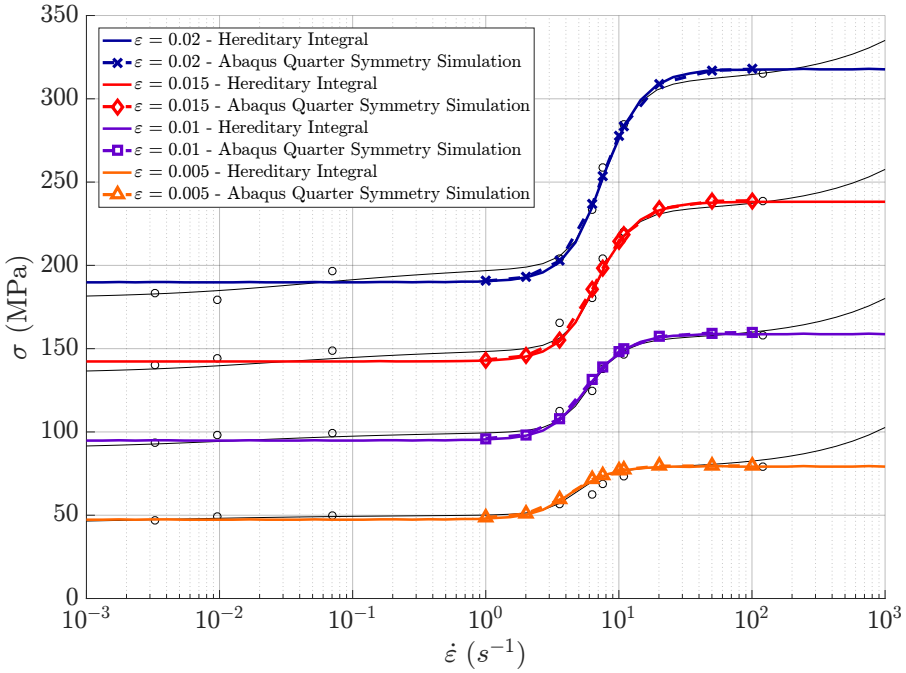


Figure C.32: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 1$, $\dot{\varepsilon}_{ref} = 0.35$)

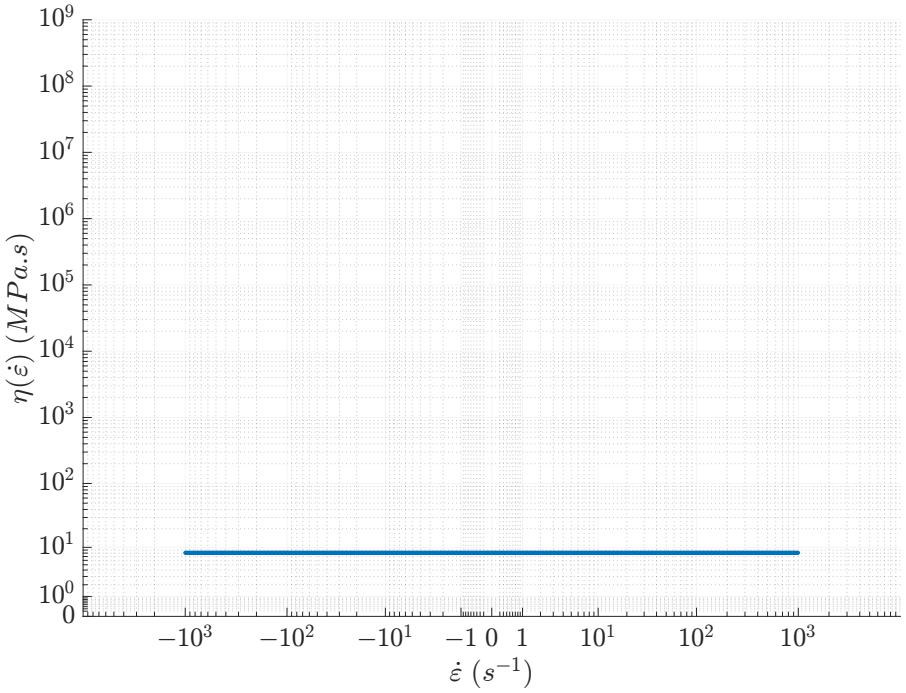


Figure C.33: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - $p = 1$ - Dashpot coefficient function

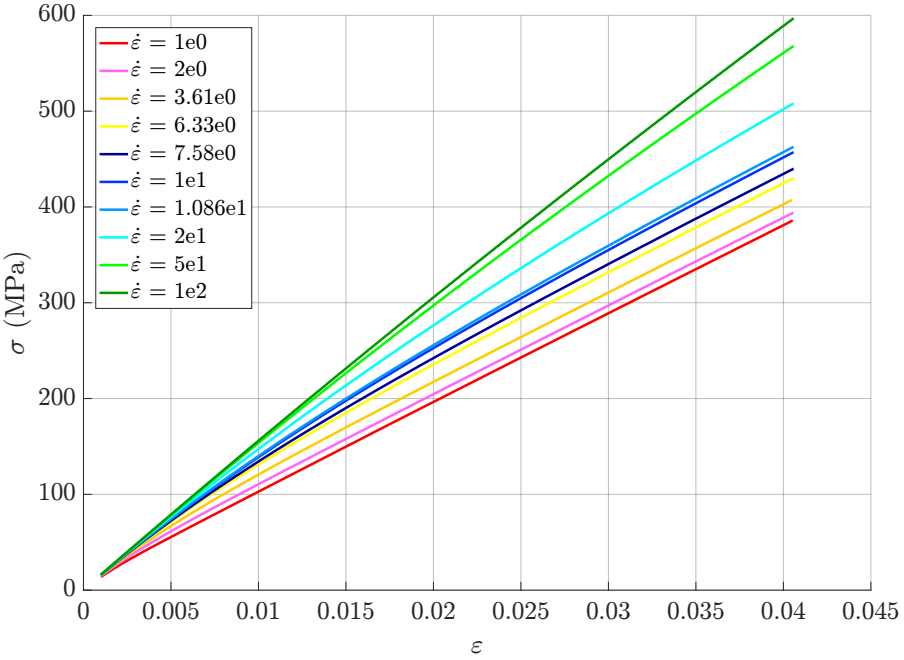


Figure C.34: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Abaqus - $p = 1$ - Stress - Strain graph

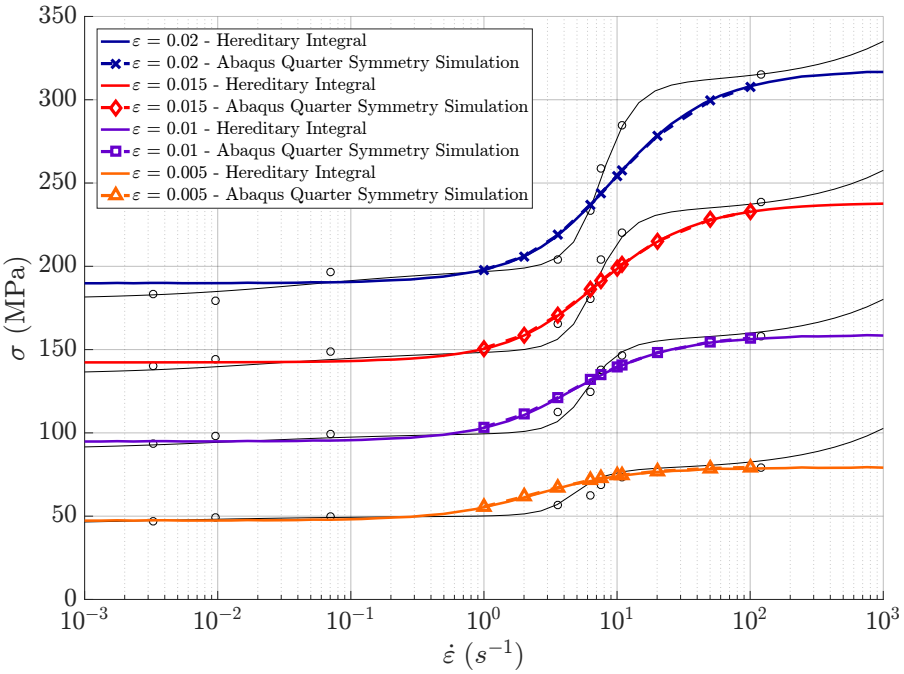


Figure C.35: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 1$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

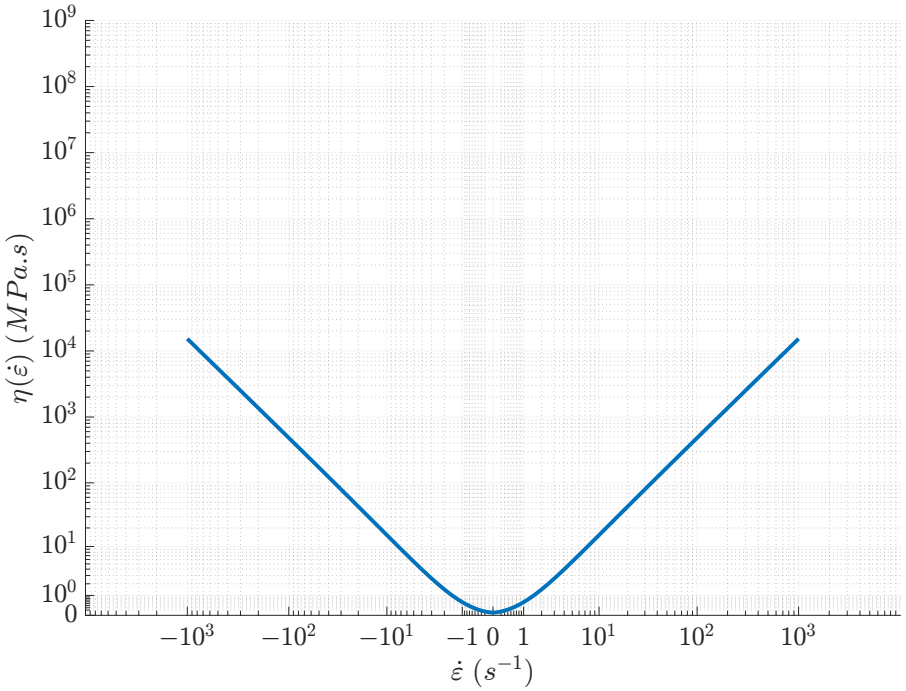


Figure C.36: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - $p = 2.5$ - Dashpot coefficient function

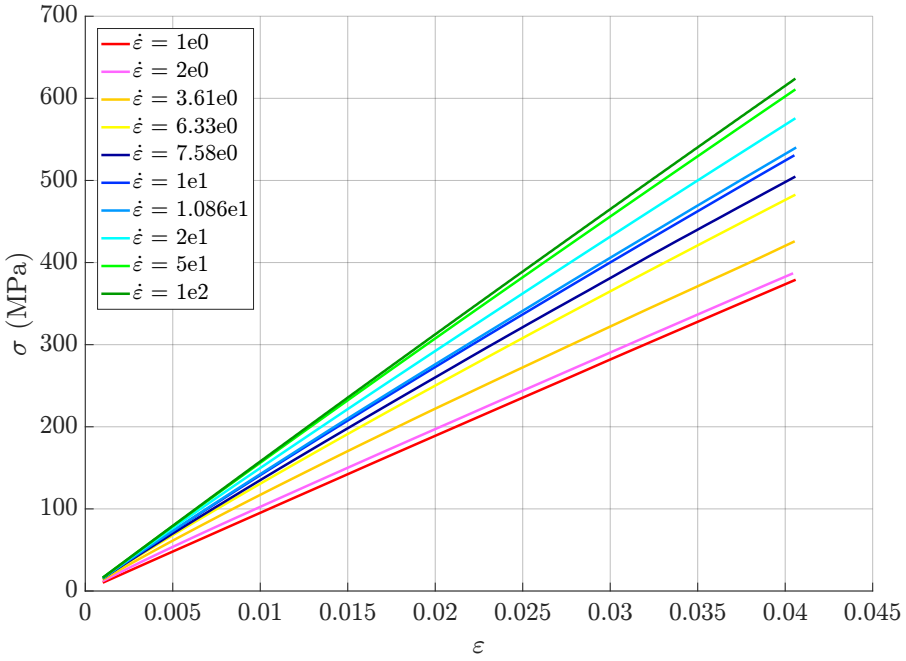


Figure C.37: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Abaqus - $p = 2.5$ - Stress - Strain graph

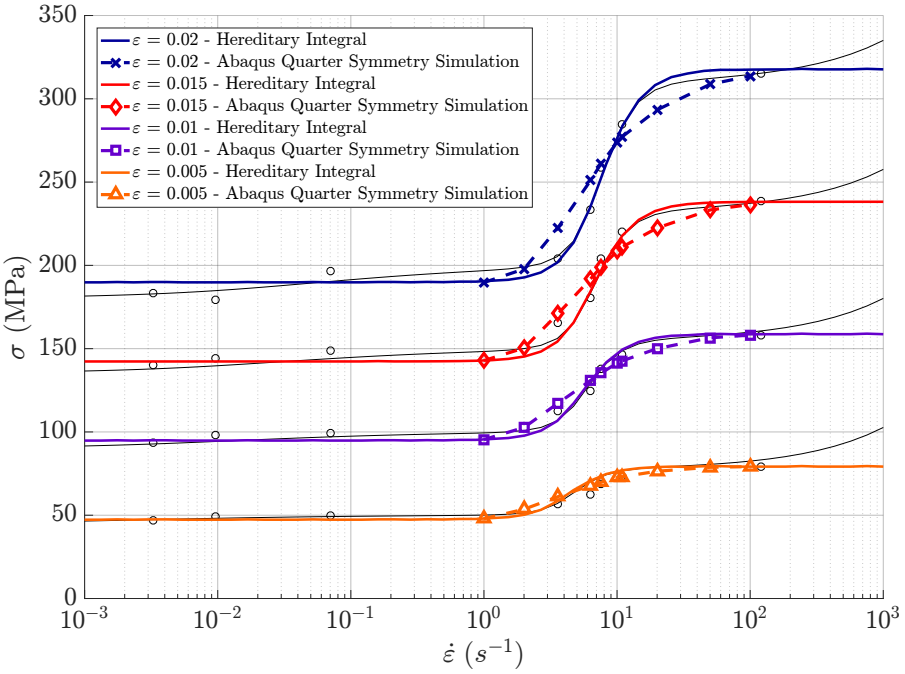


Figure C.38: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 4.5$, $\dot{\varepsilon}_{ref} = 0.35$)

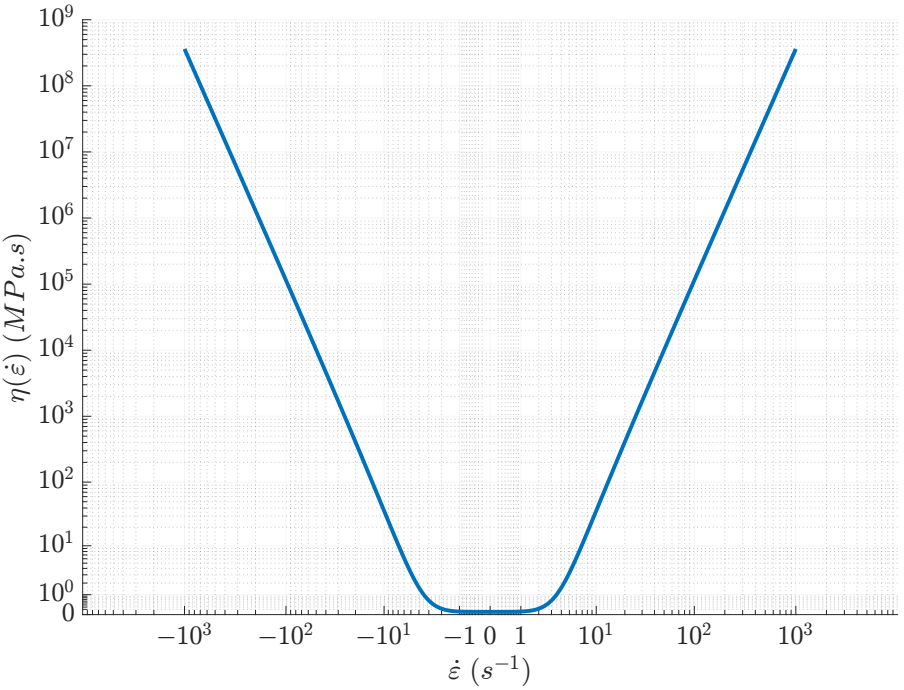


Figure C.39: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - $p = 4.5$ - Dashpot coefficient function

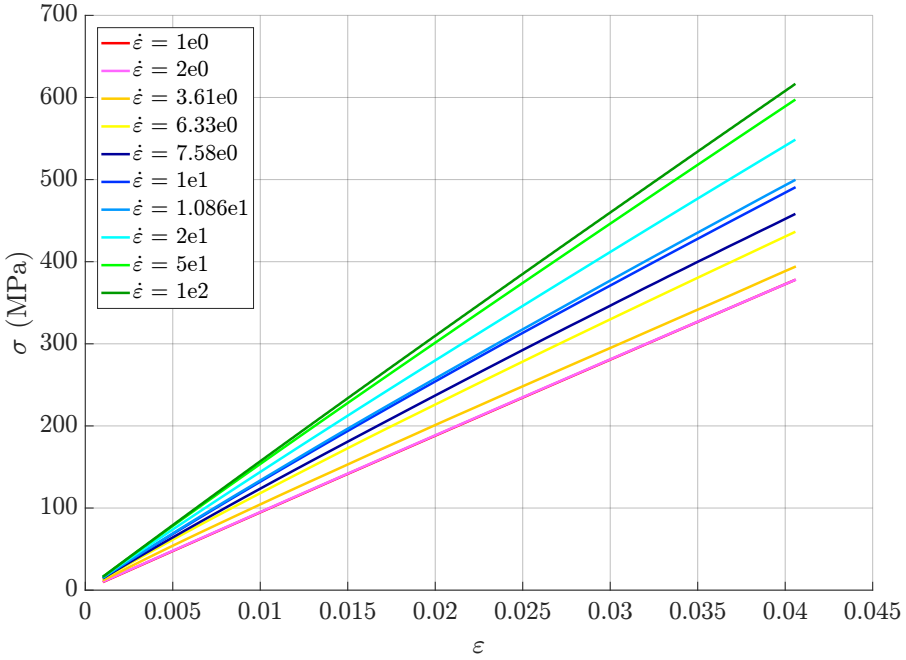


Figure C.40: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Abaqus - $p = 4.5$ - Stress - Strain graph

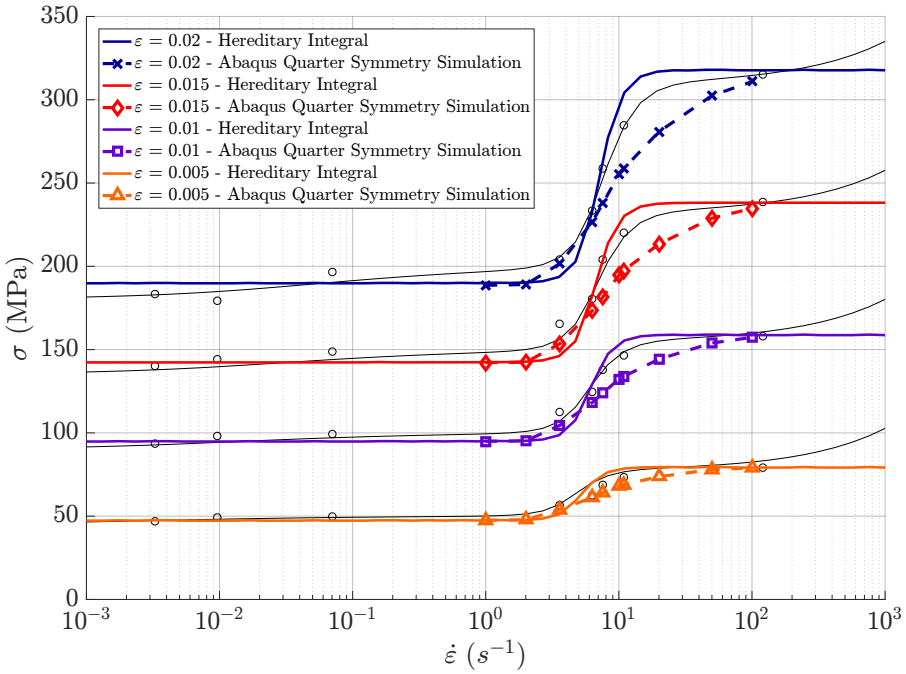


Figure C.41: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 4.5$ - Stress - Strain Rate graph

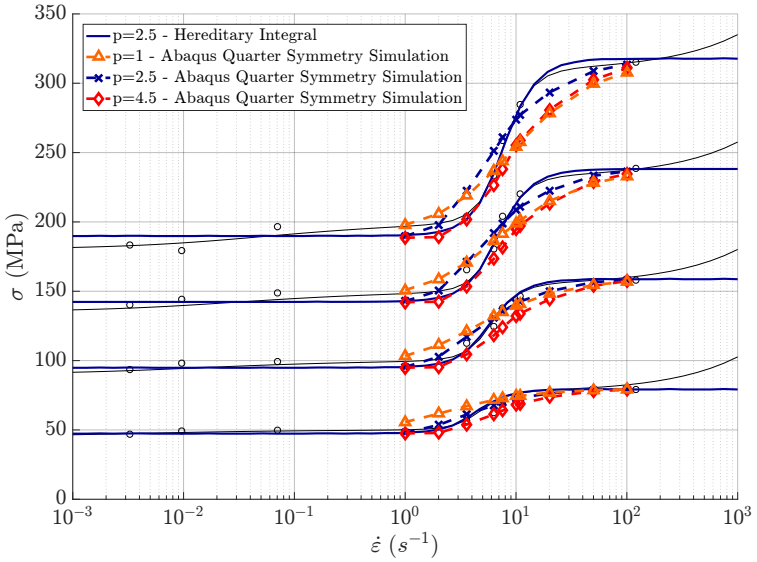


Figure C.42: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 1.1$ vs 2.5 vs 4.5 - Stress - Strain Rate graph

C.5.4 Cortical Bone Model - Dominant Strain Rate

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

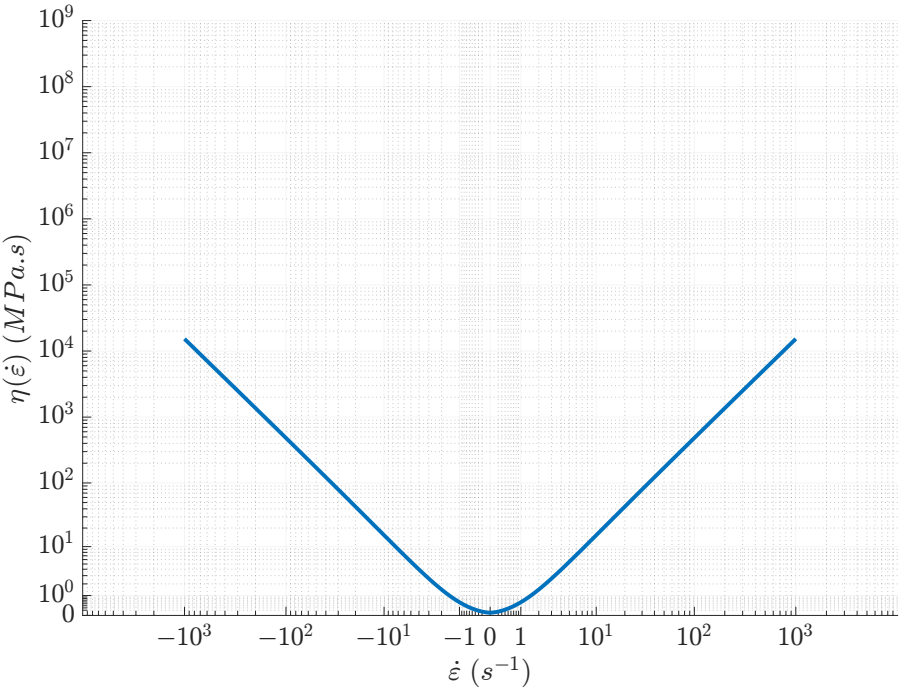


Figure C.43: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{11})$ - $p = 2.5$ - Dashpot coefficient function

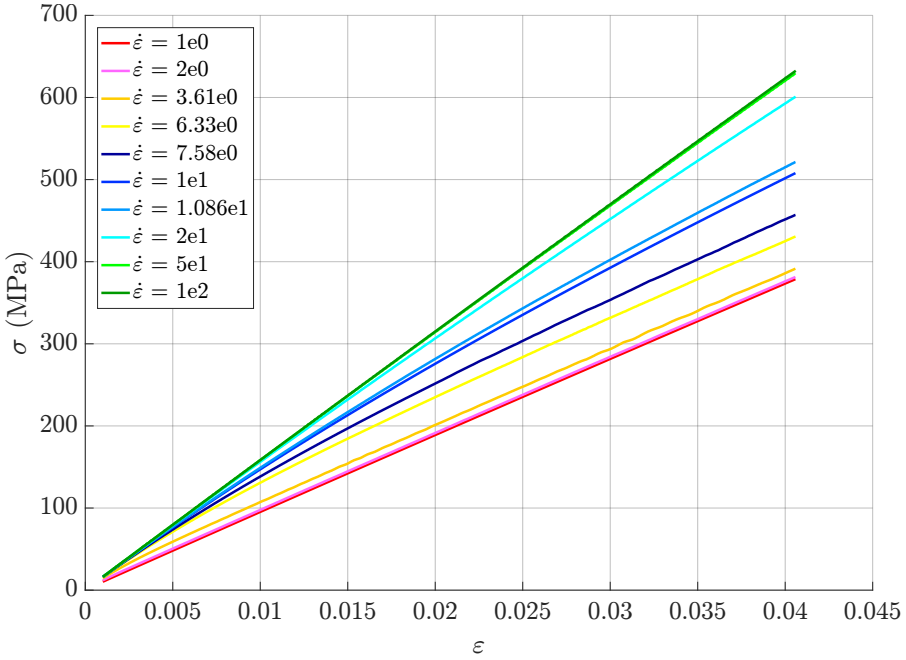


Figure C.44: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{11})$ - Abaqus - $p = 2.5$ - Stress - Strain graph

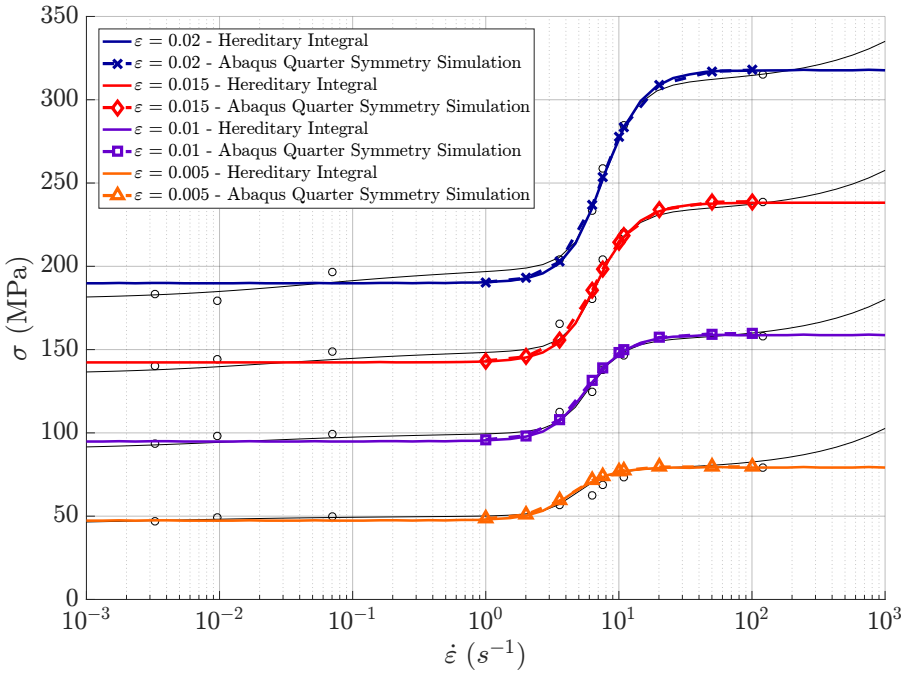


Figure C.45: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{11})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 1$, $\dot{\varepsilon}_{ref} = 0.35$)

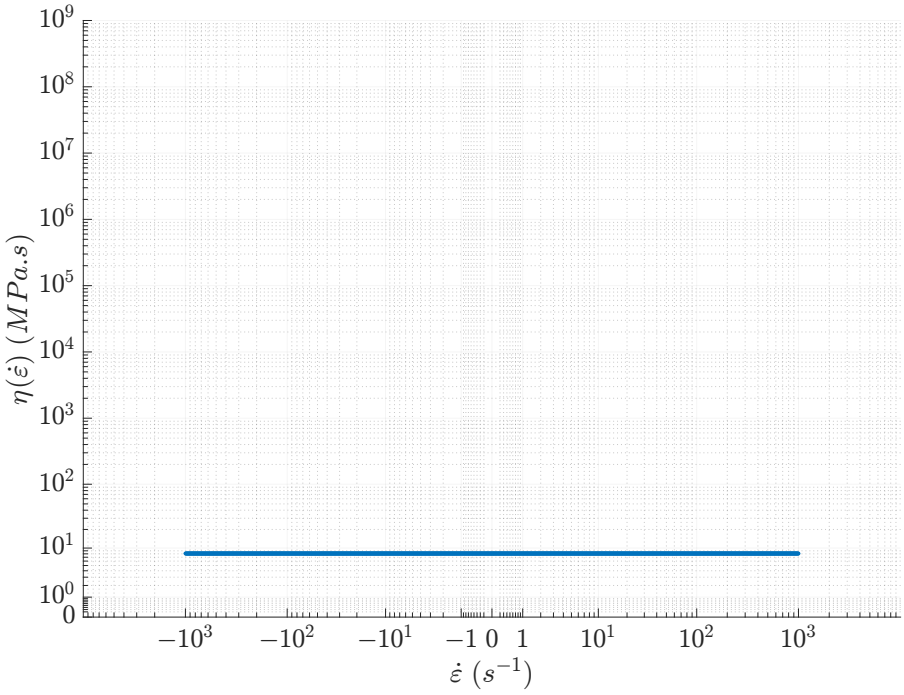


Figure C.46: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - $p = 1$ - Dashpot coefficient function

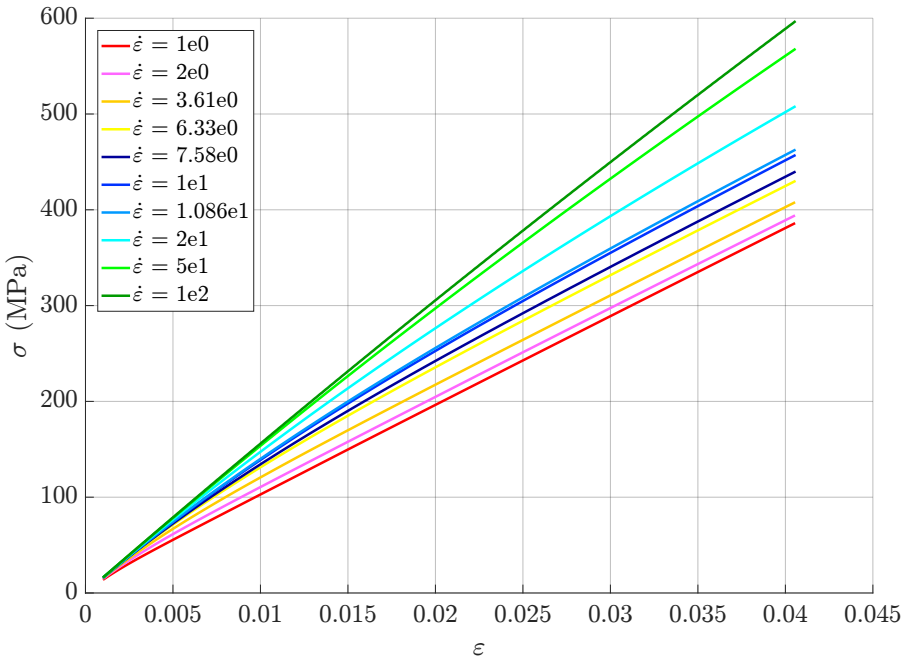


Figure C.47: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Abaqus - $p = 1$ - Stress - Strain graph

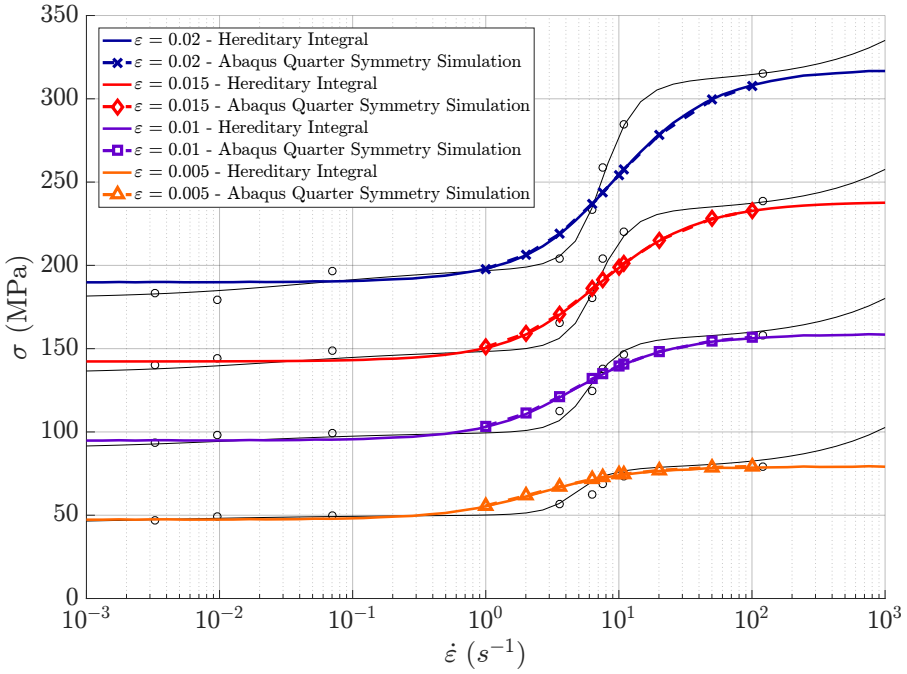


Figure C.48: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 1$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

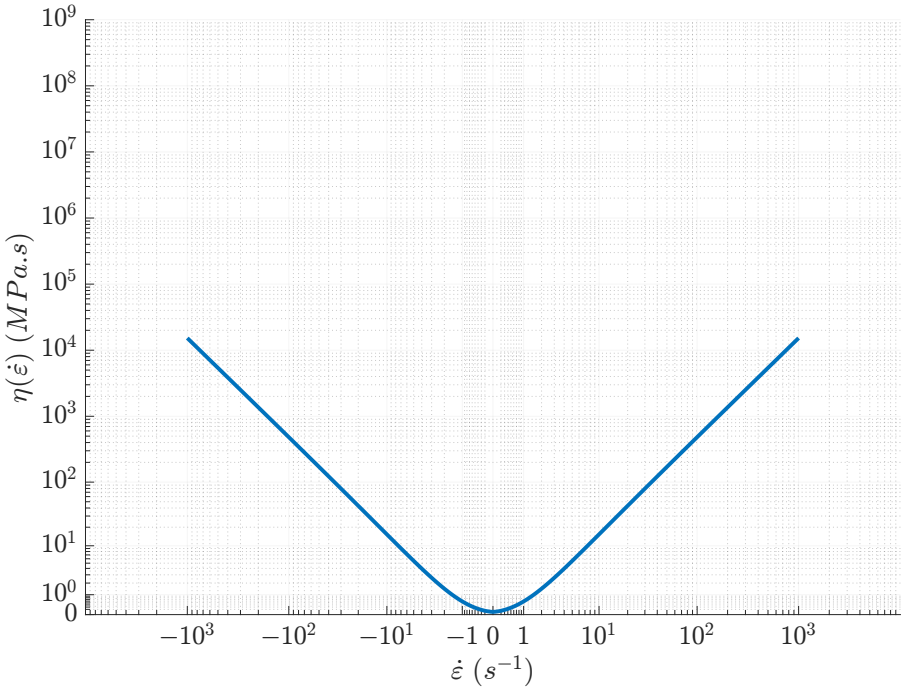


Figure C.49: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - $p = 2.5$ - Dashpot coefficient function

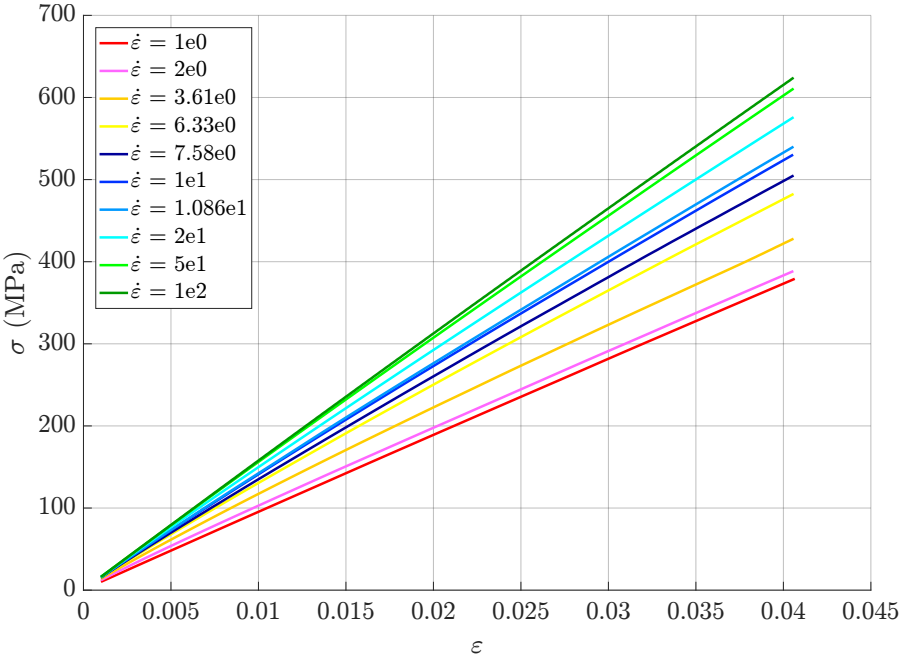


Figure C.50: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Abaqus - $p = 2.5$ - Stress - Strain graph

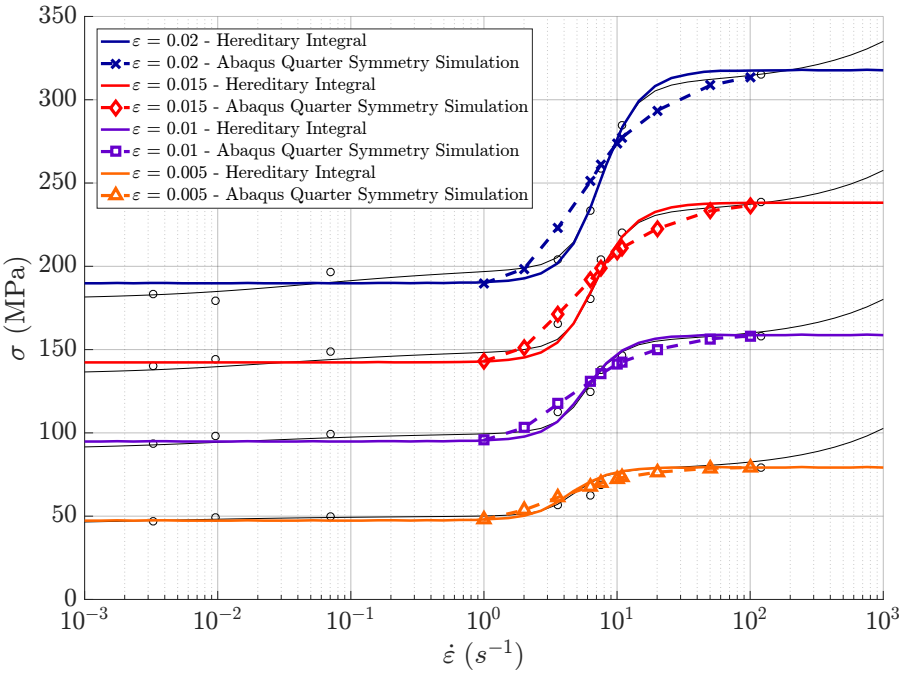


Figure C.51: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ - Stress - Strain Rate graph

Quarter Symmetry ($p = 4.5$, $\dot{\varepsilon}_{ref} = 0.35$)

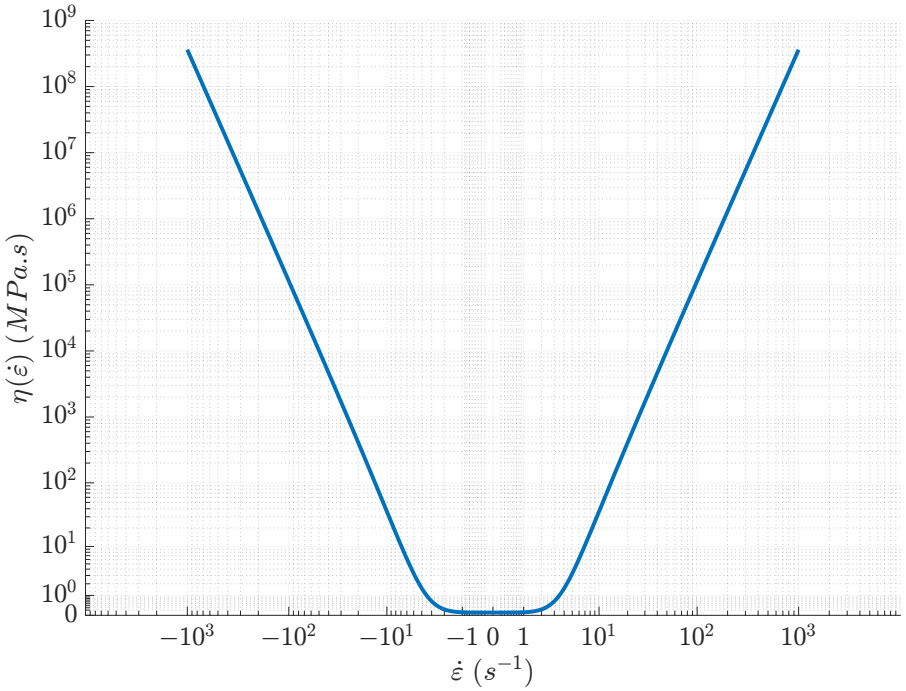


Figure C.52: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - $p = 4.5$ - Dashpot coefficient function

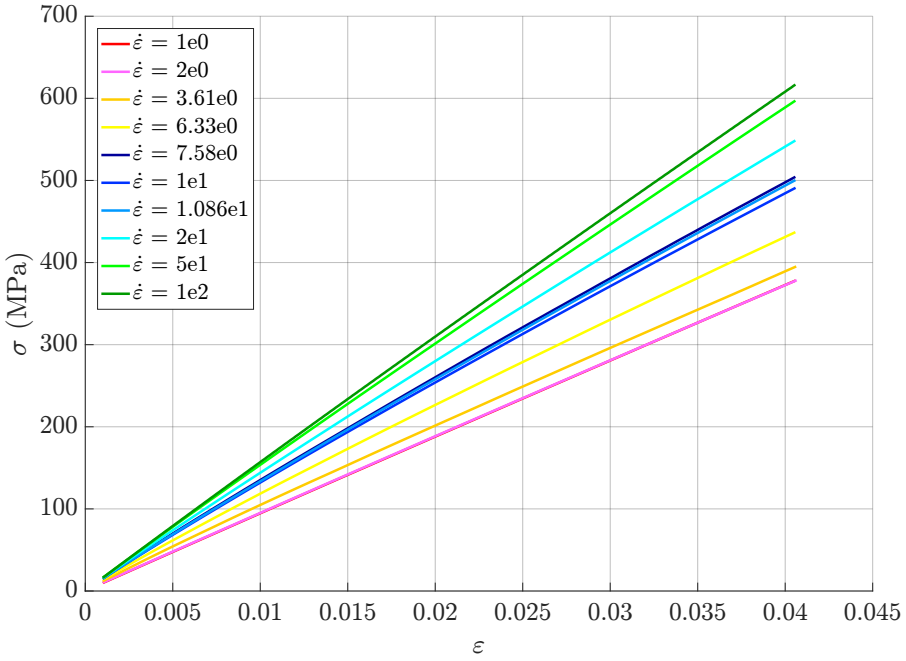


Figure C.53: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Abaqus - $p = 4.5$ - Stress - Strain graph

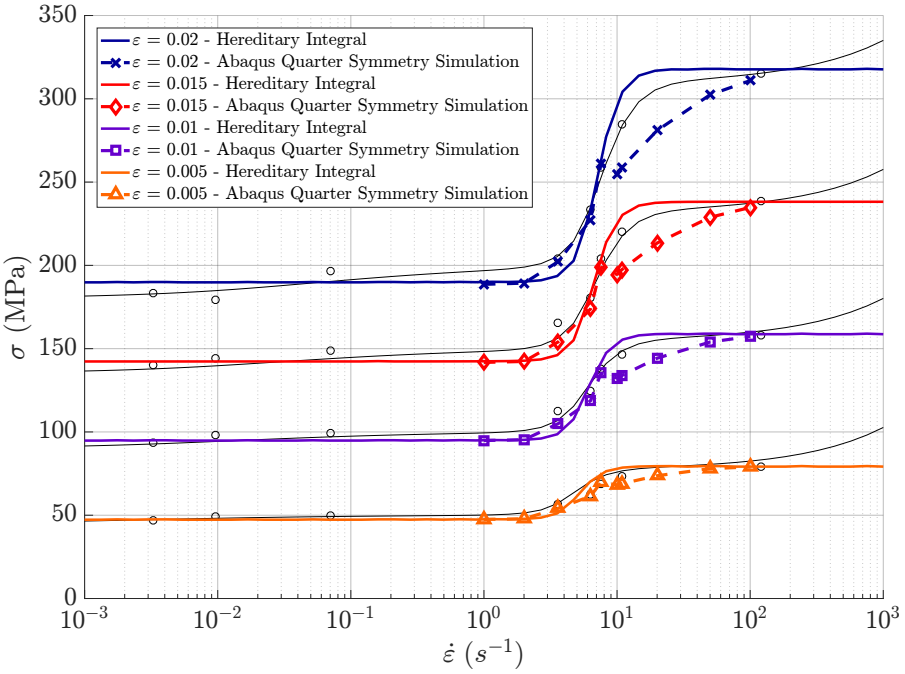


Figure C.54: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{11})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 4.5$ - Stress - Strain Rate graph

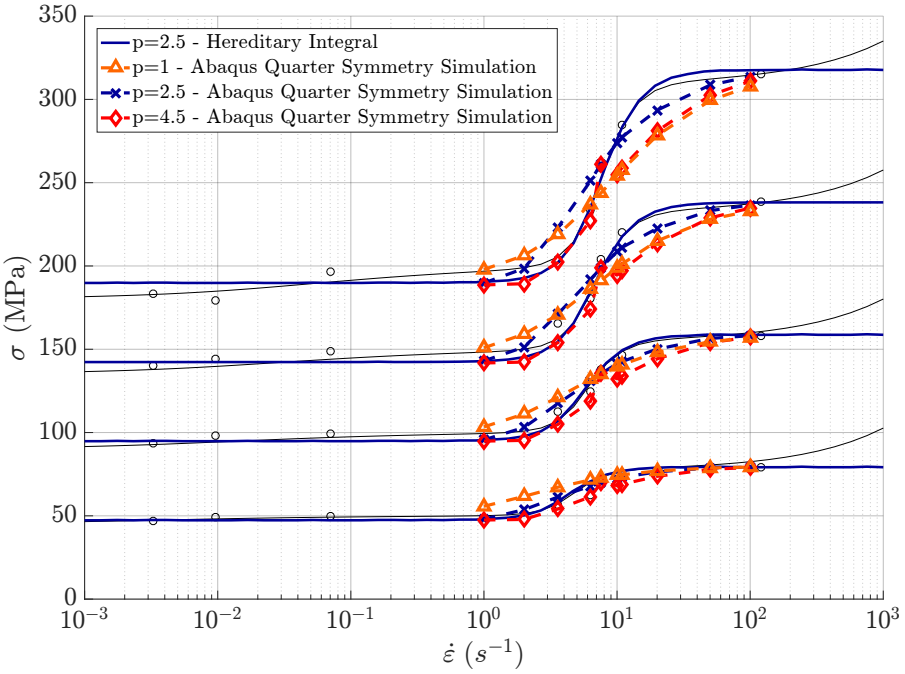


Figure C.55: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{ij})$ - Experimental vs Abaqus vs multi element hereditary integral - $p = 2.5$ vs 4.5 - Stress - Strain Rate graph

C.5.5 Cortical Bone Model - Equivalent Strain Rate

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

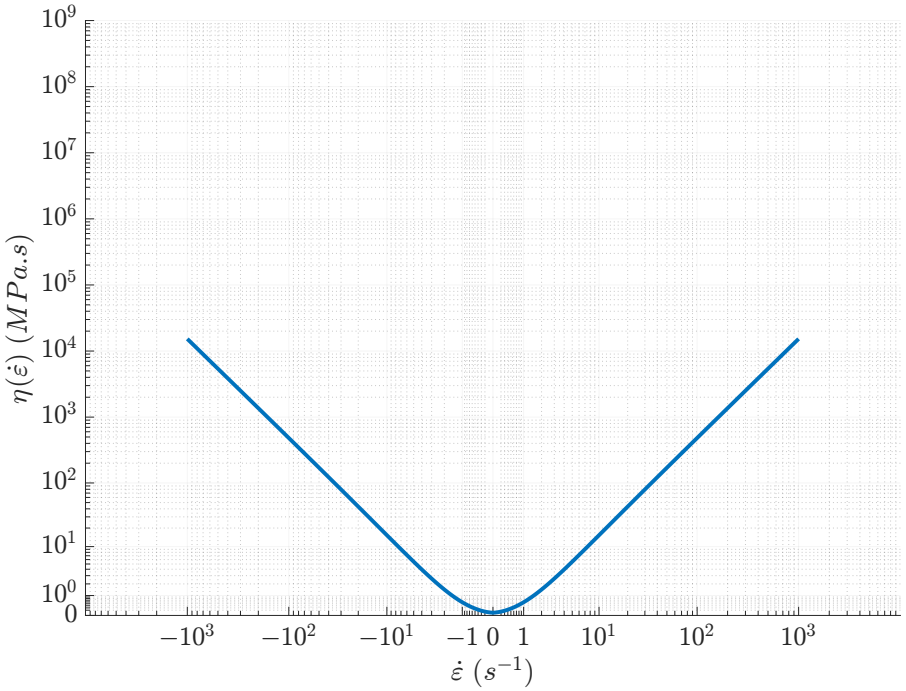


Figure C.56: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 0.35$

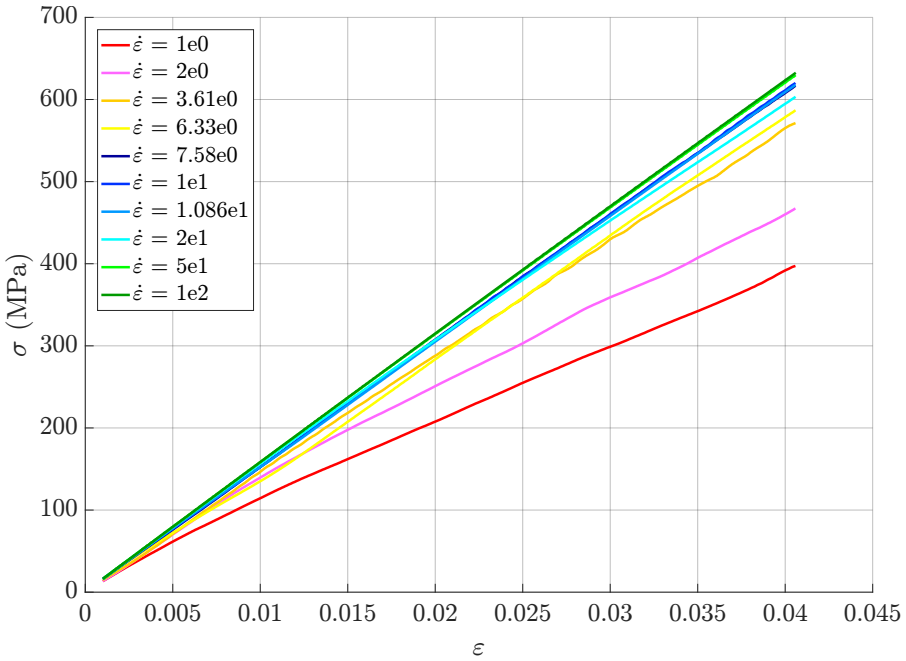


Figure C.57: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 0.35$

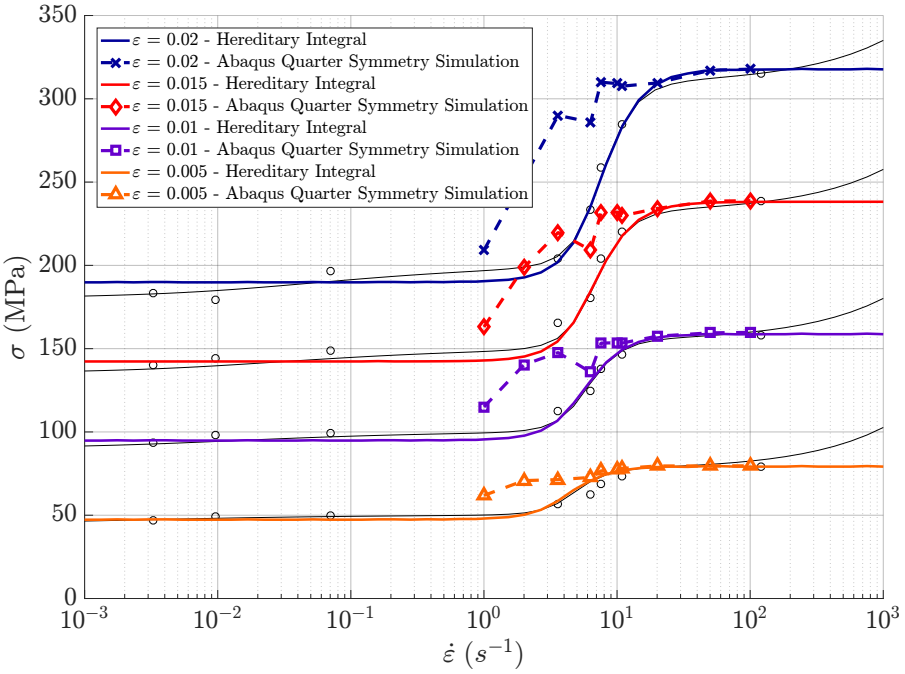


Figure C.58: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$

Single Element ($p = 2.5$, $\dot{\varepsilon}_{ref} = 1.2$)

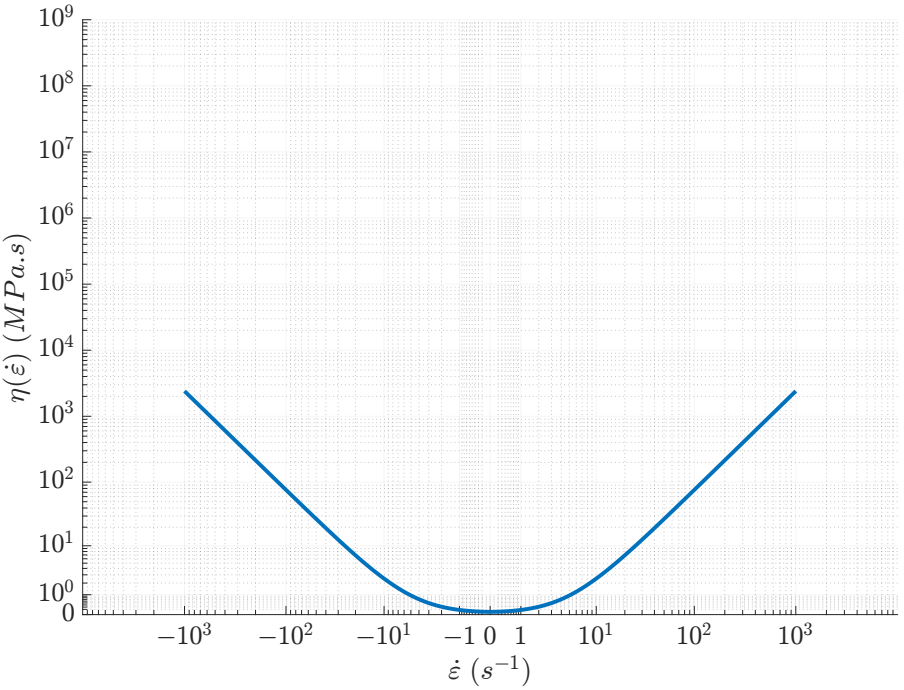


Figure C.59: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 1.2$

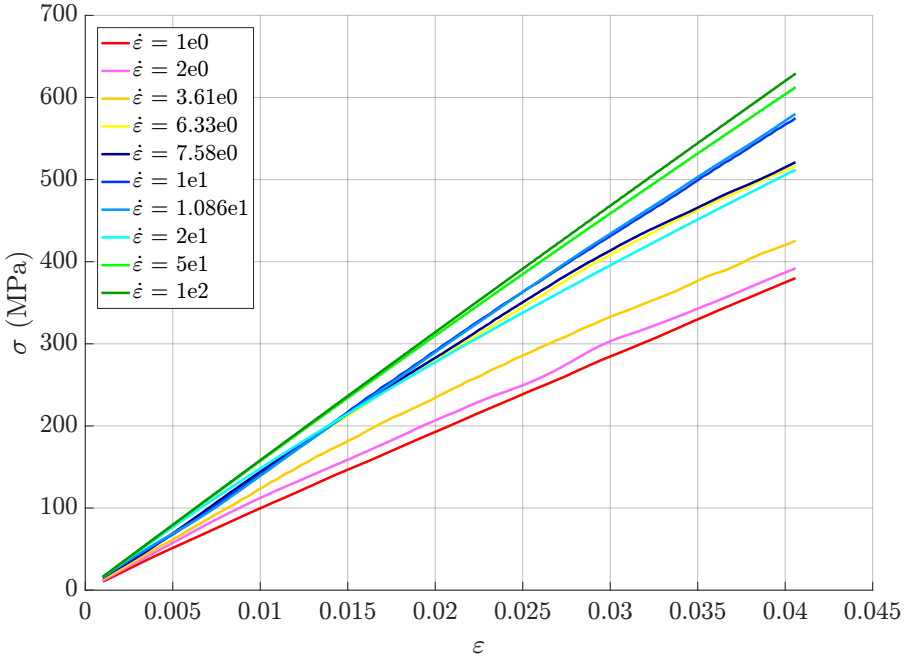


Figure C.60: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 1.2$

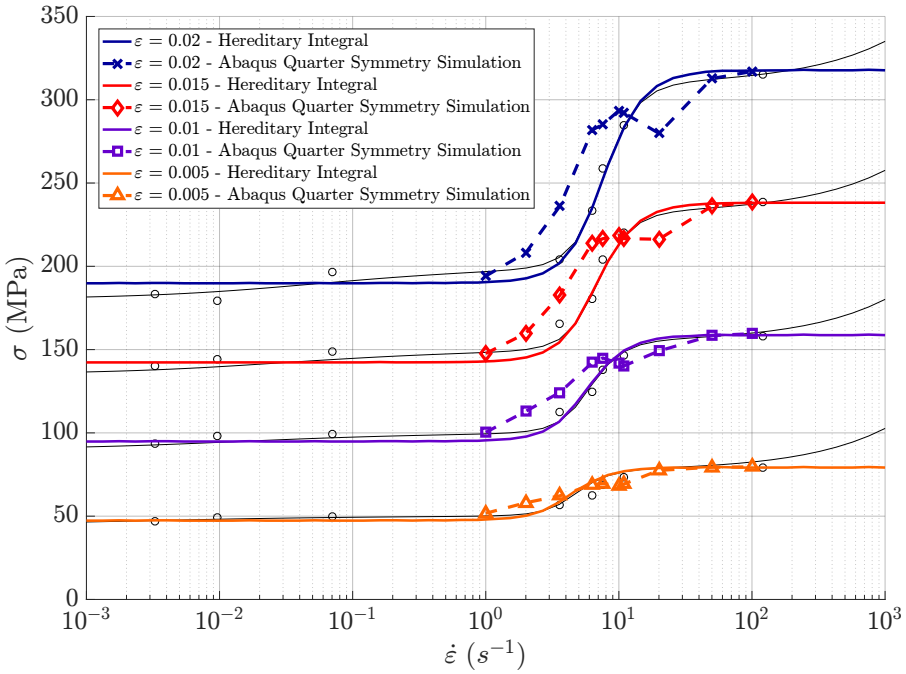


Figure C.61: Cortical Bone Model - Single Element - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 1.2$

Quarter Symmetry ($p = 1$, $\dot{\varepsilon}_{ref} = 0.35$)

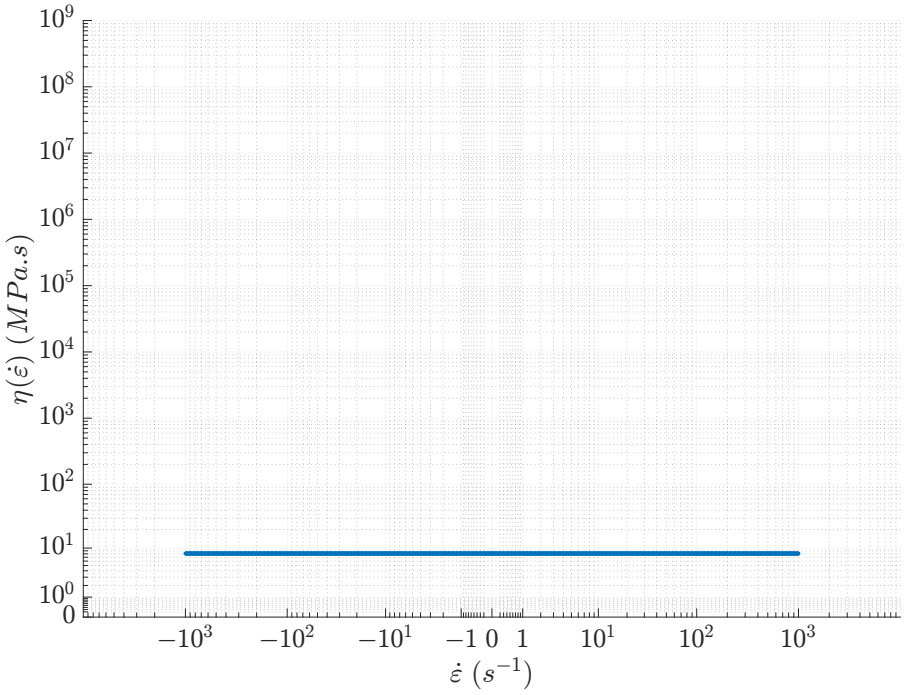


Figure C.62: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 0.35$

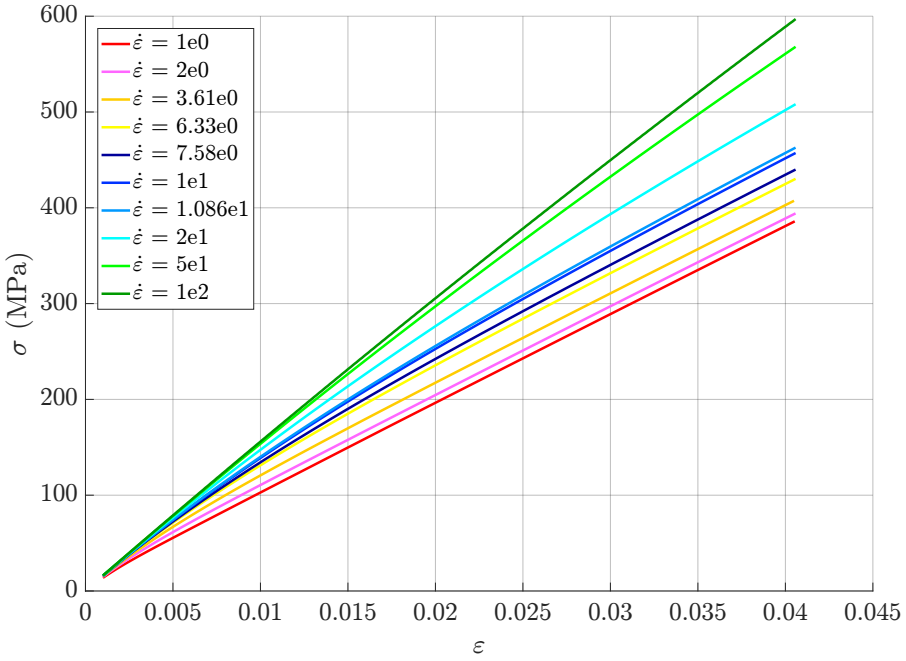


Figure C.63: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 0.35$

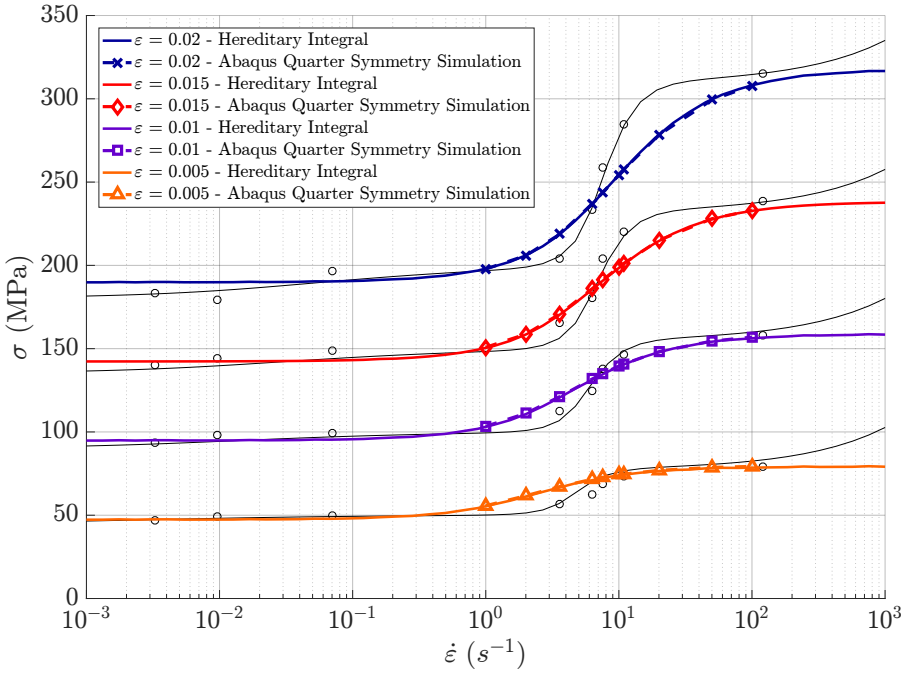


Figure C.64: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$

Quarter Symmetry ($p = 1.5$, $\dot{\varepsilon}_{ref} = 0.35$)

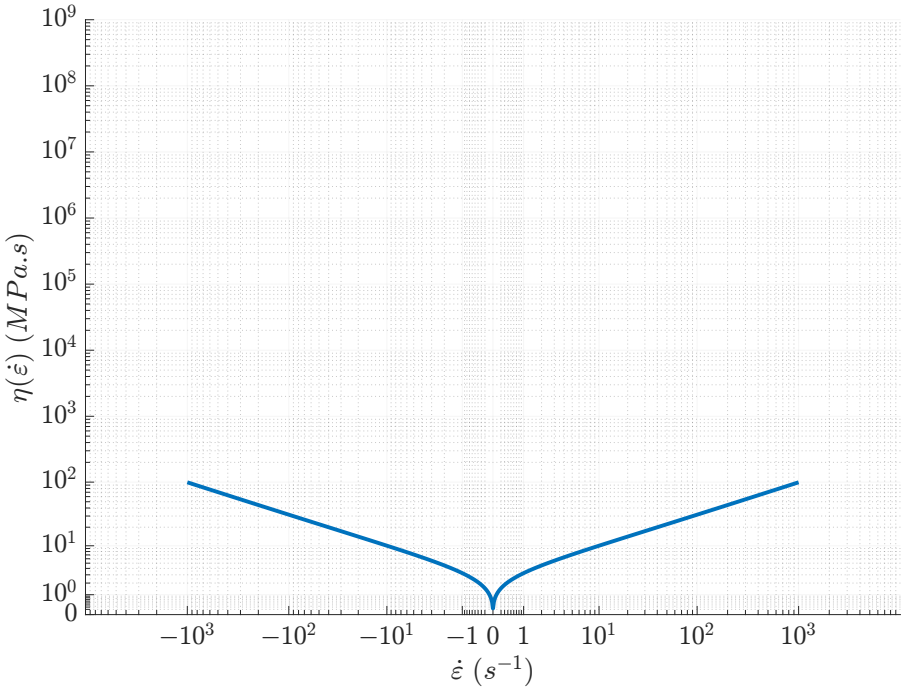


Figure C.65: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1.5$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 0.35$

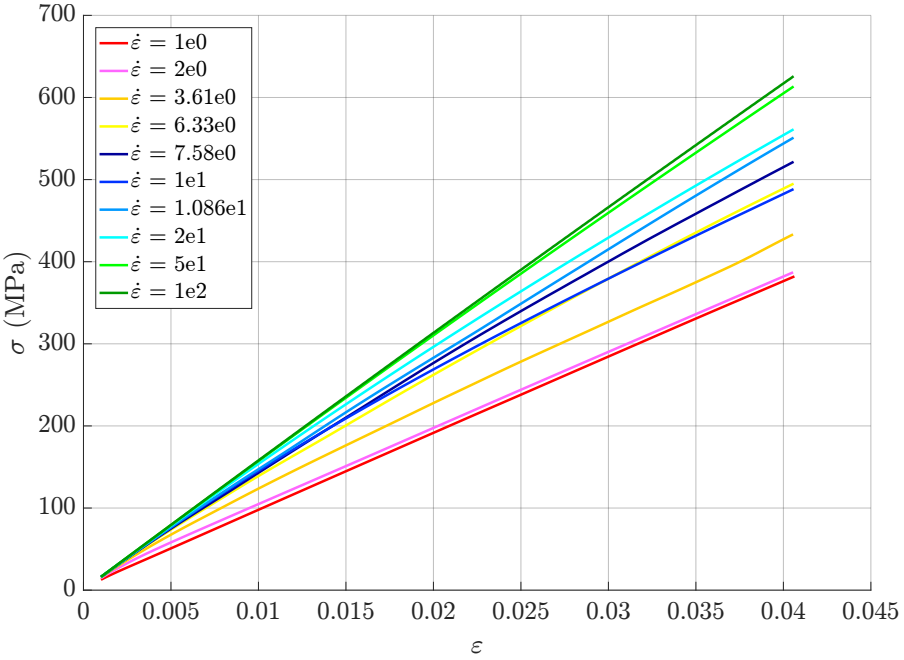


Figure C.66: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 0.35$

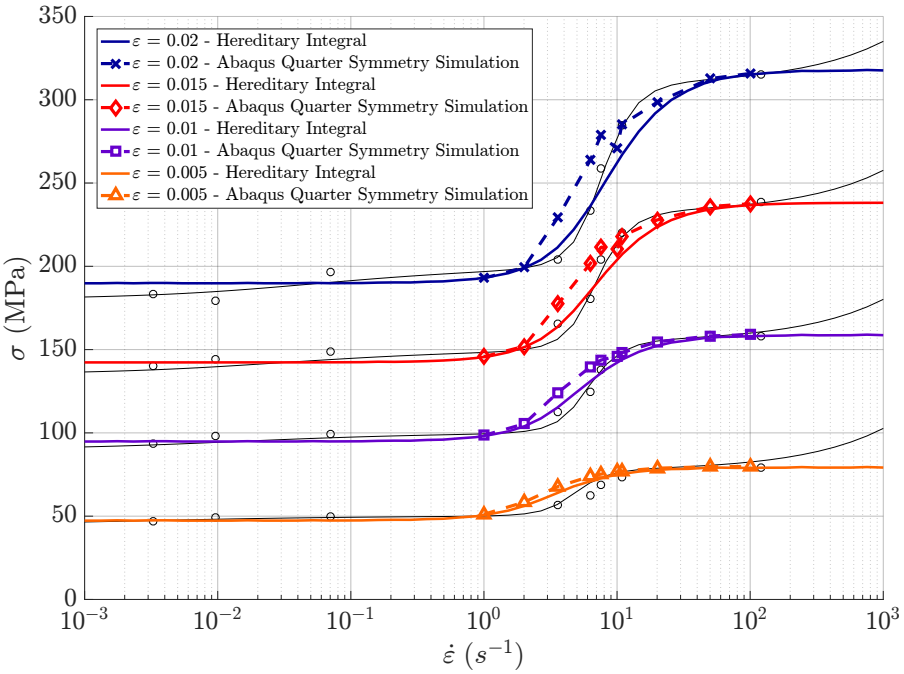


Figure C.67: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$

Quarter Symmetry ($p = 2.5$, $\dot{\varepsilon}_{ref} = 0.35$)

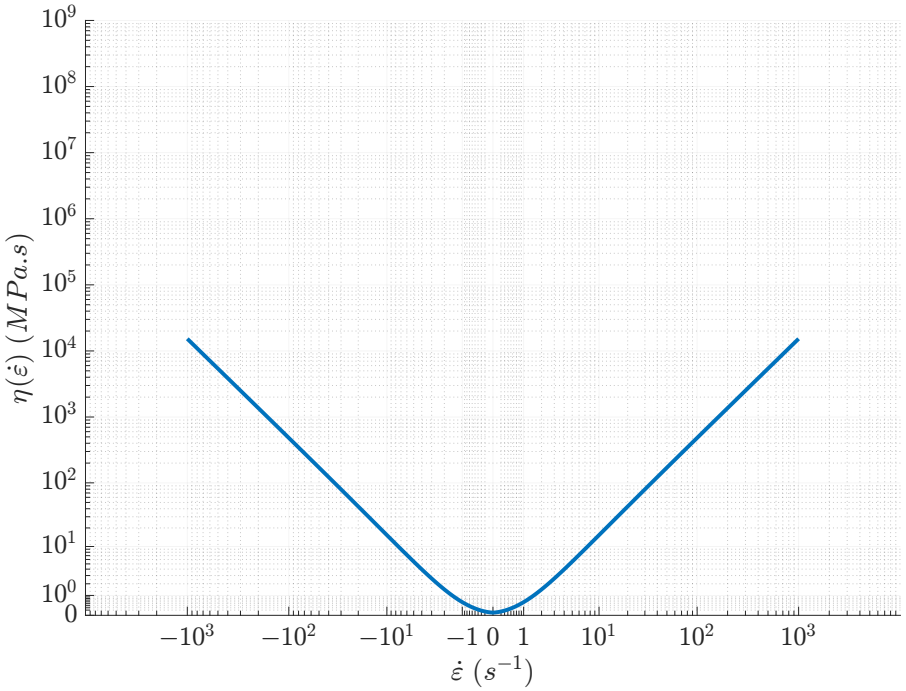


Figure C.68: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 0.35$

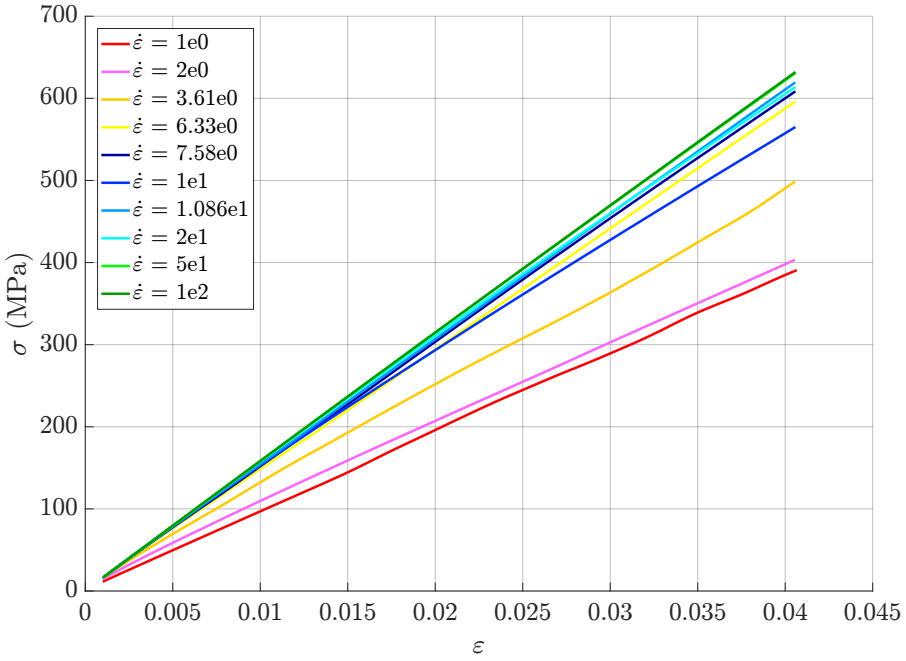


Figure C.69: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 0.35$

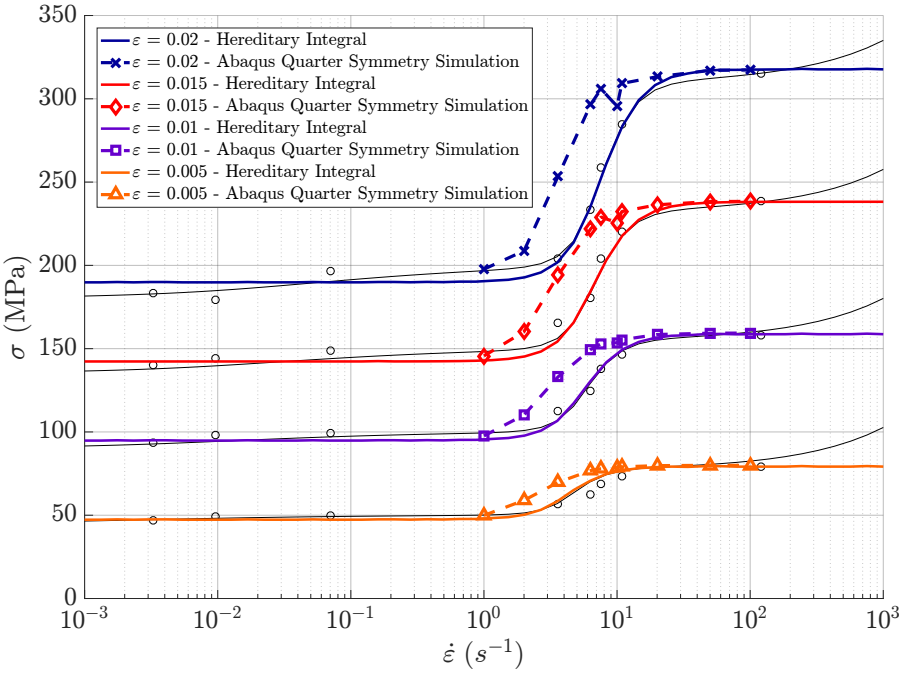


Figure C.70: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$

For a reference strain rate of 0.35, the stress - strain rate curves were compared for values of n equal to 1, 2.5 and 4.5 for the implementation of the equivalent strain rate. These were also compared to the one-dimensional hereditary integral for a value of n equal to 2.5 and the experimental results by Cloete *et al.* [3].

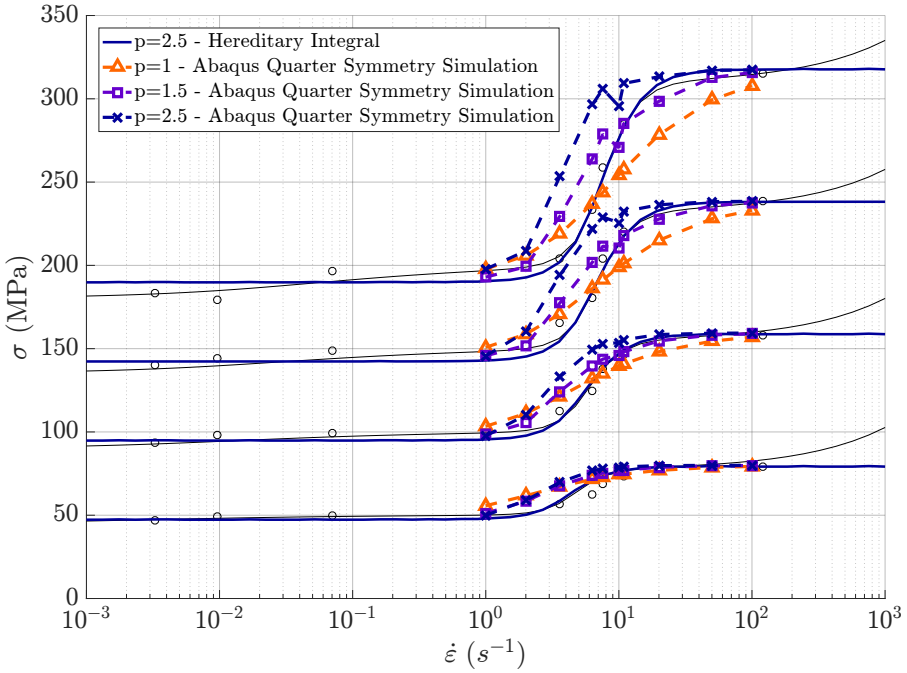


Figure C.71: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 1$ vs 1.5 vs 2.5 - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 0.35$

Quarter Symmetry ($p = 2.5$, $\dot{\varepsilon}_{ref} = 1.2$)

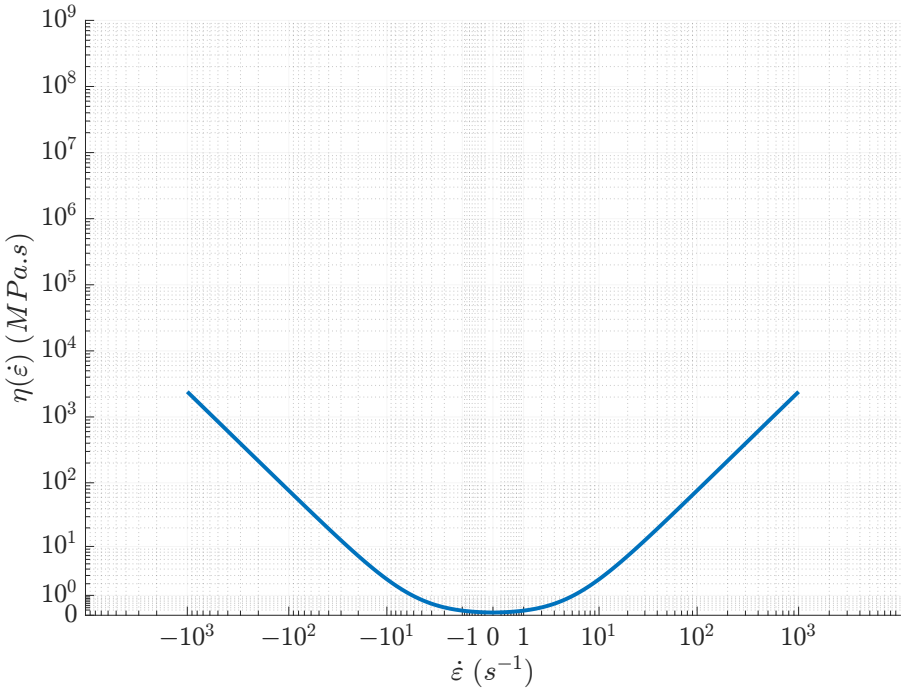


Figure C.72: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Dashpot coefficient function - $\dot{\varepsilon}_{ref} = 1.2$

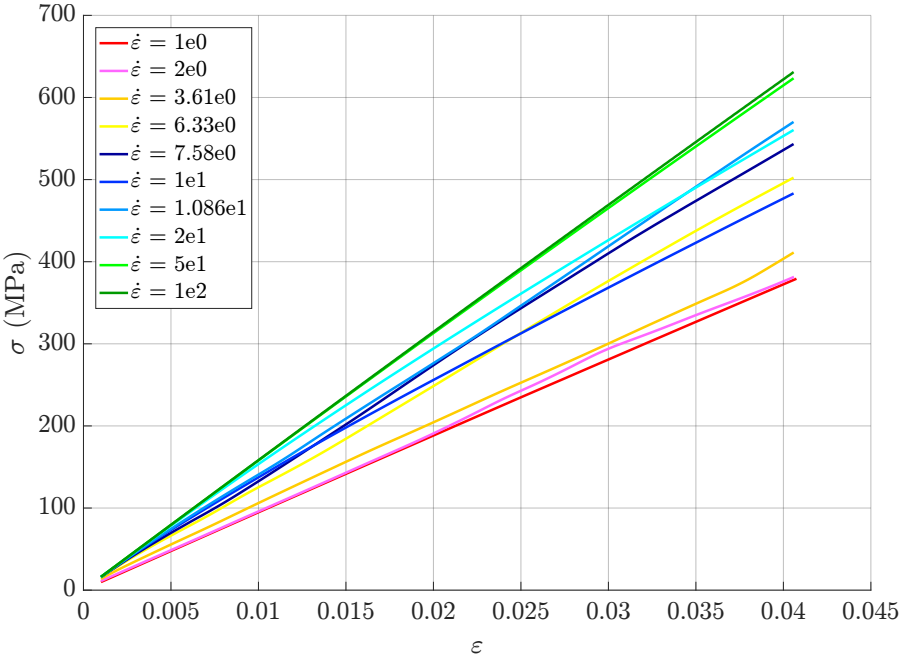


Figure C.73: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 1.2$

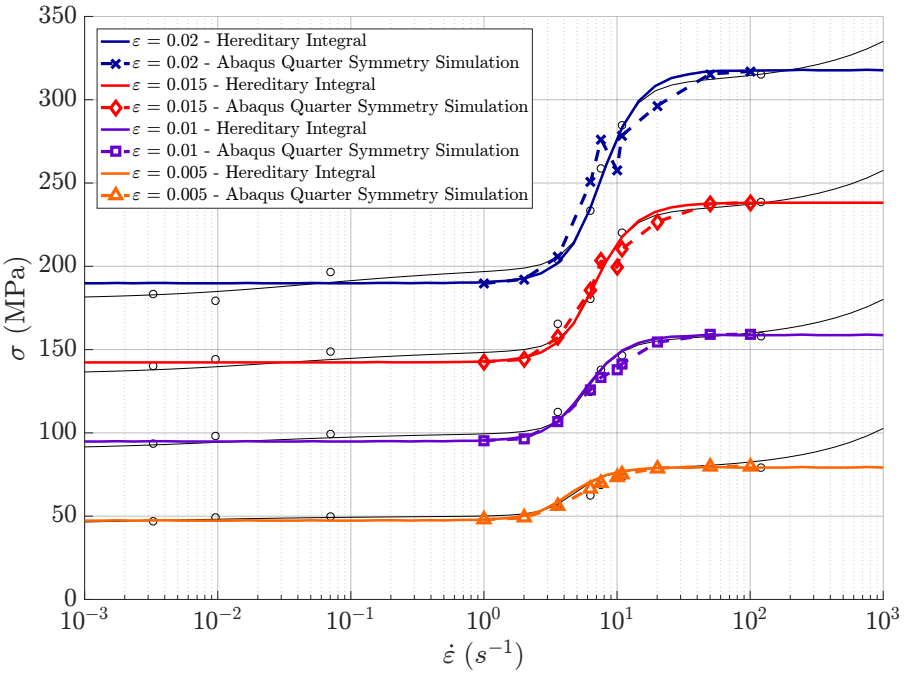


Figure C.74: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 1.2$

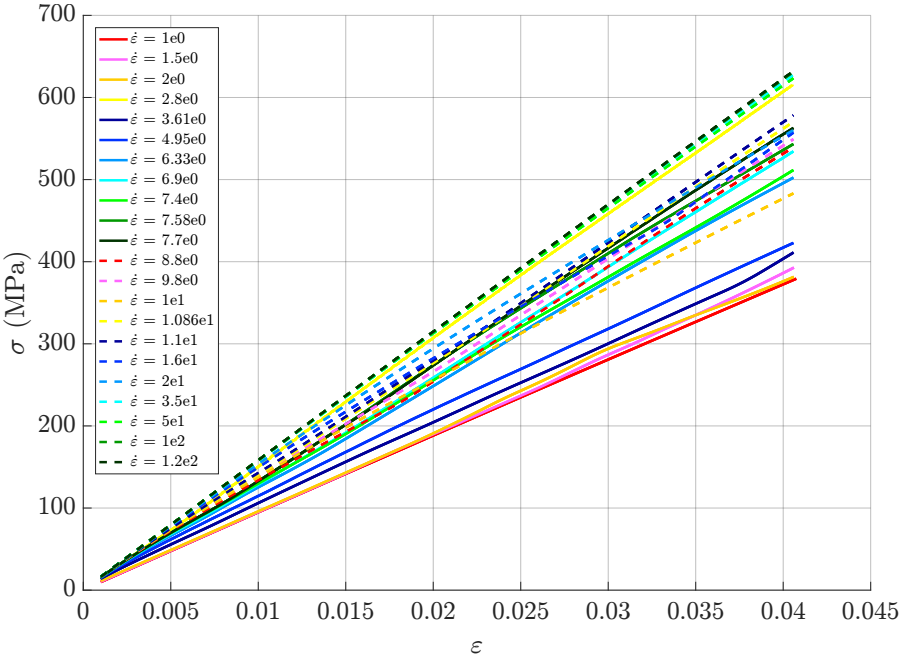


Figure C.75: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Abaqus - Stress - Strain graph - $\dot{\varepsilon}_{ref} = 1.2$

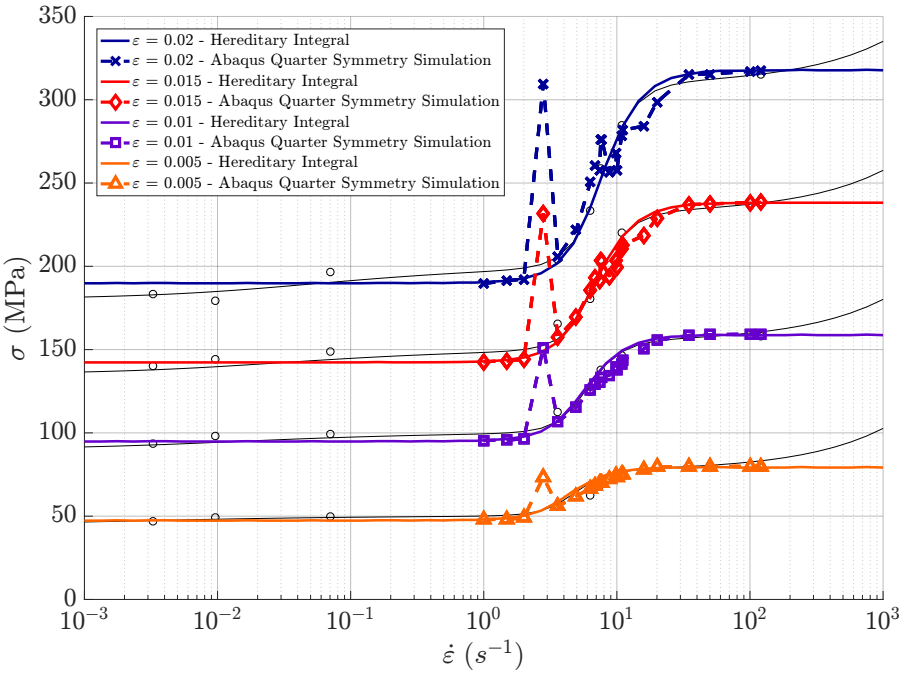


Figure C.76: Cortical Bone Model - Quarter Symmetry - $\eta(\dot{\varepsilon}_{eq})$ - $p = 2.5$ - Experimental vs Abaqus vs multi element hereditary integral - Stress - Strain Rate graph - $\dot{\varepsilon}_{ref} = 1.2$

Appendix D

Ethics Form