

© 2010 by Natarajan Chennimalai Kumar. All rights reserved.

NUMERICAL MODELING OF CORTICAL BONE ADAPTATION DUE TO
MECHANICAL LOADING USING THE FINITE ELEMENT METHOD

BY

NATARAJAN CHENNIMALAI KUMAR

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2010

Urbana, Illinois

Doctoral Committee:

Professor Jonathan A. Dantzig, Chair & Director of Dissertation Research
Professor Iwona M. Jasiuk, Contingent Chair
Assistant Professor Amy Wagoner Johnson
Professor Charles H. Turner, Indiana University–Purdue University Indianapolis

Abstract

It is well known that bone tissue adapts its shape and structure according to its mechanical environment. Bone adaptation occurs on the dense cortical bone and porous trabecular bone. The process of bone adaptation is shown to be dependent on a number of mechanical loading parameters such as magnitude, frequency, number of bouts etc. of applied loading through experimental studies. We propose to develop a numerical framework, which can simulate and predict cortical bone adaptation due to different parameters of loading. In pursuit of the development of the framework, we develop a method to generate finite element (FE) models of actual rat ulna from micro computed tomography (μ -CT) images. The external adaptation process is implemented in the model by moving the surface nodes of the FE mesh along the normal direction based on an evolution law characterized by two parameters: one that captures the rate of the adaptation process (referred to as gain); and the other characterizing the threshold value of the mechanical stimulus required for adaptation (referred to as threshold-sensitivity).

Cortical bone is firstly modeled as an elastic material. Loading from experiments of Robling *et al* [1] is applied on the FE model and the elastic boundary value problem is solved. Based on the results of the FE solution, the surface nodes are displaced according to the local strain energy density as the growth stimulus. Using this stimulus, we show that the model can simulate the effect of the magnitude of applied loading on the growth response. We calibrate the growth law parameters by comparing the results from our model to the experimental results. A parametric study is carried out to evaluate the effect of these two parameters on the adaptation response. We show, following comparison of results from the

simulations to the experimental observations, that splitting the loading cycles into different number of bouts affects the threshold-sensitivity but not the rate of adaptation. We also show that the threshold-sensitivity parameter can quantify the mechanosensitivity of the osteocytes. The use of strain energy density stimulus and elastic material model cannot simulate the effect of frequency of applied loading on the cortical bone adaptation response.

We model cortical bone as a poroelastic material to account for the interstitial fluid flow. We aim to develop a growth stimulus similar to strain energy density for the poroelastic material model. In order to achieve this goal, we develop the FE model of a rectangular beam subjected to pure bending. This geometric model is chosen for simplicity, as an idealized representation of cortical bone. We then propose the use of the dissipation energy of the poroelastic flow as a mechanical stimulus for bone adaptation, and show that it can predict the effect of frequency of the applied load. Surface adaptation in the model depends on the weighted average of the mechanical stimulus in a “zone of influence” near each surface point, in order to incorporate the non-locality in the mechanotransduction of osteocytes present in the lacunae. We show that the dissipation energy stimulus and the resulting increase in second moment of inertia of the cross section increase linearly with frequency in the low frequency range (less than 10 Hz) and saturate at the higher frequency range (greater than 10 Hz). Similar non-linear adaptation frequency response also has been observed in numerous experiments. We extend the poroelastic material model, dissipation energy stimulus, and the zone of influence to the actual rat ulna FE model. We implement orthotropic permeability on the rat ulna model in order to be anatomically consistent. We calibrate the growth law parameters (gain and threshold-sensitivity) using experimental results. We analyze the growth response of cortical bone for a range of frequencies (from 2 Hz to 25 Hz) and show that the adaptation response is non-linear with respect to the frequency of loading.

To my parents.

Acknowledgments

Firstly, I am greatly indebted to my doctoral advisors Profs. Jonathan A. Dantzig and Iwona M. Jasiuk. They have encouraged me to pursue what I wanted to do, and guided me along the way. I thank them for their patience with me when things did not go well and their enthusiastic appreciation for things well done. The past four years have been the best years of my life in terms of professional, intellectual, and personal development, all thanks to Jon and Iwona. I am especially thankful to Jon for his patience in explaining difficult concepts to me, diligent editing of my papers and presentations, and pointers on what it takes to be a good doctoral candidate. Special thanks to Iwona for introducing me to the exciting world of bone biomechanics, and her constant support for my work. I am extremely grateful to Jon, Iwona and the Department of Mechanical Science and Engineering for funding my doctoral study.

I thank Prof. Charles Turner and Prof. Amy Wagoner Johnson for being on my doctoral committee. My special thanks to Charles for his active involvement in my doctoral thesis including sharing of experimental data, many thought-provoking discussions, and his able guidance throughout. I would also like to thank Prof. Daniel Tortorelli for introducing me to the smoothing filter approach. I extend my thanks to Zhi, Maggie, and Joe for making the lab environment enjoyable, to Liang, Elham, David, and YikHan for the wonderful discussions on bone mechanics, great food and good times. I also would like to thank Profs. Gustavo Gioia and Richard Keane for being great guides of teaching when I was a Teaching Assistant. My thanks are due to the secretaries of MechSE department especially Pam Vanetta, Tammy Smith, Kathy Smith and Susan Bailey for their help throughout my doctoral study.

A strong social circle is essential to survive the highly demanding academic environment at UIUC. My heartfelt thanks to my friends Lavanya, Raja, Rajesh, Anil, Kowtilya, Dwarak, Shivakumar, Ankur, Rekha, Preeyaa, Meghana, Piyush, and a number of others for making me feel at home at UIUC. I thank my teachers, colleagues and supervisors throughout my academic life, who have been instrumental in shaping my career, especially Prof. P. Baskaran, for instilling in me a deep liking for the subject of Solid Mechanics, and Prof. Petros Sofronis, for expanding the horizons of my thinking, and Prof. Philippe Geubelle, for his guidance as my Masters advisor.

Finally, I cannot thank my family enough for their unconditional love and support. I would not have reached this point in my life without their presence. My loving thanks to Amma, Appa, sister Vani, and brother Kannan. As a small token of my gratitude towards my parents, I dedicate this work to them.

Table of Contents

| | |
|---|-----------|
| List of Tables | ix |
| List of Figures | x |
| List of Symbols | xiii |
| Chapter 1 Background | 1 |
| 1.1 Introduction & Motivation | 1 |
| 1.2 Literature review: Experimental studies | 3 |
| 1.3 Literature review: Numerical models | 6 |
| 1.4 Organization of the thesis | 8 |
| Chapter 2 Development of the numerical framework | 10 |
| 2.1 Generation of the finite element mesh | 10 |
| 2.2 Implementation of surface evolution | 11 |
| 2.3 Elastic properties, boundary conditions and assumptions | 15 |
| 2.4 Quantification of bone adaptation | 17 |
| Chapter 3 Strain energy stimulus – Growth response | 19 |
| 3.1 Introduction | 19 |
| 3.2 Growth Equation | 20 |
| 3.3 Bone growth simulations | 22 |
| 3.4 Parametric study of timesteps, gain and threshold-sensitivity | 24 |
| 3.5 Validation with experiments | 28 |
| 3.6 Biological significance of growth law parameters | 31 |
| 3.6.1 Significance of threshold-sensitivity ϕ_{ref} | 32 |
| 3.6.2 Significance of gain A | 33 |
| 3.7 Validation of assumptions - homogeneity and isotropy | 34 |
| 3.8 Summary | 35 |
| Chapter 4 Poroelasticity – Rectangular beam model | 36 |
| 4.1 Introduction | 36 |
| 4.1.1 Porosities of cortical bone | 37 |
| 4.1.2 Role of fluid flow in mechanotransduction | 38 |
| 4.2 Theory of poroelasticity | 40 |

| | | |
|-------------------|--|------------|
| 4.3 | Development of the stimulus for the growth law | 44 |
| 4.3.1 | Simulation details | 46 |
| 4.4 | Results | 47 |
| 4.5 | Discussion | 54 |
| 4.6 | Summary | 55 |
| Chapter 5 | Poroelasticity – Rat ulna model | 56 |
| 5.1 | Introduction | 56 |
| 5.2 | Methods | 56 |
| 5.2.1 | Orthotropic permeability | 57 |
| 5.2.2 | Zone of influence | 58 |
| 5.2.3 | Simulation details | 58 |
| 5.3 | Results | 59 |
| 5.4 | Discussion | 62 |
| Chapter 6 | Conclusions and Future Work | 66 |
| Appendix A | Bone adaptation code | 68 |
| Appendix B | ABAQUS Input Deck | 120 |
| Appendix C | ABAQUS <i>ORIENT</i> user-subroutine | 125 |
| References | | 130 |
| Vita | | 135 |

List of Tables

| | | |
|-----|--|----|
| 4.1 | Values of different poroelastic parameters used in the analysis | 46 |
| 5.1 | Values of different poroelastic parameters used for the rat ulna in the analysis | 59 |

List of Figures

| | | |
|-----|--|----|
| 1.1 | Schematic of the skeletal system of the human body, with zoomed in view of the femur showing the trabecular and cortical bone. Images from the bone exhibit of the Smithsonian Institute http://anthropology.si.edu/writteninbone/inside_look.html | 1 |
| 1.2 | (a) Schematic of a typical rat ulna compression experiment, and (b) Typical results from a rat ulna compression experiment: Cross-section on top shows the growth observed on the loaded ulna after 16 weeks, and the plots in the bottom show the observed increase in bone mineral content, strength and fracture toughness of the ulna due to adaptation (from [2]). | 4 |
| 1.3 | Results from different experiments showing the dependence of bone growth on the various mechanical loading parameters: (a) Magnitude of loading [3], (b) Frequency of loading [4], (c) Number of cycles of loading [5], (d) Time between the bouts of applied loading [6], (e) Variation in maximum moment of inertia I_{max} of the cross-section for multiple bouts of loading, and (f) Variation in I_{min} for multiple bouts of loading [1] | 5 |
| 2.1 | FE model of rat ulna generated using HyperMesh from the μ -CT images | 10 |
| 2.2 | Schematic of the flowchart that describes the implementation of the adaptation procedure | 11 |
| 2.3 | Computation of the normal at a surface node using the cross product. | 13 |
| 2.4 | Demonstration of the smoothing procedure on a simple circular cylinder model. (a) Jagged surface obtained when using a weighting factor $w = 1$, (b) Smooth surface for a weighting factor $w = 0$, and (c) Smooth surface for a weighting factor of $w = 0.5$, which includes the local information and is also smooth. | 14 |
| 2.5 | Comparison of the rat ulna strain values from experiments and from the FE simulation | 16 |
| 3.1 | Results from the experiments of Robling <i>et al</i> showing (a) the percentage increase in the maximum moment of inertia (I_{max}), and (b) the percentage increase in the minimum moment of inertia (I_{min}). | 20 |
| 3.2 | Contours of strain energy density in the rat ulna (a) before growth, and (b) after 20 growth timesteps. | 23 |
| 3.3 | Comparison of the cross sections between the original and grown rat ulna at the midshaft from (b) our simulations and (a) the experiments of Robling <i>et al</i> [1]. | 24 |

| | | |
|------|--|----|
| 3.4 | (a) Variation of increase in I_{min} for the baseline parameters of the gain and sensitivity parameter for different timesteps, and (b) change in the maximum increase in I_{min} with increasing values of timesteps. | 25 |
| 3.5 | (a) Variation of increase in I_{min} for different values of the gain parameter for the baseline value for the sensitivity parameter, and (b) change in the maximum increase in I_{min} with increasing values of the gain parameter A | 26 |
| 3.6 | (a) Variation of increase in I_{min} for different values of sensitivity parameter ϕ_{ref} for the baseline value of the gain parameter A , and (b) change in the maximum increase in I_{min} with increasing values of sensitivity parameter ϕ_{ref} | 27 |
| 3.7 | Plot showing the increase in the different geometric properties of the grown ulna due to adaptation. It can be observed that the increase in I_{min} is the most significant. | 28 |
| 3.8 | Plot showing the increase in the I_{min} due to adaptation from the numerical model compared to the experimental results. | 29 |
| 3.9 | Plot showing the increase in the I_{max} and area due to adaptation from the numerical model compared to the experimental results. | 30 |
| 4.1 | Schematic of the macro and microstructure of cortical bone. (a) Wall of the diaphysis of a long bone showing the vascular porosity (haversian system including the osteons, Haversian and Volkmann canals), (b) Part of the Haversian system showing the lacuno-canalicular porosity consisting of lacunae and canaliculi. Note the radially oriented canalicular channels connecting the lacunae. Figures taken from Junquera and Carneiro [7]. | 38 |
| 4.2 | Schematic of the four-point bending setup of the rectangular poroelastic beam | 43 |
| 4.3 | Schematic of the spherical zone of influence on a typical bone cross section | 45 |
| 4.4 | Finite element mesh of the rectangular beam. Note the biased mesh across the cross-section | 47 |
| 4.5 | Plots of pore pressure (a, c, and e) and fluid flow velocity (b, d, f) solutions as a function of time for several frequencies. | 48 |
| 4.6 | Plots of pore pressure (a, c, e) and fluid flow velocity (b, d, f) solutions across the cross-section at different points of time in a loading cycle shown here as the phase angle for 1 Hz, 5 Hz and 30 Hz. | 49 |
| 4.7 | Plot of pore pressure solution across the cross-section for different frequencies at the end of a loading cycle. | 51 |
| 4.8 | Plot of the fluid velocity pressure solution across the cross-section for different frequencies at the end of a loading cycle. | 51 |
| 4.9 | Plot of dissipation energy (time integral of dissipation potential) as a function of the y-coordinate for different frequencies. Also superimposed on the plot is the influence functions. We chose $\exp(-5 x /d)$ as the influence function for our simulations. We have zoomed in on the plot to show the variation in the dissipation energy inside the beam, and so the very high values near $y = -0.5$ mm are cut off. | 52 |
| 4.10 | Plot of dissipation energy stimulus at a surface node as a function of loading frequency. | 52 |

| | | |
|------|---|----|
| 4.11 | Plot showing the percentage increase in the moment of inertia of the rectangular cross section due to adaptation as a function of frequency. | 53 |
| 5.1 | Schematic of part of the Haversian system showing the lacuno-canalicular porosity consisting of lacunae and canaliculi. Note the radially oriented canaliculi channels connecting the lacunae [7]. | 57 |
| 5.2 | Fluid velocity fields for frequency of 2 Hz corresponding to (a) loading cycle, and (b) unloading cycle. | 60 |
| 5.3 | Comparison of the cross-sections between the original and grown rat ulna at the midshaft from our simulations using the dissipation energy stimulus. Compare this to the results from the experiments of Robling <i>et al</i> [1] in Chapter 3. | 60 |
| 5.4 | Plot showing the increase in the I_{min} due to adaptation from the numerical model compared to the experimental results of Robling <i>et al</i> [1]. | 61 |
| 5.5 | Plot showing the percentage increase in the moment of inertia of the rat ulna midshaft cross-section due to adaptation as a function of frequency. | 62 |
| 5.6 | Plot showing the increase in the I_{min} due to adaptation from the numerical model for the coarse and fine mesh compared to the experimental results of Robling <i>et al</i> [1]. | 63 |
| 5.7 | Plot showing the increase in the I_{min} due to adaptation from the numerical model with the FE mesh of the intermedullary canal to show the correlation between the size of the intermedullary canal and the valleys in the I_{min} plot. | 64 |

List of Symbols

| Symbol | Quantity represented |
|-----------|---|
| A | area |
| B | Skempton's Coefficient |
| E | modulus of elasticity (Young's modulus) |
| Fo | Fourier number |
| G | Shear modulus |
| I | Moment of inertia |
| I_{max} | Maximum principal moment of inertia |
| I_{min} | Minimum principal moment of inertia |
| K | Drained bulk modulus |
| K_f | Fluid bulk modulus |
| K_s | Solid grain bulk modulus |
| K_u | Undrained bulk modulus |
| L | Length |
| M | Bending moment |
| P | Applied load |
| T | Growth timescale |
| V | Bulk volume |
| V_z | Volume of zone of influence |
| X_i | Cartesian coordinates |

| | |
|-------------------|---|
| b_i | Normal displacement of surface node i |
| c | Hydraulic diffusivity |
| d | Thickness of the rectangular beam |
| e | Void ratio |
| f | Frequency of loading |
| $f(\ x\)$ | Influence function for the nodes inside the zone of influence |
| k | Intrinsic permeability |
| p | Pore pressure |
| q_i | Fluid flux component |
| r, θ, z | Cylindrical coordinates |
| t | Loading timescale |
| u_i | Displacement components |
| v_i | Velocity vector components |
| \mathbf{v}^{fl} | Fluid velocity vector |
| Δt | Time step for ABAQUS solver |
| α | Biot-Willis coefficient |
| δ_{ij} | Kronecker delta |
| ϵ_{ij} | Strain tensor components |
| μ | Fluid viscosity |
| ν | Poisson's ratio |
| ν_u | Undrained Poisson's ratio |
| ϕ_i | Growth stimulus at surface node i |
| φ | Dissipation potential due to viscous fluid flow |
| ρ_f | Fluid density |
| σ_{ij} | Stress tensor components |
| ω | Natural frequency ($2\pi \times$ frequency) |
| ζ | Variation of fluid content |

Chapter 1

Background

1.1 Introduction & Motivation

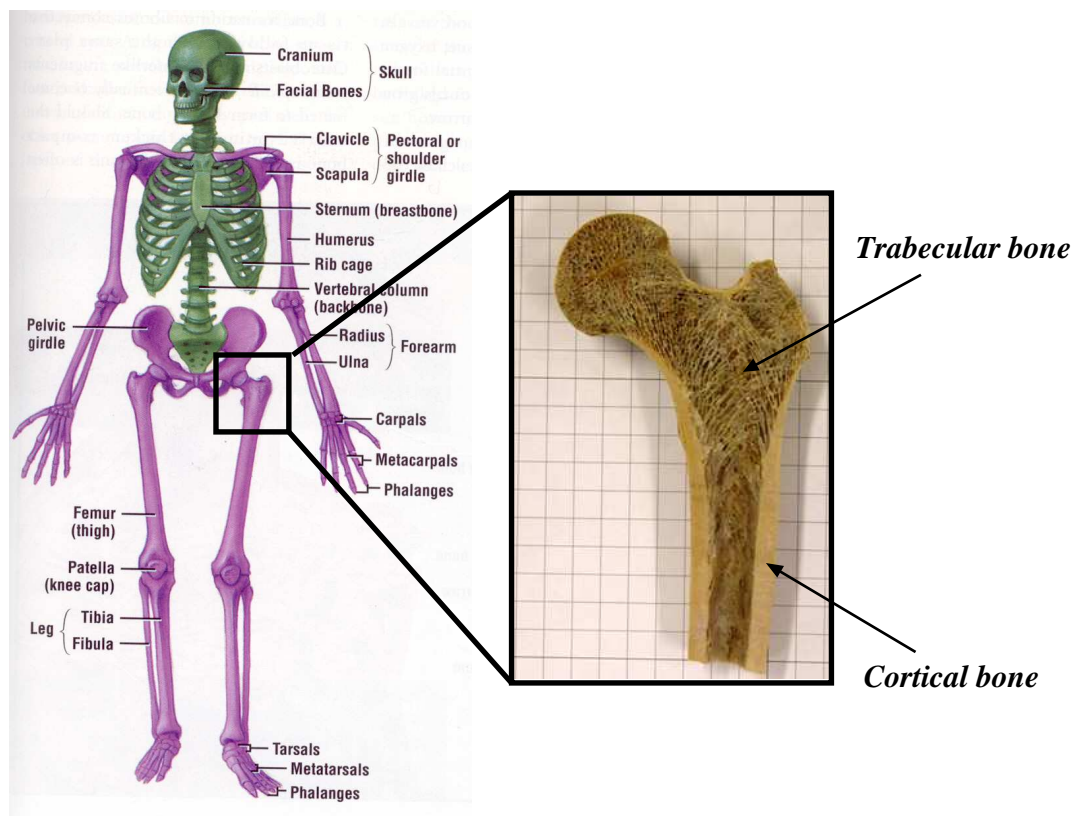


Figure 1.1: Schematic of the skeletal system of the human body, with zoomed in view of the femur showing the trabecular and cortical bone. Images from the bone exhibit of the Smithsonian Institute http://anthropology.si.edu/writteninbone/inside_look.html.

Bones are the dense connective tissues that form the skeletal system in the body. They serve several important functions including supporting and protecting the various vital organs in the body, allowing bodily movement, producing blood, and providing shape and

form to the body. The human body contains a total of 206 bones of different types. Most prominent types of bones (based on their shape) are the long bones (such as femur, ulna, tibia etc) and flat bones (such as skull, spine etc). All types bones are composed of two types of bone tissue: (a) spongy and porous trabecular bone (also referred to as cancellous bone), and (b) hard and dense cortical bone (also referred to as compact bone). Long bones are made up of trabecular bone at the ends (epiphysis), and cortical bone in the middle (diaphysis). Figure 1.1 shows schematically the skeletal system in the human body and the two different kinds of bone tissue namely trabecular and cortical bone. The current work focuses on the adaptation behavior of long bones, focusing on the cortical bone present in the diaphyseal region.

Bone is a living tissue that is constantly changing its size, shape and architecture depending on its current biochemical and/or mechanical environment. The process of adaptation manifests itself in the form of (a) external adaptation, which involves changes in shape of the cortical bone [8], and (b) internal adaptation, which involves changes in architecture and density of the trabecular bone structure [9]. Three types of bone cells have been identified to be responsible for the process of bone adaptation:

- *Osteoblasts* are the bone formation cells that secrete the bone extracellular matrix (collagen) and mineralize it
- *Osteoclasts* are the bone resorbing cells that eat away the bone from the surface onto which they are recruited
- *Osteocytes* are the sensory cells that responds to the external stimuli and activates either the osteoblasts or osteoclasts, depending on the stimuli

It has been shown by a number of researchers over the years that mechanical loading has a profound effect on the adaptation behavior of bone [10]. Disuse or lack of loading causes resorption, whereas applying a mechanical load in addition to physiological loading can stimulate bone formation. Disuse osteopenia can be seen typically in astronauts, who

lose about 20% of their bone mass over a year in outer space, and in bed rest patients, who may lose more than 25% of their bone mass in their lower extremities [11]. Overuse of bone has been shown to result in accumulation of microdamage, which can stimulate removal of the damaged part of the tissue and formation of new bone [12]. Excess bone formation due to overuse is typically observed in athletes, for example bigger and stronger bones on the dominant arms of tennis players compared to their non-dominant arm. Overuse can also lead to fatigue damage, which may result in the formation of microcracks and stress fractures [13].

We conclude that the process of bone growth/resorption in response to mechanical loading is of great significance from both clinical and research perspectives. The interesting interaction between the mechanical environment and the biochemical response of bone, and its significance in real-world applications motivates us to study the problem of bone adaptation in greater detail. Our objective in this work is to build a numerical model that can predict adaptation in cortical bone. A numerical model will be useful in two important ways: (a) it will provide us with a better understanding of the phenomenon of adaptation itself by way of validating the simulation results with experiments, and (b) it can be used as a predictive tool to predict the adaptation response on cortical bone corresponding to a multitude of loading scenarios, for which doing experiments may prove to be too time consuming and costly.

1.2 Literature review: Experimental studies

A number of experiments that try to quantify and better understand the growth response to various mechanical loading parameters have been reported in literature. One of the requirements for the onset of a growth response is the application of a dynamic stimulus, typically applied as a cyclic loading [14]. The growth response has been shown to depend upon a number of mechanical loading parameters such as magnitude and bouts of applied

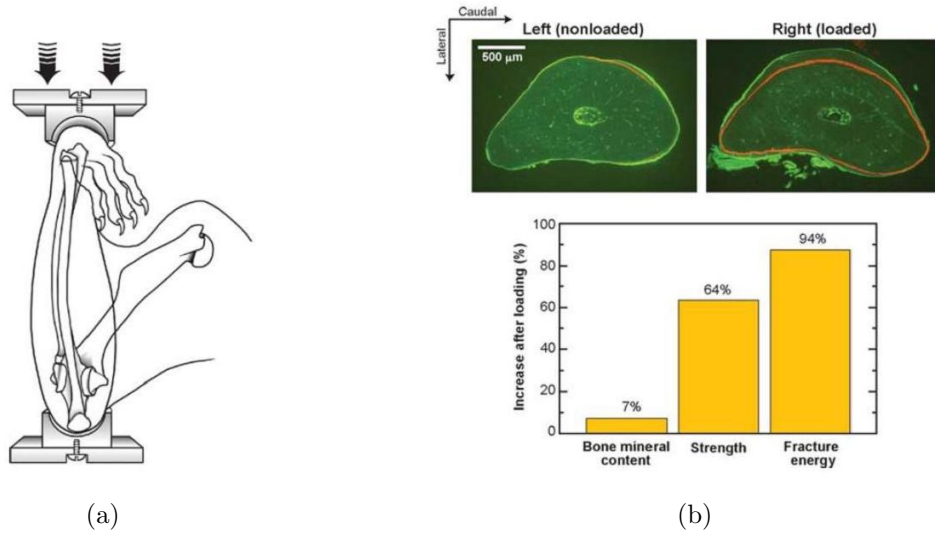


Figure 1.2: (a) Schematic of a typical rat ulna compression experiment, and (b) Typical results from a rat ulna compression experiment: Cross-section on top shows the growth observed on the loaded ulna after 16 weeks, and the plots in the bottom show the observed increase in bone mineral content, strength and fracture toughness of the ulna due to adaptation (from [2]).

load [5; 15; 16], rest periods between bouts of applied loading [1; 6; 17], and loading frequency [3; 4; 18]. All the aforementioned experiments were performed on animals such as rats, sheep, and turkeys. One of the most prominent animal models is the rat ulna or tibia model. The experiments on rat ulnae involve the loading of one ulna (usually the right) in cyclic compression for a certain period of time each day. This loading regimen is carried out for over 16 weeks and the bone formation on the loaded ulna is analyzed. Figure 1.2 shows the setup of a typical experiment and some results from such an experimental study [2]. Similar experiments have also been performed on rat tibia and sheep ulna [18–20].

The growth response of cortical bone varies with the type and mode of application of load. For example, in the study by Akther *et al*, where the rat ulna was loaded in compression at different magnitudes of peak compressive strain, the growth response increased approximately linearly with increasing magnitudes of applied strain loading (Figure 1.3(a)) [19]. In the study by Robling *et al* [1], again on the rat ulna model, they showed that the growth response on a rat ulna to applied compressive loading manifests itself as a significant increase in the

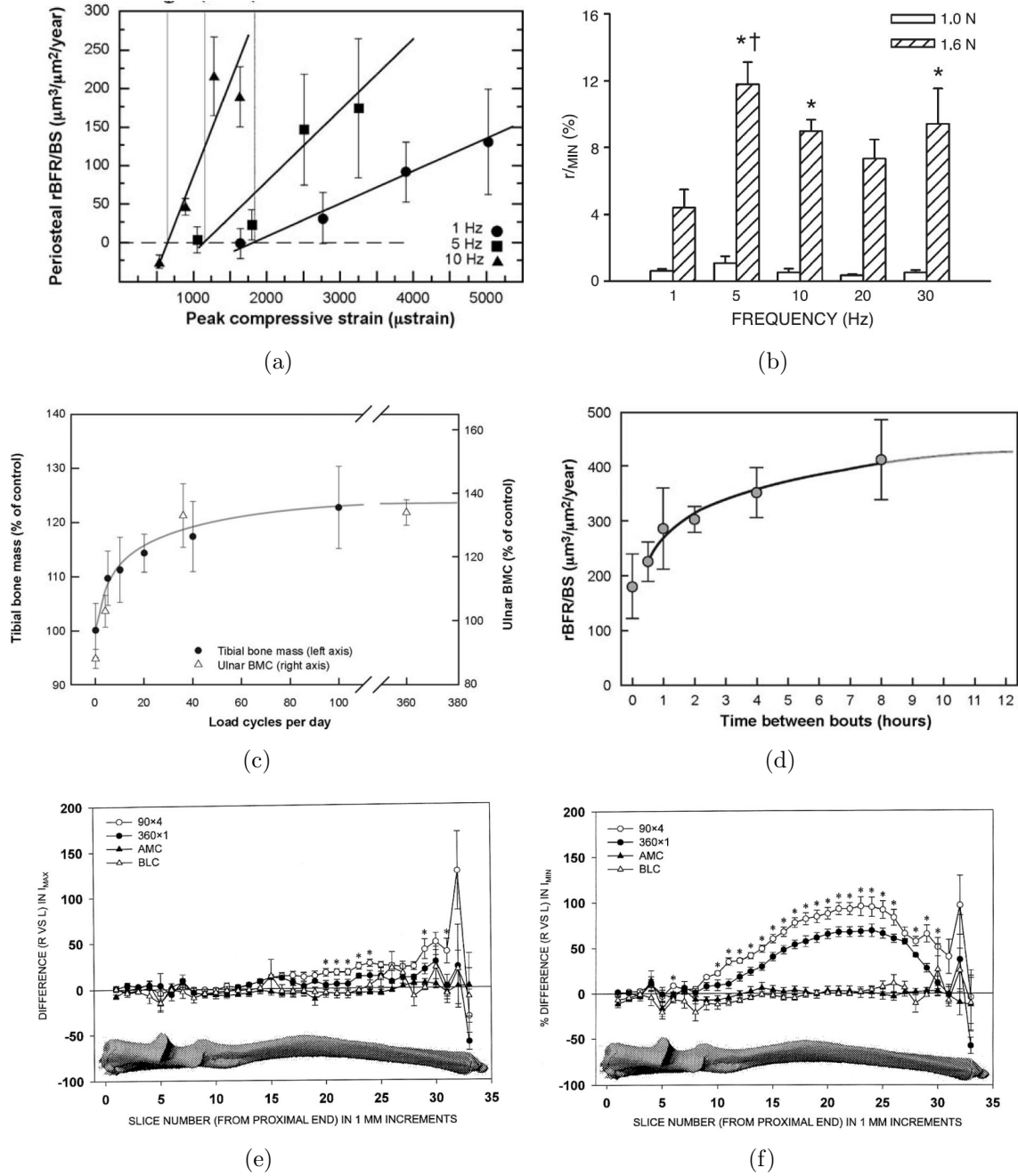


Figure 1.3: Results from different experiments showing the dependence of bone growth on the various mechanical loading parameters: (a) Magnitude of loading [3], (b) Frequency of loading [4], (c) Number of cycles of loading [5], (d) Time between the bouts of applied loading [6], (e) Variation in maximum moment of inertia I_{max} of the cross-section for multiple bouts of loading, and (f) Variation in I_{min} for multiple bouts of loading [1]

minimum moment of inertia I_{min} of the ulna. Further, it was shown that I_{min} increased further when the applied loading was divided into four bouts of 90 cycles each, compared to

loading in a single bout of 360 cycles (see Figures 1.3(e) and 1.3(f)).

Warden and Turner [4] studied the effects of the loading frequency on the adaptation response. They showed that there was no significant growth response for loading frequencies under 0.5 Hz, whereas the growth response increased with frequency from 1 to 10 Hz, but there was no significant increase in the growth response for frequencies greater than 10 Hz (Figure 1.3(b)). Other researchers have looked at the effect of the loading frequency on adaptation on different animal models. Loading sheep and turkey bones at very high frequencies (30 Hz or more) using vibrating tables resulted in an approximately 30% increase in the trabecular bone volume, but did not show a significant response in the cortical bone [18], [20]. We see that the adaptation response of the cortical bone has a non-linear dependence on the loading frequency. Robling *et al* [6] investigated the effect of time between bouts of loading on the adaptation response. They measured the amount of bone formed on the right ulna of a rat due to a series of times of rest between the four bouts of loading of 90 cycles each. The results from these experiments showed that the growth response also has a non-linear dependence on the time between the bouts of the loading. Figure 1.3(d) shows a significant increase in bone growth with increase in the time periods of rest for shorter periods of time between bouts, whereas there is not much increase in the growth for higher periods of rest. These experiments show that the adaptation response is a complicated phenomenon with different kinds of dependence on the various mechanical loading parameters. The results from these experiments establish an important requirement on the numerical model we are trying to develop: it should be capable of predicting the various dependencies of adaptation behavior for different loading conditions.

1.3 Literature review: Numerical models

A numerical model can be used to predict the growth response of the bone tissue to applied loading. Most of the existing mathematical models of bone adaptation follow an

evolution equation of the form [21–25],

$$\frac{db}{dT} = A(\phi - \phi_{ref}), \quad (1.1)$$

where b is a material characteristic (density, mass, architecture or shape), A is a proportionality constant that measures the rate of the response, ϕ is the mechanical stimulus, ϕ_{ref} is the reference mechanical stimulus, and T is growth timescale. Although experiments show that bone adaptation depends on different experimental parameters such as magnitude of load and others, these parameters need to be translated into local mechanical fields in order to apply Equation (1.1). Various mechanical stimuli have been proposed as triggers for bone adaptation, including strain [21], strain energy density [23; 24; 26], tissue damage [25; 27] and different forms of effective stress [28–30].

An important disconnect exists between the experimental, and numerical modeling studies addressing the problem of bone growth. The literature contains many experimental studies of external bone adaptation [1; 3; 5; 6; 15–18; 31], and there are numerous published computational models that attempt to predict the internal adaptation response [23–30]. However, there are relatively fewer models in the literature that concentrate on modeling of cortical bone adaptation [22; 25; 27; 29; 32–35]. Most of these latter works use simplified geometric models of bone, such as circular cylinders or regular cylinders of a complex cross-section. Cowin and Firoozbakhsh [33] modeled external adaptation analytically using a circular cylindrical model, assuming isotropic elastic material properties and strain as the stimulus in the remodeling law. Carter and coworkers developed a daily stress stimulus, based on the number of load cycles and the applied stress, to devise a general theoretical framework for bone modeling simulations [29; 32]. Van der Muelen et al. used the daily stress stimulus and modeled the adaptation process on long bones of circular cross-section [35]. Fridez *et al* [22] examined bone adaptation on a finite element (FE) model, using simple geometries and complex mechanical stimuli based on a plasticity criterion. Prendergast

and co-workers [25; 27] simulated the ulnar osteotomy experiment of Lanyon *et al* [15] on sheep forearms, using a simplified three-dimensional FE model of the radius and ulna, and a damage parameter as the stimulus. Recently, Garcíá-Aznar and co-workers modeled the adaptation of cortical bone using the BEM on simple geometries, such as a rectangular beam and a uniform cylindrical model of a sheep radius [34]. They used a stimulus based on the strain and damage in the remodeling law. The results of their remodeling algorithm were compared qualitatively with the experiments of Lanyon *et al* [15] by comparing the final shape of the computed and experimentally measured cross sections of the adapted cortical bone.

Some of the drawbacks of these models can be summarized as follows:

- External bone adaptation was studied only on simple geometric models.
- Results from the numerical simulations were compared only in a qualitative manner to existing experimental data, without any quantitative comparisons.
- The existing models could not simulate the complicated dependence of the growth response on the various mechanical loading parameters.

These limitations of the existing models call for a systematic approach towards developing a numerical framework for simulating external bone adaptation on real bone geometries, which can predict bone growth that is consistent with quantitative experimental predictions along the whole length of the bone and accommodate the effect of the various loading parameters on the growth response.

1.4 Organization of the thesis

Our objective in this thesis is to extend these previous studies to develop a numerical framework that is capable of predicting external adaptation response on actual cortical bone geometries, using finite element analysis (FEA) coupled with a growth law. We aim to

develop different stimuli that can accommodate the effect of different mechanical loading parameters into the growth law, and validate the results from the simulations with quantitative experimental results. We impose the following requirements on the numerical framework:

- The implementation of the framework should be general, i.e., independent of the complexity of the model, the shape of the elements used in the model etc.
- The growth law should include an appropriate mechanical stimulus that can efficiently simulate the effect of the different mechanical loading parameters.
- The parameters used in the growth law and material model used in the finite element procedure should be consistent with cortical bone biophysics.

This thesis is organized as follows: In pursuit of the development of the numerical model, we first explain in detail the different steps in the development of the finite element model in Chapter 2. We also summarize the general implementation of the surface evolution procedure to simulate cortical bone growth. In Chapter 3, the implementation of the growth algorithm for an elastic material model with strain energy density as the growth stimulus is explained in detail. The results from these analyses are compared to the experiments of Robling *et al* [1]. The results of a parametric study investigating the effect of the parameters of the growth model are explained as well. Chapter 4 deals with the implementation of a poroelastic material model for bone. We explain briefly the development of the field equations of linear poroelasticity, and describe in detail the finite element solution of poroelastic rectangular beam bending. We develop a mechanical stimulus, which can include the effect of loading frequency, based on the results from the poroelastic FE analysis. In Chapter 5, we explain the implementation of the poroelastic material model on the FE model of the rat ulna. This chapter deals with the calibration of the parameters of the growth law by comparing the results from the FE simulations to the results from the experiment of Robling *et al* [1] for the case of loading at 2 Hz, and the adaptation results on the rat ulna for other loading frequencies from 1 Hz to 30 Hz.

Chapter 2

Development of the numerical framework

2.1 Generation of the finite element mesh

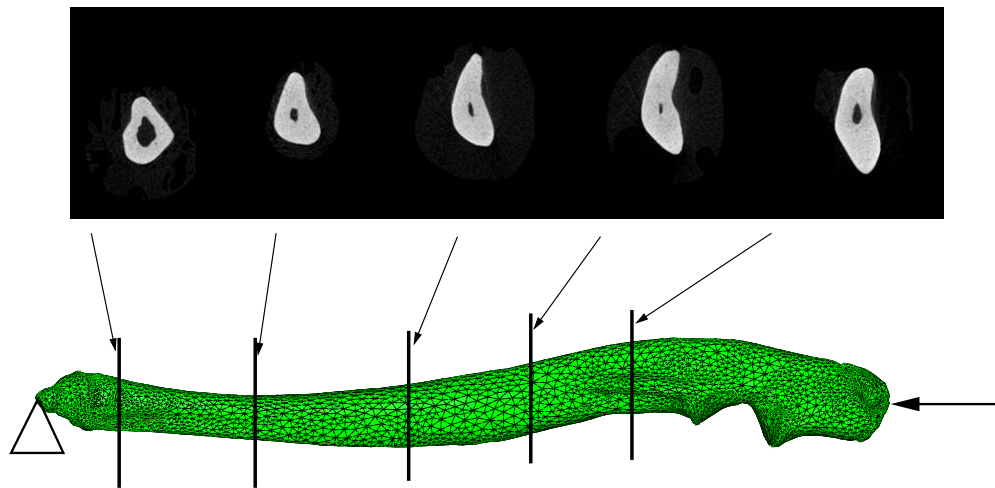


Figure 2.1: FE model of rat ulna generated using HyperMesh from the μ -CT images

We obtained a series of micro-computed tomography (μ -CT) images of the rat ulna from Prof. Charles Turner and co-workers. A brief description of the procedure for obtaining the μ -CT images is given here. The ulna was mounted in the scanning plane of a SkyScan 1172 benchtop CT machine (SkyScan, Aartselaar, Belgium) while bathed in 70% ethanol. The entire bone was scanned every 0.3 degrees using a 60 kV source with a 0.5mm Al filter at 10 μ m resolution. An average of 4 frames per field was used to generate the projections. The projections were reconstructed into 3170 transverse slices using the SkyScan software, with an isotropic voxel size (10 μ m).

The images of the slices were then imported into Amira [36], an image processing pro-

gram for μ -CT images, where a three dimensional surface model of the rat ulna was constructed. The surface model was then imported to the commercial FE meshing software Hypermesh [37]. The surface model was cleaned up in Hypermesh to generate a contiguous two-dimensional surface FE mesh of an enclosed volume. The volume mesh was then generated on the enclosed surface mesh to produce a three-dimensional FE mesh consisting of 37,949 parabolic tetrahedral elements and 62,782 nodes. A selection of μ -CT images at different locations and the final finite element mesh of the rat ulna are shown in Figure 2.1.

2.2 Implementation of surface evolution

We implemented external adaptation by displacing a set of nodes on the outer (periosteal) surface of the rat ulna FE model along the normal direction using an adaptation law similar to equation (1.1). Figure 2.2 shows the flowchart of the algorithm used for the implementation of our numerical framework. The implementation of the surface evolution, as shown in Figure 2.2, is independent of the actual stimulus used in the adaptation law.

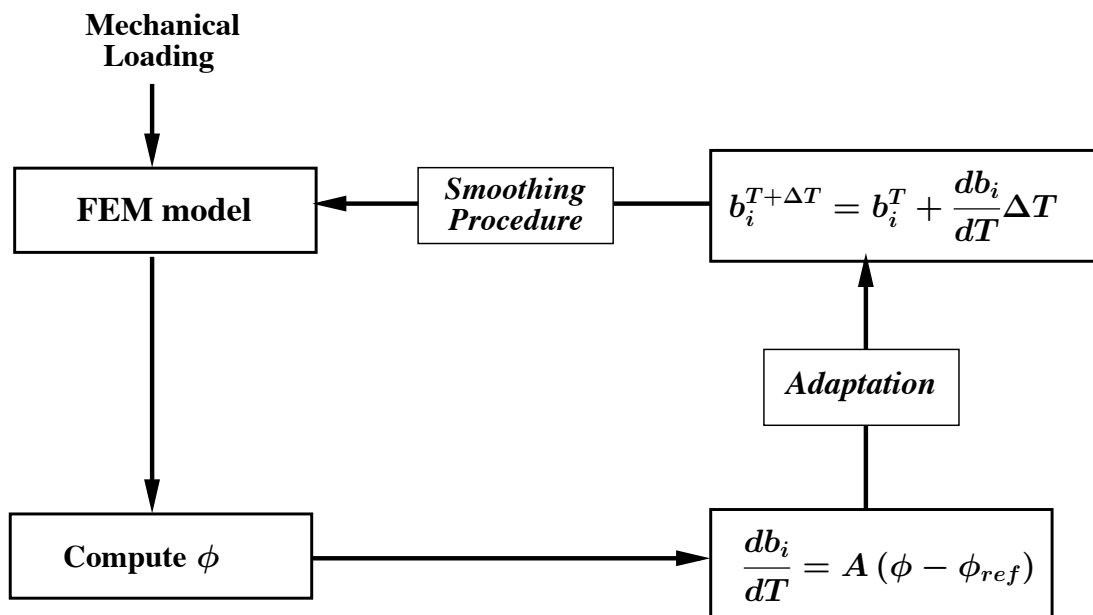


Figure 2.2: Schematic of the flowchart that describes the implementation of the adaptation procedure

The procedure for the simulation of cortical bone adaptation consists of the following steps:

- Solve the BVP of compressive loading of the cortical bone (using appropriate material models) by finite element analysis (FEA)
- Compute the mechanical stimulus at each surface node from the results of FEA
- Compute the magnitude of the incremental displacement of the surface node from the adaptation equation (similar to equation (1.1))
- Evaluate the new positions of the surface nodes based on a forward time-stepping scheme as follows,

$$b_i^{T+\Delta T} = b_i^T + \frac{db_i}{dT} \quad (2.1)$$

- Compute the normal to the surface node from the cross-product as,

$$\hat{\mathbf{n}}_i = \sum_{k=1}^N \frac{\mathbf{a}_k \times \mathbf{b}_k}{\|\mathbf{a}_k \times \mathbf{b}_k\|}, \quad (2.2)$$

where N is the number of elements connected to the surface node i . Figure 2.3 shows the schematic of the procedure for computation of normals using the cross-product. The surface nodes are then displaced along this normal direction.

- Update the FE mesh to account for the normal displacements of the surface nodes as,

$$\mathbf{X}_i^{T+\Delta T} = \mathbf{X}_i^T + b_i^{T+\Delta T} \hat{\mathbf{n}}_i, \quad (2.3)$$

where \mathbf{X}_i is the position vector of surface node i .

- Apply smoothing filter and repeat the algorithm until the required number of growth-timesteps or convergence (when the surface mechanical stimulus becomes equal to the reference mechanical stimulus).

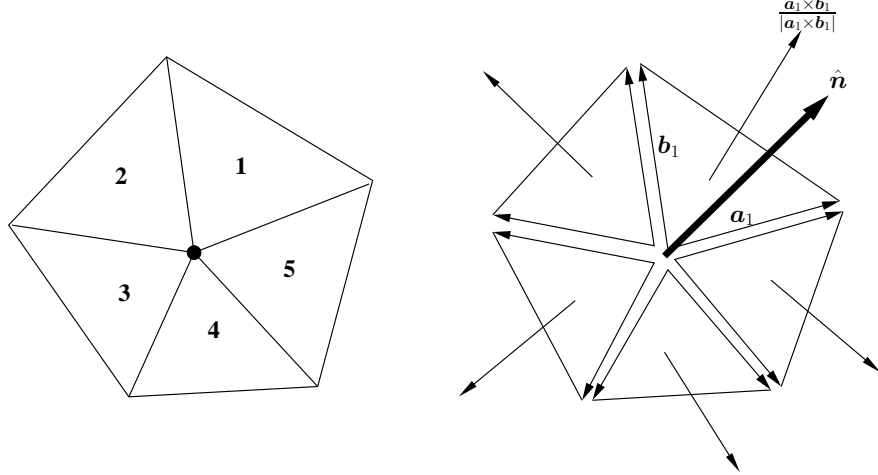


Figure 2.3: Computation of the normal at a surface node using the cross product.

We found that using the displacements given by Eq.(2.1) directly to update the mesh resulted in a jagged surface, because each node is being displaced individually. We therefore used a smoothing filter by defining a distance-weighted average displacement $\hat{b}_i^{T+\Delta T}$ given by,

$$\hat{b}_i^{T+\Delta T} = w b_i^{T+\Delta T} + (1 - w) \frac{\sum_{n=1}^{N_{neigh}} \frac{b_n^{T+\Delta T}}{\|\mathbf{x}_i - \mathbf{x}_n\|}}{\sum_{n=1}^{N_{neigh}} \frac{1}{\|\mathbf{x}_i - \mathbf{x}_n\|}}, \quad (2.4)$$

where N_{neigh} is total number of nearest-neighbors of the surface node i , $\|\mathbf{x}_i - \mathbf{x}_n\|$ is the distance between the nodes i and n , and w is a weighting factor between 0 and 1. We investigated the effect of using different weighting factors on the surface smoothness of the FE mesh. We found that using $w = 1$, i.e., using only the local value of the surface node displacement, resulted in a jagged surface, as seen in Figure 2.4(a). Using a weighting factor of $w = 0$ resulted in a very smooth surface, as seen in Figure 2.4(b), but Equation (2.4) shows that we have completely lost the local displacement information at that particular surface node. In order to maintain a balance between the local displacement at a particular node, and the averaged displacement with its neighbors, we use the value of $w = 0.5$ in all the simulations reported in this work. Figure 2.4(c) shows that smooth FE mesh obtained

using the smoothing filter approach for a weighting function of $w = 0.5$.

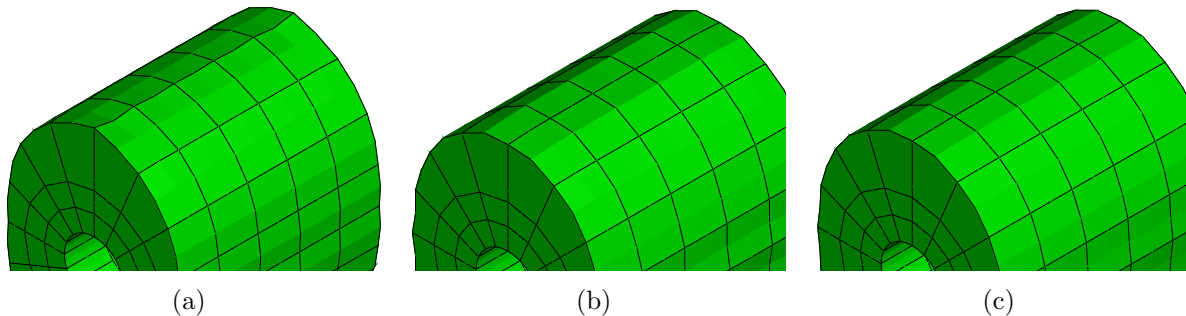


Figure 2.4: Demonstration of the smoothing procedure on a simple circular cylinder model. (a) Jagged surface obtained when using a weighting factor $w = 1$, (b) Smooth surface for a weighting factor $w = 0$, and (c) Smooth surface for a weighting factor of $w = 0.5$, which includes the local information and is also smooth.

Note that we displace only the surface node i by the amount of \hat{b}_i , and the interior nodes are not displaced. This can result in badly shaped finite elements when the magnitude of the surface displacements become bigger. In order to avoid such bad elements in the grown bone model, we undertake the following approach. After computing the surface displacement and the direction of the normal, an intermediate elastic analysis is carried out, in which all of the inner-surface (endosteal) nodes and all nodes on the outer surface that do not participate in the growth process are constrained in all three directions. On the surface nodes where growth is activated, the computed displacement \hat{b}_i is imposed along the normal direction, and we solve this linear elastic displacement boundary value problem (BVP). The computed positions of the interior nodes from this elastic analysis are included into the grown bone FE model. Thus in addition to the smooth surface displacements, the interior node displacements are also smooth resulting in a good quality FE mesh of the growing bone throughout the simulation.

2.3 Elastic properties, boundary conditions and assumptions

In this section, we now focus on the determination of material properties and boundary conditions for the elastic analysis on the cortical bone. We also implement a poroelastic material model to analyze cortical bone, which we will discuss in detail in Chapter 4. We intend to apply our numerical model to simulate the results of the experiments of Charles Turner and co-workers [1; 3; 6; 17].

In the experiments, a compressive load of 17 N was applied to the right forearm of the rat. This load was borne by both the radius and ulna, and we estimate that the ulna itself is subjected to an effective load of 9 N, based on strain gage data from Prof. Turner. Single element strain gages (EA-06-015DJ-120; Vishay Micro-Measurements, Raleigh, North Carolina) were bonded onto both the medial and lateral surfaces of the ulna at distances of 15.85 mm, 11.14 mm, and 6.20 mm from the distal end, using M-Bond 200 adhesive (Vishay Micro-Measurements). Once the gages were attached, the ulna was subjected to cyclic axial compression by an electromagnetic actuator with feedback control (Bose Enduratec 3200, Eden Prairie, Minnesota). Five trials were run per gage location. The average measured strains (in μ strain) on the medial surface were -3710, -3552, and -3064, and on the lateral surface were 3141, 3123, and 2817 at the various gage locations.

Figure 2.1 shows the applied boundary conditions for the stress analysis on the FE mesh. To simulate the experimental loading conditions in the FE model, a compressive load of 9 N is applied on one node on the proximal side along the length of the ulna, i.e., the z -axis (while the displacements along the other two directions at that node are zero), and one node on the distal end is constrained in all three directions. Because the ulna is curved, the compressive load results in bending along the length of the bone. The constrained nodes are selected such that they are in a straight line to ensure that the compressive loading by itself did not induce a moment on the bone.

Young’s modulus (applicable to both elastic and poroelastic material cases) of cortical bone was determined from the experimental strain measurements. We assume that the cortical bone is linear, homogeneous and isotropic. We investigated the strains at the strain gage locations in the FE model for different values of Young’s modulus (E) and a Poisson’s ratio ν of 0.3. We found that for the case of $E = 17$ GPa the strains (in μ strains) were computed to be $-3715, -3491, -3046$ on the medial surface and $2777, 3112,$ and 2751 on the lateral surface.

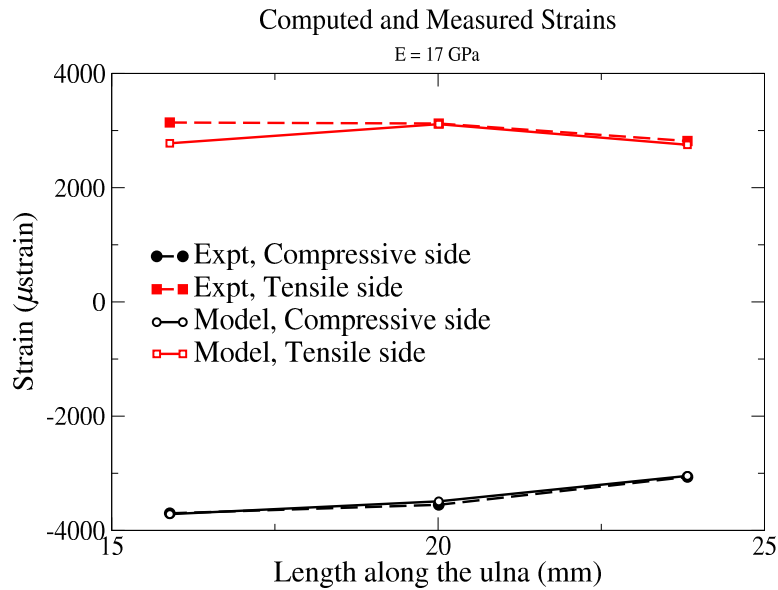


Figure 2.5: Comparison of the rat ulna strain values from experiments and from the FE simulation

Figure 2.5 shows the comparison of the strains observed from the strain gages used in the experiments and from the FE model for a Young’s modulus of 17 GPa. The strains in the FE simulations at five of the six locations match the experimentally measured average strains within less than 2% of the measured strain. The sixth gage, where the measured and computed strains differ by 13%, is at a section of the bone where the strain varies rapidly. The larger discrepancy at this point can be attributed to a small error in exactly locating the sixth strain gage and the fact that the strain gage itself, which is of $5\text{mm} \times 3\text{mm}$, averages the strains over the area. Based on these results, we used the stated constant values for the

elastic modulus and Poisson’s ratio ($E = 17$ GPa, $\nu = 0.3$) in the simulations presented in this work.

Portions of the rat ulna, near both the proximal and distal ends, were porous due to the presence of trabecular bone. In these highly porous regions, Amira[©] was unable to produce a contiguous surface mesh, which resulted in poor quality tetrahedral elements. To circumvent this problem, these regions were filled in with tetrahedral elements, as though the regions were solid, to obtain an analyzable FE mesh. We expected this approach of filling in the trabecular bone region to have very little effect on the adaptation behavior in the diaphyseal cortical bone for two reasons: (a) the trabecular bone is confined to small regions near the ends, and (b) the axial loading is manifested primarily as a bending load in the diaphyseal region. Thus, we modeled the rat ulna as a homogeneous material in this study. Results validating the homogeneous and isotropic assumptions for cortical bone will be presented in Chapter 3.

2.4 Quantification of bone adaptation

In a series of experimental studies by Prof. Turner and co-workers, the growth response was quantified using the change in the minimum moment of inertia (I_{min}) of cross -sections of the ulna along its length, derived from CT images [1]. We follow a similar approach of computing the change in I_{min} in the FE mesh of the grown bone and original bone. Cross-sections of the rat ulna at various positions along its length are obtained from the ABAQUS [38] post-processor using an automated script, and stored as images. The second moments

of area (or moments of inertia) are computed from the images using the following equations:

$$\begin{aligned}
\mathbf{x}_{cg} &= \frac{1}{A_s} \int_{A_s} \mathbf{x} dA = \frac{1}{A_s} \sum_{i=1}^{N_p} \mathbf{x}_i A_i, \\
I_{xx} &= \int_{A_s} y^2 dA - y_{cg}^2 A_s = \sum_{i=1}^{N_p} y_i^2 A_i - y_{cg}^2 A_s, \\
I_{yy} &= \int_{A_s} x^2 dA - x_{cg}^2 A_s = \sum_{i=1}^{N_p} x_i^2 A_i - x_{cg}^2 A_s, \\
I_{xy} &= \int_{A_s} xy dA - x_{cg} y_{cg} A_s = \sum_{i=1}^{N_p} x_i y_i A_i - x_{cg} y_{cg} A_s,
\end{aligned} \tag{2.5}$$

where N_p are the number of pixels in the image, $A_s = \sum_{i=1}^{N_p} A_i$ is the area of the section under consideration, \mathbf{x}_{cg} are the x , y , and z coordinates of the centroid of the section, I_{xx} , and I_{yy} are the second moments of inertia of the section about the x and y axes respectively, and I_{xy} is the mixed-moment of inertia. The principal moments of inertia can be computed from the values of I_{xx} , I_{yy} , and I_{xy} as follows:

$$I_{max,min} = \frac{I_{xx} + I_{yy}}{2} \pm \sqrt{\left(\frac{I_{xx} - I_{yy}}{2}\right)^2 + I_{xy}^2}. \tag{2.6}$$

We use Eqs. (2.5) and (2.6) throughout this thesis to compute the geometric properties of the cross-sections of the rat ulna at different locations along its length and compare them to the corresponding measurements from the experiments of Robling *et al* [1].

Chapter 3

Strain energy stimulus – Growth response

3.1 Introduction

The objective of this chapter is to develop a growth law based on strain energy density stimulus, to combine it with the numerical framework introduced in Chapter 2, and then to use it to simulate the results of the experiments of Robling *et al* [1]. In those experiments, the right ulnae of two groups of rats were subjected to a compressive load of 9 N at a frequency of 2Hz for 360 cycles in one day (a total of 180 seconds a day), every two days for a duration of 16 weeks. One of the groups had all 360 loading cycles applied continuously in one bout (360×1). The other group received the loading in four bouts of 90 cycles each, with a rest period of 4 hours inserted between each bout (90×4). The rats were sacrificed after 16 weeks, their ulnae excised and imaged using quantitative computed tomography (q-CT). The amount of bone formation for each case was quantified by computing the maximum and minimum moments of inertia (I_{max} and I_{min}) of the q-CT images.

Figures 3.1(a) and 3.1(b) show the changes in I_{max} and I_{min} in the ulna due to the applied loading. One can see that the change in I_{max} is much smaller than the change in I_{min} . We can also see that the change in I_{min} is much greater for the 90×4 loading (4 bouts of 90 cycles each) than the 360×1 loading. The authors hypothesized that when the loading was carried out all in one bout, the stimulus-sensing osteocyte cells become desensitized resulting in less bone formation. When the loading cycles were separated into bouts with rest periods, the osteocyte cells become resensitized, leading to greater bone growth for the same total number of load cycles. We describe in this chapter how this effect of the loading bouts on

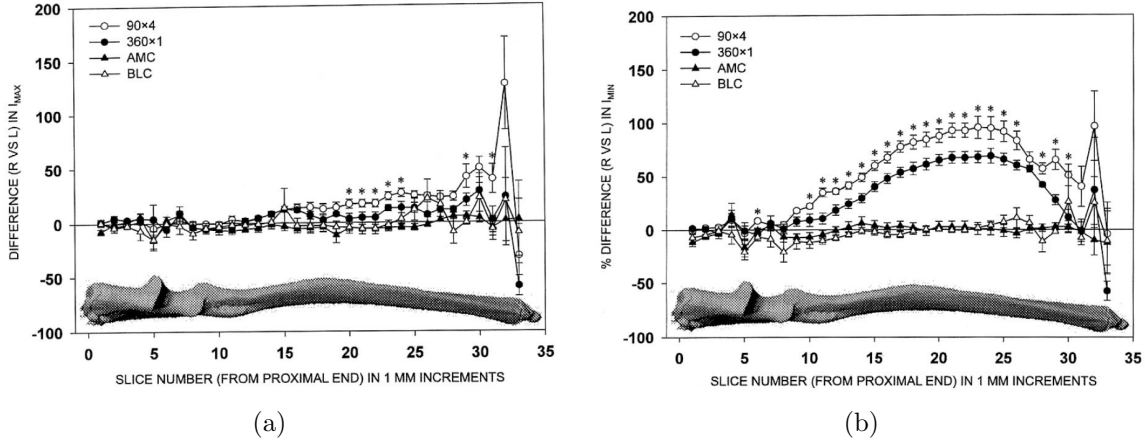


Figure 3.1: Results from the experiments of Robling *et al* showing (a) the percentage increase in the maximum moment of inertia (I_{max}), and (b) the percentage increase in the minimum moment of inertia (I_{min}).

adaptation response can be incorporated into the numerical framework. Note that most of the material in this Chapter is derived from our publication “Numerical Modeling of Long Bone Adaptation due to Mechanical Loading: Correlation with Experiments” in *Annals of Biomedical Engineering*, March, 2010 [39].

3.2 Growth Equation

We use strain energy density as the mechanical stimulus that triggers bone adaptation, because it is a single scalar quantity that combines both stress and strain, and it has been used by many researchers previously [23; 24; 26]. We change the adaptation law given in Eq. (1.1) slightly to preclude resorption, since no resorption was observed in the experiments of Robling *et al* [1]. The adaptation equation used is given by,

$$\frac{db_i}{dT} = \begin{cases} A(\phi_i - \phi_{ref}) & \phi_i \geq \phi_{ref} \\ 0 & \phi_i < \phi_{ref} \end{cases} \quad (3.1)$$

where b_i is the normal displacement of surface node i , A is a proportionality constant, ϕ_i is the local strain energy density, and ϕ_{ref} is the reference strain energy density. The nodal

values of strain energy density ϕ_i were computed by extrapolating the computed strain energy density at the element integration points to the nodes. The proportionality constant A (with units $\text{m}^3/\text{N}/\text{timestep}$) plays the role of a “gain” in the adaptation process, whereas the sensitivity of the osteocytes to mechanical stimulation is represented by the reference strain energy density ϕ_{ref} . Hereafter, we use the terms “gain” and “threshold-sensitivity” respectively to refer to A and ϕ_{ref} .

It is important to understand that there are three different time scales in the experiments: (a) the time scale of the loading cycles, on the order of seconds; (b) the time scale of the bouts over which the load is administered, varying from minutes to hours; and (c) the time scale for growth and adaptation, which was 16 weeks in the experiments of Robling *et al* [1]. In this work, the time T in Eq. (3.1) refers to (c), the growth/adaptation time scale. There were no measurements of the temporal evolution of adaptation in the experiments of Robling *et al* [1], so the growth-timestep T should be viewed as a pseudo-timestep. We chose the growth-timesteps such that the total adaptation occurred over 20 timesteps.

Scaling Eq. (3.1) in terms of a different pseudo-time variable $\tau = cT$, where c is a constant, and applying the chain rule for differentiation, the adaptation law becomes,

$$\frac{db_i}{d\tau} = \begin{cases} \frac{A}{c}(\phi_i - \phi_{ref}) & \phi_i \geq \phi_{ref} \\ 0 & \phi_i < \phi_{ref} \end{cases} \quad (3.2)$$

Thus, we see that A and T are convolved, and choosing the time step size such that the growth occurs over a fixed number of time steps (usually 20) ensures that the results are independent of the scaling. It should be noted that the form of the growth law in Eq. (3.1) admits the possibility that given sufficient time to adapt, the bone will eventually grow enough that $\phi_i = \phi_{ref}$ everywhere, i.e., a steady-state or saturation is achieved. None of the simulations reported in the next sections reached steady-state.

The goal of this Chapter is to combine the growth Eq. (3.1) with the methods described in Chapter 2 to simulate the growth behavior of the rat ulna model. We next describe

a series of growth simulations on the rat ulna FE model with different values of the gain (A) and threshold-sensitivity (ϕ_{ref}) parameters, and validate the simulation results with the experimental results of Robling *et al* [1].

3.3 Bone growth simulations

In this section, we attempt to address the following important questions:

- (a) How do the number of timesteps, and the gain and sensitivity parameters, in the growth law affect the growth response ?
- (b) Can the mathematical model simulate the results of the experiments of Robling *et al*, or in other words, can the numerical model can be used to interpret the experiments ?
- (c) What is the biological significance of the two growth law parameters ?

We conducted a series of adaptation simulations on the rat ulna FE model to answer these questions, the results of which are described below. We first present the results obtained for a set of baseline growth law parameter values and then investigate the effect of systematic variation of the parameters on the adaptation response. The baseline value of ϕ_{ref} was determined using the data of Hsieh and Turner [3], who reported a threshold strain for adaptation of 1820 μ strain for a rat ulna loaded at 9 N at 1 Hz. The approximate strain energy density corresponding to this strain value was found to be 2.7×10^4 J/m³ (using the 1-D approximation of strain energy density as $\phi = 1/2E\epsilon^2$). Based on this calculation, we simulated adaptation for $\phi_{ref} = 1 \times 10^4, 2 \times 10^4, 4 \times 10^4, 5 \times 10^4$ J/m³ taking the value of 3×10^4 as the baseline. The baseline value of $A = 1 \times 10^{-10}$ m³/N/timestep was determined by numerical experimentation to find a value that gave significant growth in 20 timesteps. As noted earlier, time should be regarded as a pseudo-time in this analysis. For all of the bone adaptation simulations, a weighting factor w of 0.5 was used in Eq. (2.4) for the smoothing and the simulations were carried out to a maximum of 20 timesteps. We followed the same

procedure to quantify the growth response as used in the experiments by Robling *et al* [1], by computing the percentage change in geometric properties such as cross-sectional area, and minimum and maximum principal moments of area (I_{max} and I_{min}) from pixelated images of the sections along the length of the FE model of the rat ulna.

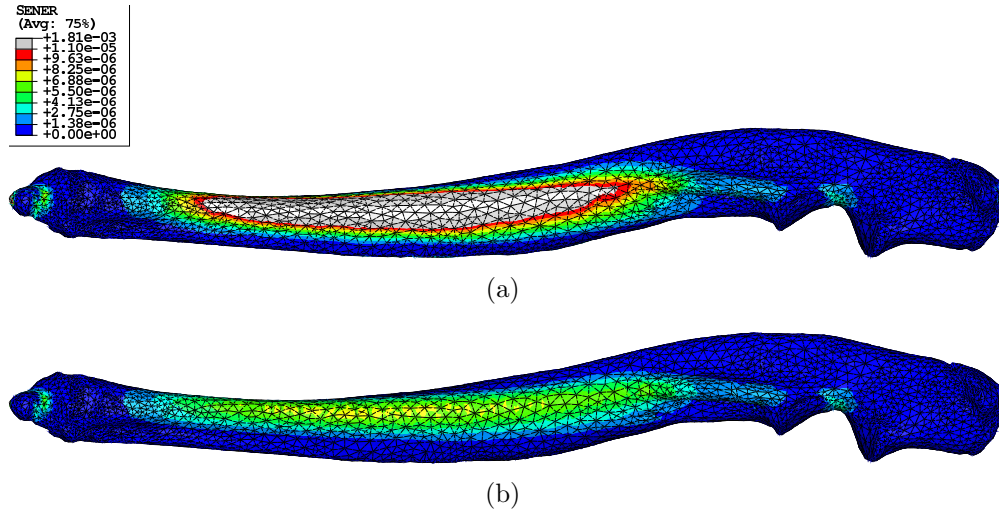


Figure 3.2: Contours of strain energy density in the rat ulna (a) before growth, and (b) after 20 growth timesteps.

The distribution of strain energy density in the ulna before growth is shown in Figure 3.2(a), and that of the adapted bone at the end of 20 growth timesteps using the baseline parameter set is shown in Figure 3.2(b). The regions shown in grey correspond to elements whose strain energy density exceeds $1.1 \times 10^5 \text{ J/m}^3$. One can clearly see that the strain energy density is reduced in the adapted ulna, due to the added bone. This indicates that adaptation enables the bone to resist the applied bending load. Figure 3.3(a) shows an overlay of the computed cross sections of the original and adapted bone at the ulna midshaft (dashed and solid lines, respectively), and Figure 3.3(b) shows a similar overlay at the ulna midshaft obtained in the experiments of Robling *et al* [1]. This result shows that the simulation captures the adaptation observed in the experiments at least qualitatively.

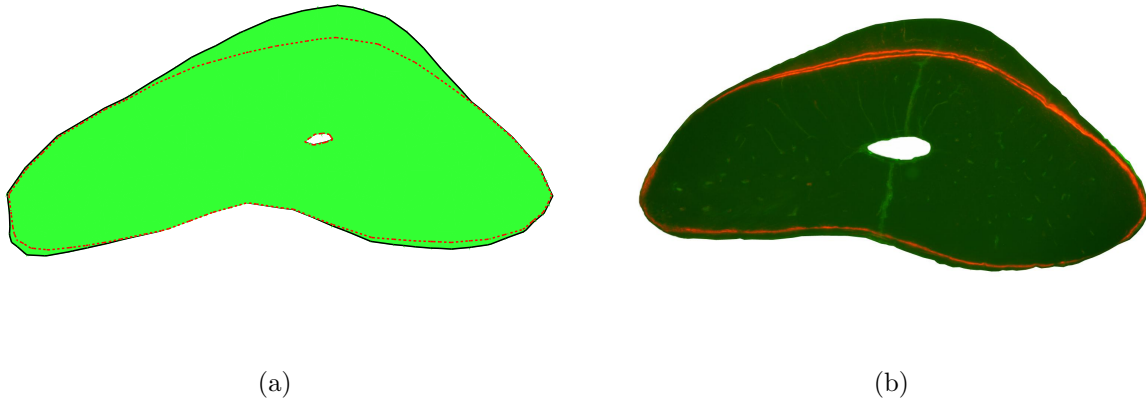


Figure 3.3: Comparison of the cross sections between the original and grown rat ulna at the midshaft from (b) our simulations and (a) the experiments of Robling *et al* [1].

3.4 Parametric study of timesteps, gain and threshold-sensitivity

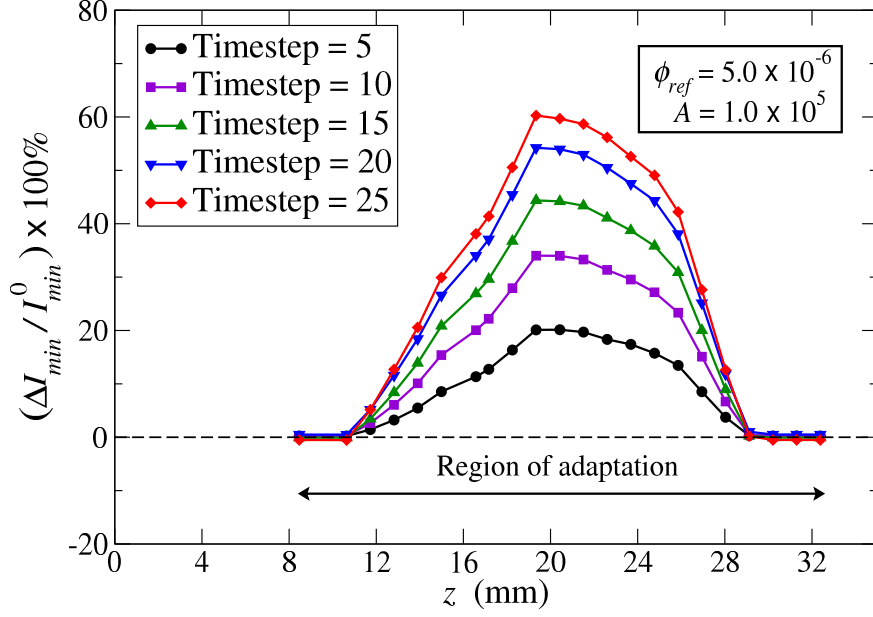
We now summarize the results from our parametric study showing the effect of variation of timesteps, gain parameter A , and the threshold-sensitivity parameter ϕ_{ref} on the adaptation response.

(a) Effect of number of growth timesteps

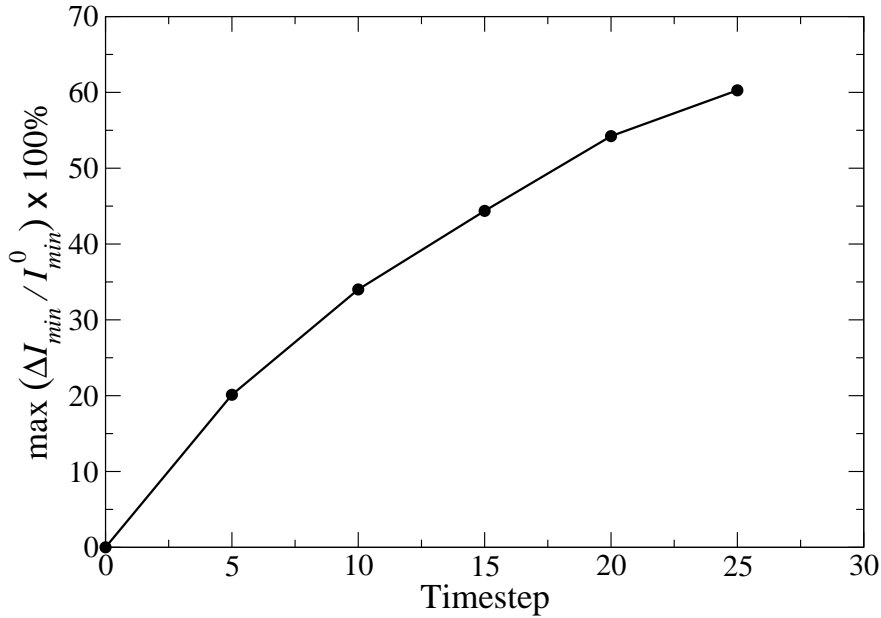
Figure 3.4(a) shows the variation of the adaptation response of the ulna at several growth-timesteps in the adaptation simulation. We can see that as the number of timesteps increases, the percentage change in I_{min} tends to saturate. This can be seen more readily in Figure 3.4(b), which shows the evolution of the peak height. The saturation in the response occurs because as the bone grows, the strain energy density decreases, leading to a slower growth rate.

(b) Effect of gain parameter

The effect of the gain parameter on the growth response was investigated by carrying out the adaptation simulations for a sensitivity of 3×10^4 J/m³, five different values of the gain parameter A . The percent change in I_{min} for the different values of A is shown in



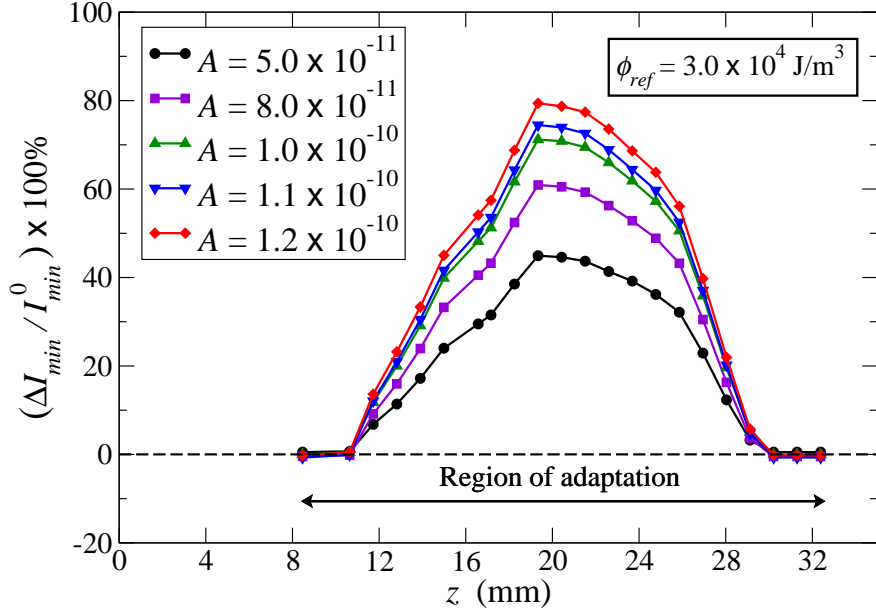
(a)



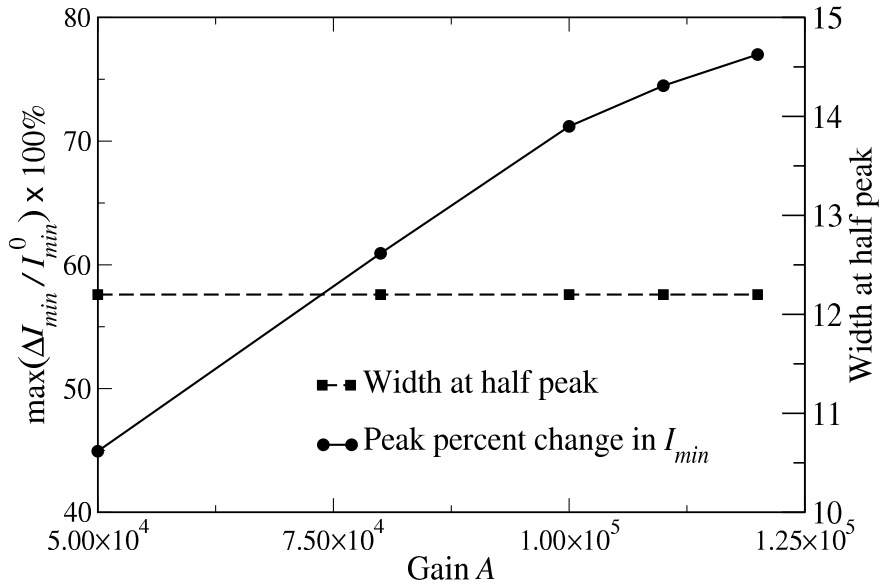
(b)

Figure 3.4: (a) Variation of increase in I_{min} for the baseline parameters of the gain and sensitivity parameter for different timesteps, and (b) change in the maximum increase in I_{min} with increasing values of timesteps.

Figure 3.5(a). We can observe from Figure 3.5(a) that as the value of gain parameter A increases, the percentage increase in I_{min} increases in a non-linear fashion. Specifically, we can see that the variation of the maximum percent increase in I_{min} tends to saturate for



(a)



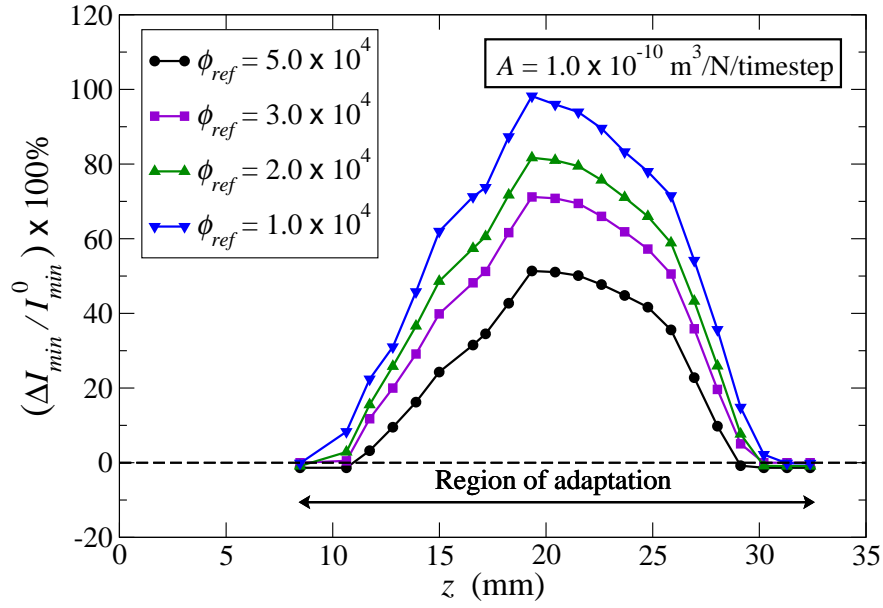
(b)

Figure 3.5: (a) Variation of increase in I_{min} for different values of the gain parameter for the baseline value for the sensitivity parameter, and (b) change in the maximum increase in I_{min} with increasing values of the gain parameter A .

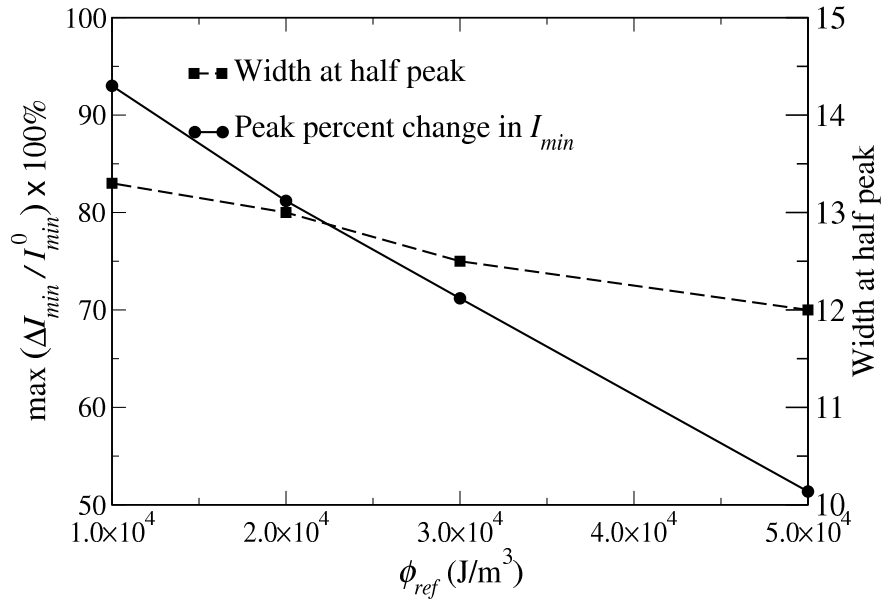
increasing values of gain parameter A (Figure 3.5(b)). This saturation in the adaptation response occurs because as A increases, the strain energy density stimulus decreases, thus leading to slower bone growth rate. The same behavior was observed in the change in I_{min}

with increasing timesteps.

(c) Effect of threshold-sensitivity



(a)



(b)

Figure 3.6: (a) Variation of increase in I_{min} for different values of sensitivity parameter ϕ_{ref} for the baseline value of the gain parameter A , and (b) change in the maximum increase in I_{min} with increasing values of sensitivity parameter ϕ_{ref} .

We investigated the effect of the sensitivity parameter on the growth response using

a series of simulations carried out for 20 timesteps and a fixed value of A of 1.0×10^{-10} $\text{m}^3/\text{N}/\text{timestep}$, for five different values of the threshold-sensitivity parameter. Figure 3.6(a) shows the effect of variation of ϕ_{ref} on the bone adaptation response. One can clearly see that with decreasing values of reference strain energy (ϕ_{ref}), there is a corresponding increase in the percent change in I_{min} in the diaphyseal region. From Figure 3.6(b), it can be seen that the peak percent change in I_{min} varies linearly with decreasing values of sensitivity. In addition, the curves widen approximately linearly with decreasing values of ϕ_{ref} , unlike the gain parameter A . This can be visualized better in the plot in Figure 3.6(b), which shows increase in the “half-peak-width” of each of the curves in Figure 3.6(b) for decreasing values of sensitivity. The increase in half-peak-width clearly indicates that the region of adaptation extends towards both ends of the bone as the threshold sensitivity decreases.

3.5 Validation with experiments

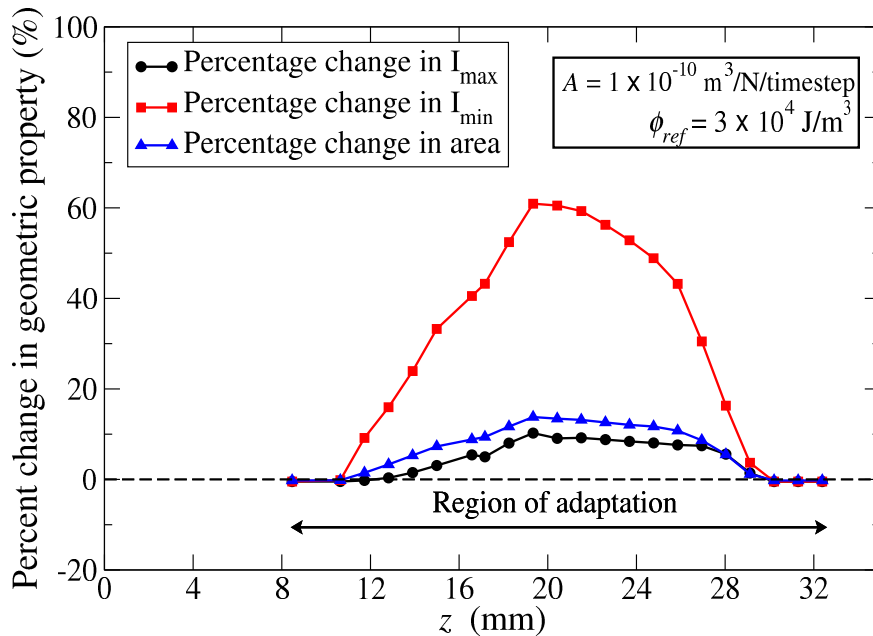


Figure 3.7: Plot showing the increase in the different geometric properties of the grown ulna due to adaptation. It can be observed that the increase in I_{min} is the most significant.

One of the basic observations from the experimental results was that bone growth in-

creased the minimum principal moment of inertia I_{min} more significantly than the maximum principal moment of inertia I_{max} . This is to be expected because it is well known that the stress due to unsymmetrical bending is given by,

$$\sigma_{zz} = -\frac{M_{x'}y'}{I_{x'}} - \frac{M_{y'}x'}{I_{y'}},$$

where x' and y' are the coordinates along the principal axes, and $M_{x'}$ and $M_{y'}$ are the components of the applied moment vector about those axes. Considering x' to be the minimum principal axis, we see that the value of the stress is dominated by the term $M_{x'}y'/I_{x'}$. Thus, it is reasonable to expect the bone to respond to the applied loading by an increase in I_{min} in order to decrease the mechanical stimulus. Based on the numerical results, Figure 3.7 shows the distribution of percentage change in these geometric properties, comparing the original and grown ulna using the baseline values of the parameters for 20 timesteps. We can see that the change in I_{min} is much more significant than the changes in I_{max} and area, which is consistent with the experimental results.

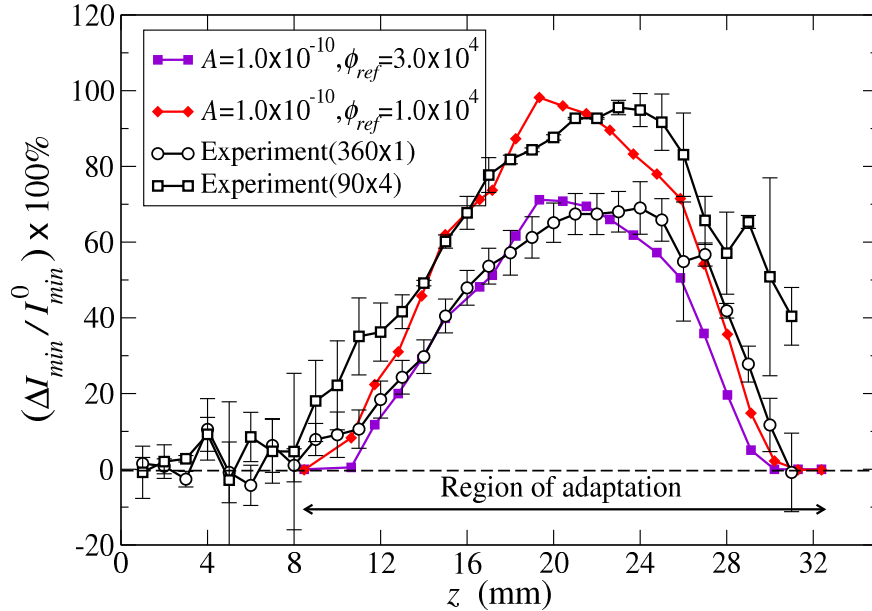


Figure 3.8: Plot showing the increase in the I_{min} due to adaptation from the numerical model compared to the experimental results.

Further validation of the model can be obtained by direct comparison of the numerical and experimental results. Figure 3.8 compared the percent change in I_{min} observed from the simulation for different values of gain and sensitivity, with those observed in the experiments of Robling *et al* [1] for different bouts of loading. The filled symbols in Figure 3.8 represent the results from the simulation obtained using a gain A of $1.0 \times 10^{-10} \text{ m}^3/\text{N}/\text{timestep}$ and two different values of threshold-sensitivity ϕ_{ref} , whereas the open symbols correspond to the results from the experiments. We found that we could fit the experimental results corresponding to the adaptation observed for a single bout and four bouts of loading by changing only the sensitivity from 3.0×10^4 to $1.0 \times 10^4 \text{ J/m}^3$ in the simulations. We also

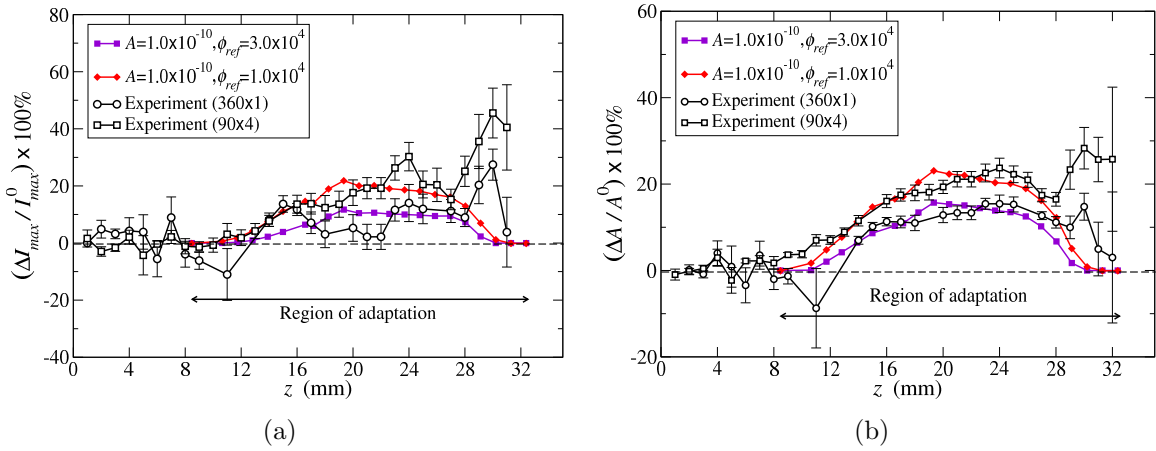


Figure 3.9: Plot showing the increase in the I_{max} and area due to adaptation from the numerical model compared to the experimental results.

compare the percent change in I_{max} from the simulation and experiment in Figure 3.9(a), and the percent change in area in Figure 3.9(b). These plots demonstrate the ability of the numerical model to quantitatively predict the surface-specific growth response of the cortical bone over the whole length of the bone.

3.6 Biological significance of growth law parameters

The numerical results discussed in the previous section can shed more light on the biomechanical relevance of the two parameters in the growth law. We found that increasing the gain A increases the rate of adaptation (db_i/dT) at any given location of the bone, as long as the mechanical stimulus there exceeds the threshold-sensitivity ϕ_{ref} . This is demonstrated, for example, in Figure 3.5(b), which shows that the width at half-peak in the percent change in I_{min} is independent of A . Thus, we identify A as a parameter that amplifies the response of the mechanosensory system (primarily osteocytes) that senses the effect of the loading. We also note that A is a conjugate of the pseudo-time T , so that increasing A (for fixed T) or T (for fixed A) are equivalent, and hence the gain parameter, when set in appropriate units of time, will be able to differentiate between different duration of experiments (e.g., 20 weeks opposed to 16 weeks). We also observe that the growth response begins to saturate for large A .

The second parameter, ϕ_{ref} , affects the adaptation in a different way than A . As ϕ_{ref} decreases, not only is there an increase in the rate of adaptation response at any given location, but also an expansion of the domain over which adaptation occurs. This is illustrated in Figure 3.6(b), which shows that the width at half-peak in the percent change of I_{min} increases as ϕ_{ref} decreases. Thus, we identify ϕ_{ref} as the parameter that quantifies the threshold response of the mechanosensory system, which triggers bone growth.

Our parametric study shows that similar adaptation response can result from different combinations of the gain and threshold-sensitivity parameters. The threshold-sensitivity parameter ϕ_{ref} can be determined uniquely based on the threshold strain for adaptation and the loading history (e.g., the number of bouts of loading, as pertaining to this study). We have also seen that the growth-timestep and the gain parameter A are convolved. Thus, in order to establish a unique value of A , the growth-timestep needs to be calibrated. More experimental results quantifying the growth of the periosteal surface over time (say the

growth of bone per day or week) would permit one to obtain a unique value for the gain parameter A .

3.6.1 Significance of threshold-sensitivity ϕ_{ref}

It is well known that most cell types sense and respond to their mechanical environment by reorganizing their cytoskeleton [40; 41]. This changes the cell's sensitivity, and hence its response, to the mechanical environment. Since osteocytes are widely considered to be the sensor cells in a bone matrix, any changes in the external loading conditions alter its mechanical environment, which can result in a change in mechanosensitivity. In the experiments of Robling *et al* [1], more growth was observed when the load cycles were divided into four bouts, while maintaining the magnitude and frequency of loading the same. Cellular de-sensitization, due to continuous loading, resulted in lower growth, whereas cellular re-sensitization, due to the rest periods, was hypothesized to be responsible for the improved bone adaptation response.

We have shown that the numerical model can predict the re-sensitization of osteocyte cells associated with the separated bouts by decreasing only the threshold-sensitivity parameter ϕ_{ref} from 3×10^4 to 1×10^4 , while maintaining $A = 1 \times 10^{-10}$ fixed (Figure 3.8). Based on this observation, the adaptation response of bone as a result of cellular level stimulation (desensitization and re-sensitization) can be captured in the numerical model through the threshold-sensitivity parameter ϕ_{ref} alone. Thus, the threshold-sensitivity parameter ϕ_{ref} is a quantitative measure of the mechanosensitivity of the osteocytes to their extracellular mechanical environment. Also, since the threshold-sensitivity changes from 3×10^4 to 1×10^4 for the two cases of loading with one bout and with four bouts, respectively, we can deduce that the threshold-sensitivity is determined by the loading history.

3.6.2 Significance of gain A

The gain parameter A amplifies the growth for any given value of the threshold-sensitivity parameter ϕ_{ref} . We have shown that varying the gain parameter A does not result in the expansion of the adaptation region, a characteristic effect of changes in threshold-sensitivity. Thus the gain parameter is not a measure of the mechanosensitivity of the osteocytes, but rather it must be a measure of other biological parameters that can amplify the adaptation response, such as the amount of osteoblastic or osteoclastic activation by the osteocyte, the amount of available nutrients for bone formation, etc. We have shown that the value of the threshold-sensitivity used is consistent with the threshold strains observed previously. A biologically consistent value for the gain parameter A could be determined from an experiment that measures the rate of bone surface expansion, e.g., using fluorochrome labels, and fitting the results of such an experiment to the growth law.

Since the model parameters incorporate biological information about the cellular mechanical environment and the current health status of the bone tissue (and body as a whole), more experiments are needed to quantify these parameters, for the model to be used as a predictive tool. For example, experiments can be designed to quantify the growth response to different diets, drugs, diseases, and other factors, and these experiments can inform the numerical model on how the growth law parameters are affected by each of these biological parameters. While the threshold-sensitivity parameter is a convenient way to quantify the biomechanical status of the osteocyte, we need more experiments both at the lengthscale of the bone and at the lengthscale of the osteocyte cell itself to obtain a better understanding of process of mechanotransduction of the osteocyte and subsequent bone adaptation. Although some cellular level mechanical models already exist in literature[42; 43], there is no clear experimental evidence connecting the mechanosensitivity of the osteocyte cells to bone adaptation response at the cellular lengthscale. These are some of the open issues that remain to be addressed.

3.7 Validation of assumptions - homogeneity and isotropy

We made several simplifying assumptions in performing the FE analyses, which bear further examination. In particular, we consider the importance of filling-in the trabecular regions with solid elements (refer to Chapter 2), and the assumption of anisotropy in the modulus.

To examine the effect of filling in the trabecular region with solid elements, we performed simulations with the value of Young's moduli in these regions reduced to 5 GPa and 10 GPa, respectively, while maintaining the baseline value of 17 GPa in the central cortical bone. In both of these simulations, the adaptation response was exactly the same as that computed using Young's modulus of 17 GPa everywhere. This result is not too surprising for several reasons: (a) the load was applied as an axial force, which implies that the bending stresses in the diaphysis region were unaffected by the reduction in modulus; (b) the trabecular region is small and confined to the ends of the bone so that there is little effect on the stress and strain energy density solution in the diaphyseal region; and (c) although the strain energy density in the trabecular region did increase slightly, it never reached the threshold-sensitivity, whereas the strain energy density in the diaphyseal region remained unaffected.

In order to assess the possible effect of transverse isotropy on the adaptation response, we performed a simulation using 17 GPa as the longitudinal modulus and 9.5 GPa as the transverse modulus for the cortical bone. We observed about 2% difference in adaptation in distribution of percentage change in I_{min} between the cases of isotropic and transversely isotropic properties. This result is also not surprising because the bending stress is the major contributor to the strain energy density stimulus, and the magnitude of bending stress is governed mainly by the value of the longitudinal stiffness. Thus we see that our assumptions of homogeneity and isotropy are reasonable for the present study.

3.8 Summary

We have successfully combined a strain energy density based growth law with the numerical framework developed in Chapter 2 to simulate the process of external bone adaptation with respect to applied mechanical loading, using a homogeneous, isotropic, linear elastic material model for the cortical bone. We have explored the individual effects of the two parameters in the remodeling law, gain and threshold-sensitivity, on the cortical bone adaptation behavior. The effects of magnitude and bouts of bending load on cortical bone adaptation in rats can be modeled using one of the simplest of material models (isotropic, linear elastic material model) and a simple adaptation law, to a good degree of accuracy. Discrepancies between model predictions and experimental data were significant only near the distal end of the ulna. These can be attributed to osteophyte (bone spur) formation, which is a pathological process observed in the experiments of Robling *et al* [1]. Since osteophyte formation is not due to adaptation, we cannot expect the model to predict this elevated growth near the ends of the ulna [1].

An important requirement for the onset of bone adaptation is the presence of dynamic loads, rather than static loads, as pointed out by a number of researchers [14; 29; 32; 35; 44]. It has also been shown that cyclic loading in synergy with cellular stimuli (such as growth factors) result in much higher adaptation response [44]. One of the important shortcomings of this model is that using strain energy density as the stimulus precludes modeling of the effect of frequency, and other time-dependent parameters. The next two chapters will describe a different material model for cortical bone adaptation including the development of a frequency-dependent mechanical stimulus and its integration into our numerical framework to account for the effect of frequency of loading on the adaptation response.

Chapter 4

Poroelasticity – Rectangular beam model

4.1 Introduction

We, in Chapter 3, used our numerical model to simulate the effect of magnitude and bouts of loading on the adaptation response, using elastic material and strain energy density stimulus. The objective of this chapter is to develop a stimulus that can include the effect of loading frequency on adaptation. In Chapter 1, we established the importance of the presence of a dynamic stimulus (i.e., cyclic loading) for cortical bone adaptation. We present here again a brief review of the experimental results. Turner *et al* [45] observed significant cortical bone adaptation when the loading frequency exceeded 0.5 Hz. At low frequencies (between 0.5 and 10 Hz), it was observed that adaptation in rat ulnae follows an approximately linear dose-response relationship with frequency [3]. Warden and Turner [4] found no significant increase in the adaptation response when the frequency of loading was increased beyond 10 Hz. Rubin and coworkers investigated the effect of very low magnitude, high frequency loading (greater than 30 Hz) on sheep standing on a vibrating plate 20 minutes per day for a year. They found that trabecular bone volume increased more than 30%, but no significant changes were found in the cortical bone [20; 46]. Qin *et al* showed that cortical bone responds to applied pressure gradients in the intermedullary fluid through periosteal bone formation [47–49], without any applied mechanical loading. These experimental results indicate that a material model more complicated than a simple linear elastic material is needed.

Several groups have examined the relationship between mechanical loading of cortical bone and the associated fluid flow inside the various internal porosities. Knothe Tate and

Knothe observed the fluid flow into and out of cortical bone in sheep forearms under applied load [50]. Knothe Tate *et al* showed similar load-induced fluid flow in rat tibia subjected to bending, and also indicated the role of the fluid flow in mechanotransduction [51]. This phenomenon has been explained by numerous authors as a direct mechanical effect on the osteocytes, such as by shear stress on the cell surface[52], drag force on the transverse fibrils that tether the osteocyte to the canalicular walls [42], strain generated electric potentials [53; 54], and biochemical diffusive gradients [55]. These studies motivate the use of poroelastic material model of cortical bone.

4.1.1 Porosities of cortical bone

Cortical bone is a porous structure with different porosities at different scales. These different porosities provide an exchange medium between the osteocytes and the blood stream. We briefly describe here the mesostructure of cortical bone and the associated porosities through which bone fluid and blood can flow. Figure 4.1 shows the schematic of the various porosities on the cortical bone structure on the macro and microstructural levels.

Vascular porosity which comprises the cylindrical pathways of blood flow in the Haversian and Volkmann canals that form the osteonal structure of the cortical bone. The fluid permeability at this scale is of the order of 10^{-12} m^2 [56]. This porosity plays an important role in adaptation by delivering nutrients to the cells that form or resorb bone. Since the blood flow is from the vascular system, the pressure of the fluid in the vascular porosity is of the order of the physiological blood pressure.

Lacuno-canalicular porosity consisting of the porous regions of the lacunae surrounding the center of the osteon and the interconnecting canaliculi. The stimuli-sensing osteocyte cells reside within the pores of the lacunae and their processes are housed in the canaliculi. Surrounding the cell processes in the canaliculi is a proteoglycan fiber matrix, which is thought to make the bone fluid act more like a gel. Zhang *et al* [57]

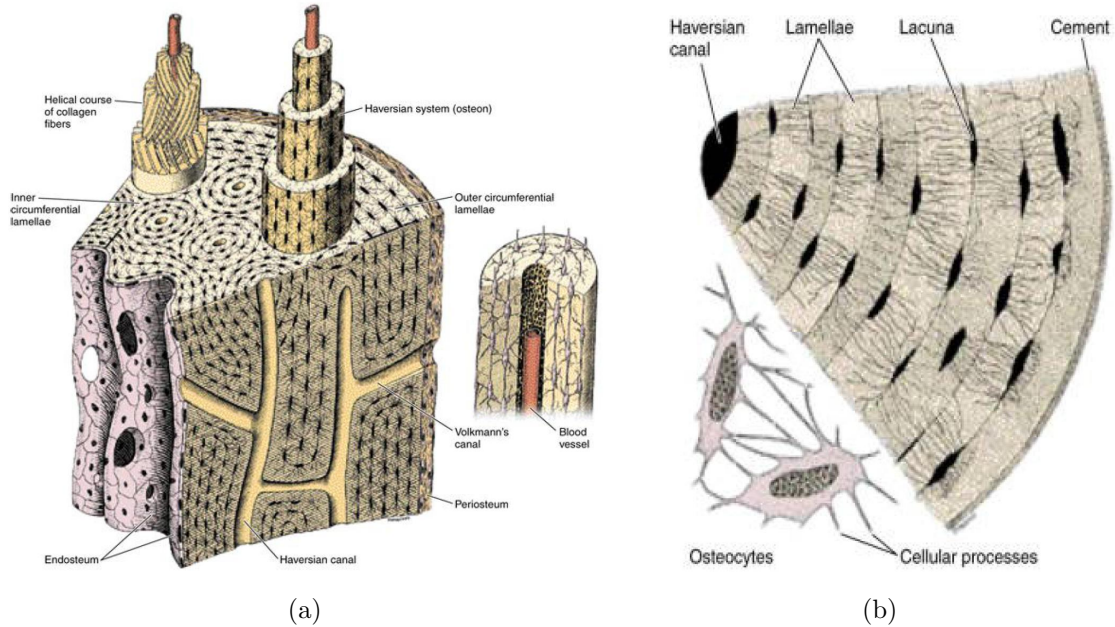


Figure 4.1: Schematic of the macro and microstructure of cortical bone. (a) Wall of the diaphysis of a long bone showing the vascular porosity (haversian system including the osteons, Haversian and Volkmann canals), (b) Part of the Haversian system showing the lacuno-canalicular porosity consisting of lacunae and canaliculi. Note the radially oriented canalicular channels connecting the lacunae. Figures taken from Junqueira and Carneiro [7].

estimated permeability at this scale of porosity to be $10^{-18} - 10^{-21} \text{ m}^2$.

Collagen-apatite porosity is seen at the nanostructural scale. The water present in this scale is considered to be part of the building block of the bone in addition to the collagen and apatite. The flow of water through the pores between the organic collagen and the apatite mineral can be neglected [10].

4.1.2 Role of fluid flow in mechanotransduction

It has been shown in a number of experimental and analytical works that the application of mechanical loading on the cortical bone results in the flow of fluid in and out of the different aforementioned porosities [58]. Weinbaum and co-workers showed the importance of lacuno-canalicular porosity for mechanotransduction in bone, and they hypothesized that osteocytes are deformed through the viscous shear stress due to the fluid flow inside the

canaliculi [42; 43; 59]. These studies motivate the use of a poroelastic material model for the cortical bone with lacuno-canalicular porosity.

Weinbaum and co-workers also developed an analytical model that describes the hierarchical nature of the lacuno-canalicular porosity in the cortical bone, using the results from Zhang and Cowin [60] at the macroscale, coupled to mechanical model at the cellular length-scale [43; 59; 60]. These works demonstrated that the fluid shear stress and drag force on the osteocyte process tethering fibers can amplify by more than 50 times the applied strains in the macroscale [43]. They also showed that the strain amplification factor varies non-linearly with respect to the loading frequency, in a similar way as the adaptation response observed in the experiments mentioned earlier. Fritton and Weinbaum's [61] extensive review provides a more complete description of flow-induced mechanotransduction in cortical bone.

In this chapter, we extend our numerical model described in Chapter 3 to incorporate poroelastic analysis. Previous works have included poroelasticity in the adaptation equation, using the fluid shear stress and octahedral strain to compute the tissue phenotype for fracture [62; 63]. We propose a new mechanical stimulus based on the viscous dissipation energy due to the fluid flow as the stimulus for adaptation. We will show that this stimulus can simulate the dependence of adaptation on loading frequency similar to the adaptation response observed experimentally. We consider an idealized bone geometry in the shape of a rectangular beam (following Zhang and Cowin [60]). This allows us to focus on the frequency trends by conducting a parametric study with relatively small computational resources.

This chapter is organized as follows: We begin with a description of the theory of poroelasticity as it applies to cortical bone, and identify a problem with an analytical solution to test our formulation. We then develop a measure of the mechanical stimulus to be used in an adaptation law. The results of simulations performed for a range of frequencies are presented, and these results are then interpreted in the context of mechanotransduction.

4.2 Theory of poroelasticity

We provide here a short overview of the theory of poroelasticity to establish the context for the poroelastic analyses presented in this chapter, and refer the interested reader to more comprehensive treatments in the literature, cf. [64–67]. A poroelastic medium is made up of a solid matrix and pores. We consider saturated media, in which all of the pores are filled with fluid. Let us consider a poroelastic representative volume element (RVE) of volume V . It can be written as the sum of the volumes of the solid matrix (V_s) and of the pores (V_p). Now, we can define porosity of the poroelastic medium n_p , and void ratio e as,

$$n_p = \frac{V_p}{V}; e = \frac{V_p}{V_s} = \frac{n_p}{1 - n_p}. \quad (4.1)$$

There are four field variables: stress tensor $\boldsymbol{\sigma}$, strain tensor $\boldsymbol{\epsilon}$, pore pressure p , and the variation in fluid content ζ . The variation in fluid content is a non-dimensional quantity defined as the variation of fluid volume per unit volume of the porous material due to fluid mass transport [68]. The medium is characterized by its porosity n_p , the bulk modulus of the solid K_s , shear modulus of the solid G , Poisson's ratio for the solid ν , and the bulk modulus of the fluid K_f . Biot viewed these field variables as conjugate pairs of stress quantities (σ_{ij}, p) , and strain quantities (ϵ_{ij}, ζ) . A fundamental idea put forth by Biot [64] is that the work done on a poroelastic medium is an exact differential given as, $dW = \sigma_{ij}d\epsilon_{ij} + pd\zeta = \epsilon_{ij}d\sigma_{ij} + \zeta dp$. Using this statement and concepts of thermodynamics, the constitutive equations for a general non-linear elastic poroelastic material can be derived (for more details, please consult Coussy [66]). Specializing the general constitutive equations for an isotropic, homogeneous, linear poroelastic material gives

$$2G\epsilon_{ij} = \sigma_{ij} - \left(\frac{\nu}{\nu + 1}\right)\sigma_{kk}\delta_{ij} + \alpha\left(\frac{1 - 2\nu}{1 + \nu}\right)p\delta_{ij}, \quad (4.2)$$

$$2G\zeta = \alpha\left(\frac{1 - 2\nu}{1 + \nu}\right)\left(\sigma_{kk} + \frac{3p}{B}\right). \quad (4.3)$$

We have introduced two additional parameters, the Willis coefficient α , and Skempton's coefficient B . The Willis coefficient α can be thought of as the ratio of fluid volume gained (or lost) in a poroelastic element due to volume change when loaded under drained condition ($p = 0$). It can be shown that $\alpha = 1 - K/K_s$, where K is the drained bulk modulus [68]. Skempton's coefficient B is obtained from the undrained condition ($\zeta = 0$) as $p = -B\sigma_{kk}/3$. Skempton's coefficient is the measure of the relative compressibilities of the solid and fluid phases, and is given as,

$$B = \frac{\alpha K_f}{[\alpha - n_p(1 - \alpha)]K_f + n_p K}. \quad (4.4)$$

In addition to these two parameters, for an ideal poroelastic medium, the undrained elastic properties (K_u and ν_u) may be expressed in terms of the four basic material parameters K, ν, B and α , as follows:

$$K_u = K \left[1 + \frac{\alpha^2 K_f}{(1 - \alpha)(\alpha - n_p)K_f + n_p K} \right]; \nu_u = \frac{3\nu + B(1 - 2\nu)(1 - K/K_s)}{3 - B(1 - 2\nu)(1 - K/K_s)}. \quad (4.5)$$

The fluid flow rate is computed from the pore pressure using Darcy's law,

$$q_i = -\kappa \frac{\partial p}{\partial x_i}, \quad (4.6)$$

where q_i is the fluid mass flow rate along the coordinate direction i and κ is the hydraulic permeability ($\kappa = k/\mu$, where k is the isotropic intrinsic permeability with units of m^2 and μ is the dynamic viscosity of the fluid). The fluid mass flow rate q_i is related to the volume averaged fluid flow velocity v_i^{fl} as $q_i = \rho_{fl} n_p v_i^{fl}$, where ρ_{fl} is the density of the fluid.

Inserting the constitutive equations and Darcy's law into the mass and momentum balance equations leads to the following equation for the pore pressure:

$$c \nabla^2 \left(\sigma_{kk} + \frac{3}{B} p \right) = \frac{\partial}{\partial t} \left(\sigma_{kk} + \frac{3}{B} p \right), \quad (4.7)$$

where c is the hydraulic diffusivity, defined as

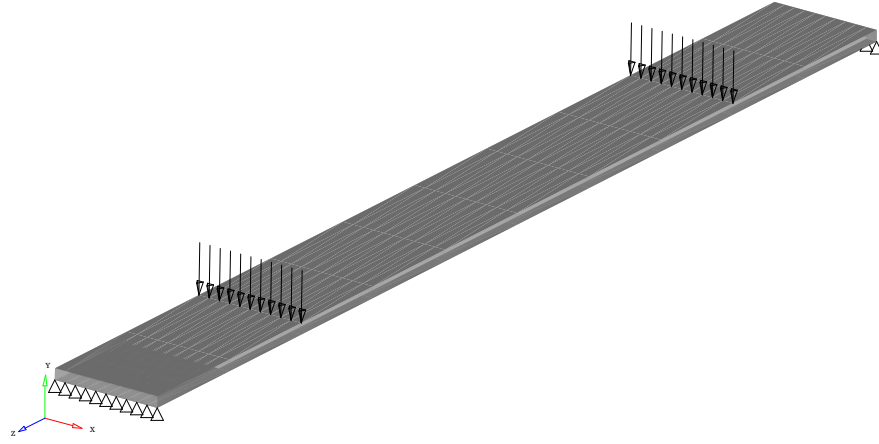
$$c = \frac{2GB^2\kappa(1-\nu)(1+\nu_u)^2}{9(1-\nu_u)(\nu_u-\nu)}. \quad (4.8)$$

Scaling the length in all directions by a characteristic length d ($\mathbf{X}^* = \mathbf{x}/d$), and scaling time using the frequency ω ($T^* = \omega t$), Eq. (4.7) becomes

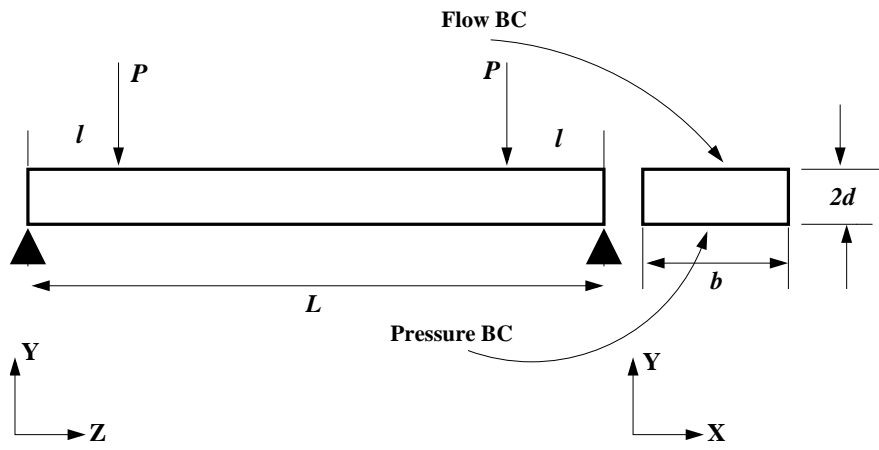
$$c\nabla^{*2} \left(\sigma_{kk} + \frac{3}{B}p \right) = Fo \frac{\partial}{\partial T^*} \left(\sigma_{kk} + \frac{3}{B}p \right), \quad (4.9)$$

where $Fo = \omega d^2/c$ is the Fourier number, which represents the ratio of the timescale for hydraulic diffusion (d^2/c) to the timescale of the applied load ($1/\omega$). For small Fourier number ($Fo \ll 1$), the transient term on the right hand side can be neglected in comparison to the left hand side, and the pressure solution will be essentially quasi-static. For large values of Fourier number ($Fo \gg 1$), the Laplacian on the left hand side of the equation can be neglected and the pore pressure follows the stress solution. When Fo is of order one, the two sides balance. This observation will help us to explain the results of our simulations below.

The objective of this chapter is to establish a frequency-dependent mechanical stimulus. To this end, we use as a test problem the bending of a poroelastic rectangular beam, similar to that considered by Zhang and Cowin [60]. Since our eventual goal is to analyze geometrically accurate bone models, we solve this problem using the finite element (FE) method, as implemented in the commercial software ABAQUS [38]. We consider a beam of length L , width b , and thickness d , subjected to a cyclic 4-point bending load of the form $P = (P_0/2)(1 - \cos \omega t)$. The geometry and boundary conditions (BC) are illustrated in Figure 4.2. In order to simulate the BC's for 4-point beam bending, we set $u_y = 0$ at both ends of the beam, and constrain one point such that $u_x = 0$ and $u_z = 0$ to avoid rigid body motion. The bending loads are at a distance l from each end of the beam, as shown. The periosteal



(a) Isometric view



(b) Orthographic view with boundary conditions

Figure 4.2: Schematic of the four-point bending setup of the rectangular poroelastic beam (outer) surface of a long bone is highly impermeable and the endosteal (inner) surface is highly permeable [69]. To simulate these properties in our model, we apply a zero pressure BC on the bottom surface, and zero flow BC on the top surface. We note that the latter BC is actually implemented via Darcy's law, Eq. (4.6) as $\nabla p \cdot \mathbf{n} = 0$, where \mathbf{n} is the normal vector. This leads to small residual velocities at the surface when implemented in the finite element method. The magnitude of these velocities is mesh-dependent, as discussed further below.

4.3 Development of the stimulus for the growth law

The form of the growth law we use to model the bone adaptation (see Eq. (3.1)) is

$$\frac{db_i}{dT} = \begin{cases} A(\phi_i - \phi_{ref}) & \phi_i \geq \phi_{ref} \\ 0 & \phi_i < \phi_{ref} \end{cases} \quad (4.10)$$

where b_i is the displacement of the surface node i , ϕ_i is the stimulus for adaptation at node i , and T is the growth timescale. For the elastic material case, we used the local strain energy density as the adaptation stimulus. However, the strain energy density cannot simulate the effect of loading frequency. Other stimuli such as maximum or average stresses or strains, which are used in other models ([21; 23–30]) are independent of loading frequency. Our goal in this chapter is to develop a stimulus similar to the strain energy density that is dependent on frequency. We would like to have a stimulus that captures the fluid flow in the bone, and reduces it to a convenient scalar quantity. Based on the second law of thermodynamics, there are three dissipation potentials identified: intrinsic dissipation due to the elastic deformation (which is accounted for in the derivation of constitutive equations), extrinsic dissipation due to flow of the viscous fluid and heat. Since we assume isothermal process, we choose the dissipation due to viscous fluid flow [66] as a candidate for the mechanical stimulus for adaptation. The dissipation due to viscous fluid flow φ is defined as,

$$\varphi = -n_p \mathbf{v}^{fl} \cdot \nabla p = \frac{1}{2} (n_p \mathbf{v}^{fl}) \cdot \kappa^{-1} \cdot (n_p \mathbf{v}^{fl}), \quad (4.11)$$

where \mathbf{v}^{fl} is the fluid velocity vector, ∇p is the pressure gradient, κ is the hydraulic permeability tensor, and n_p is the porosity of the poroelastic medium. We choose this form because it has been shown in a number of experiments, and hypothesized in analytical models, that the shear stress exerted by the fluid flow on the osteocyte cells is a possible candidate for mechanotransduction. Note that we model the rectangular beam as an isotropically per-

meable material for simplicity, even though the canalicular channels in the cortical bone are oriented preferentially in the radial direction. Anisotropic permeability is introduced in Chapter 5.

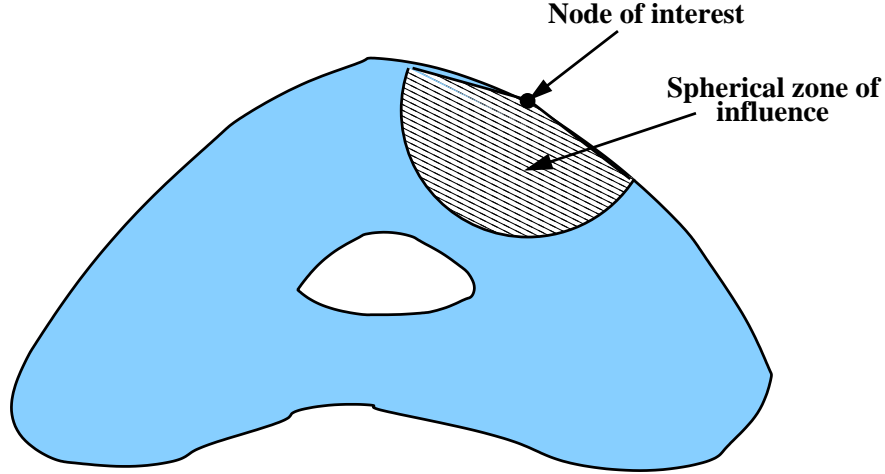


Figure 4.3: Schematic of the spherical zone of influence on a typical bone cross section

Osteocytes are interconnected through the processes inside the canalicular space. It can be hypothesized that they can communicate with each other through these processes, and exchange information on the current state of the stimulus at each location. To simulate this non-local behavior, we propose the use of a spatially averaged stimulus over a “zone of influence” (ZOI), shown schematically in Figure 4.3. For the sake of simplicity, we choose a spherical zone of influence of radius r . The stimulus that triggers the growth response at a surface node i is then defined as follows:

$$\phi_i = \frac{\int_{V_z} \left(\int_0^t \frac{1}{2} n_p \mathbf{v}^{fl} \cdot \kappa^{-1} \cdot n_p \mathbf{v}^{fl} dt \right) f(\|x\|) dV}{\int_V f(\|x\|) dV}, \quad (4.12)$$

where $f(\|x\|)$ is a function that weights the dissipation potential at an inner node by its distance $\|x\|$ from the surface node i , and V_z is the volume of the zone of influence. It seems logical for the nodes closest to the surface node to have the largest influence on growth. After trying several functional forms for f , we chose one that decreases exponentially with

distance, $f(\|x\|) = \exp(-5\|x\|/d)$, where d is the thickness of the beam.

4.3.1 Simulation details

The various material parameters used in the rectangular beam simulations, tabulated in Table 4.1, were chosen to be similar to properties of rat bone. The permeability of the lacunocanalicular porosity is reported to be in the range of 10^{-18} to 10^{-21} m² [42; 54]. Permeability will be shown to be a very important parameter in this problem, which directly affects the timescale of fluid diffusion in the poroelastic medium. The value of permeability we chose results in a significant range of Fourier numbers (from 12 to 380) for the physiological range of frequencies (1 Hz to 30 Hz). The hydraulic diffusivity corresponding to the permeability given in Table 4.1 is $c = 0.5606$ mm²/s. One can then estimate the characteristic time for hydraulic diffusion in a 1 mm thick beam to be of the order of one second.

| Property | Value | Units |
|-------------------------------|---------------------|----------------|
| Young's modulus of bone E | 12.0 | GPa |
| Fluid bulk modulus K_f | 2.3 | GPa |
| Solid bulk modulus K_s | 17.0 | GPa |
| Porosity n_p | 0.05 | - |
| Drained Poisson's ratio ν | 0.3 | - |
| Intrinsic permeability k | 3×10^{-20} | m ² |

Table 4.1: Values of different poroelastic parameters used in the analysis

The FE mesh for our analyses, shown in Figure 4.4, consists of 18,821 nodes and 4000 tri-quadratic hexahedral elements. The elements are graded along the y -axis in order to accurately enforce the zero-flow boundary condition. A grid convergence study showed that this grid resulted in velocities of the order 10^{-4} mm/s on the top surface, which when compared to velocities in the bulk ($O(1)$ mm/s) were judged to be sufficiently small. The poroelastic solver in ABAQUS [38] uses an unconditionally stable backward Euler integration scheme to solve Eq. (4.7). The accuracy of the solution still depends on the size of the timestep. We performed a convergence study to establish the timestep size as well, where

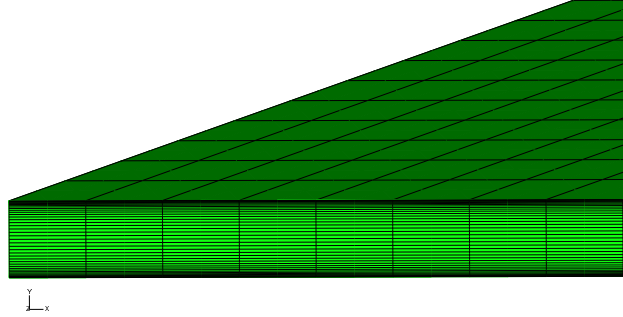


Figure 4.4: Finite element mesh of the rectangular beam. Note the biased mesh across the cross-section

we found that accurate results were obtained for $\Delta t < 0.5h^2/c$, where h is the characteristic length of the smallest element in the flow direction. For the grid and properties used in the simulations reported below, this criterion gives $\Delta t = 5 \times 10^{-4}$ s.

4.4 Results

Figures 4.5(a)-4.5(f) show the time evolution of pore pressure and the fluid velocity at different points in the rectangular cross section for several different frequencies. In each case, there is an initial transient in both the pore pressure and fluid velocity that dies out after about 0.5 sec, which is of the order of the timescale of hydraulic diffusion noted earlier. After this time, both quantities repeat with the same frequency as the applied load. This is important, because it allows us to extrapolate the pore pressure and velocity solutions at the end of the initial transient to the entire duration of the experiments, which may be several minutes [4], thus reducing the computational cost significantly. The velocities are maximal at the bottom surface, falling to nearly zero at the top. Note that both the pore pressure at the upper surface, and the velocity at the lower surface, are out of phase with their respective values at the neutral axis.

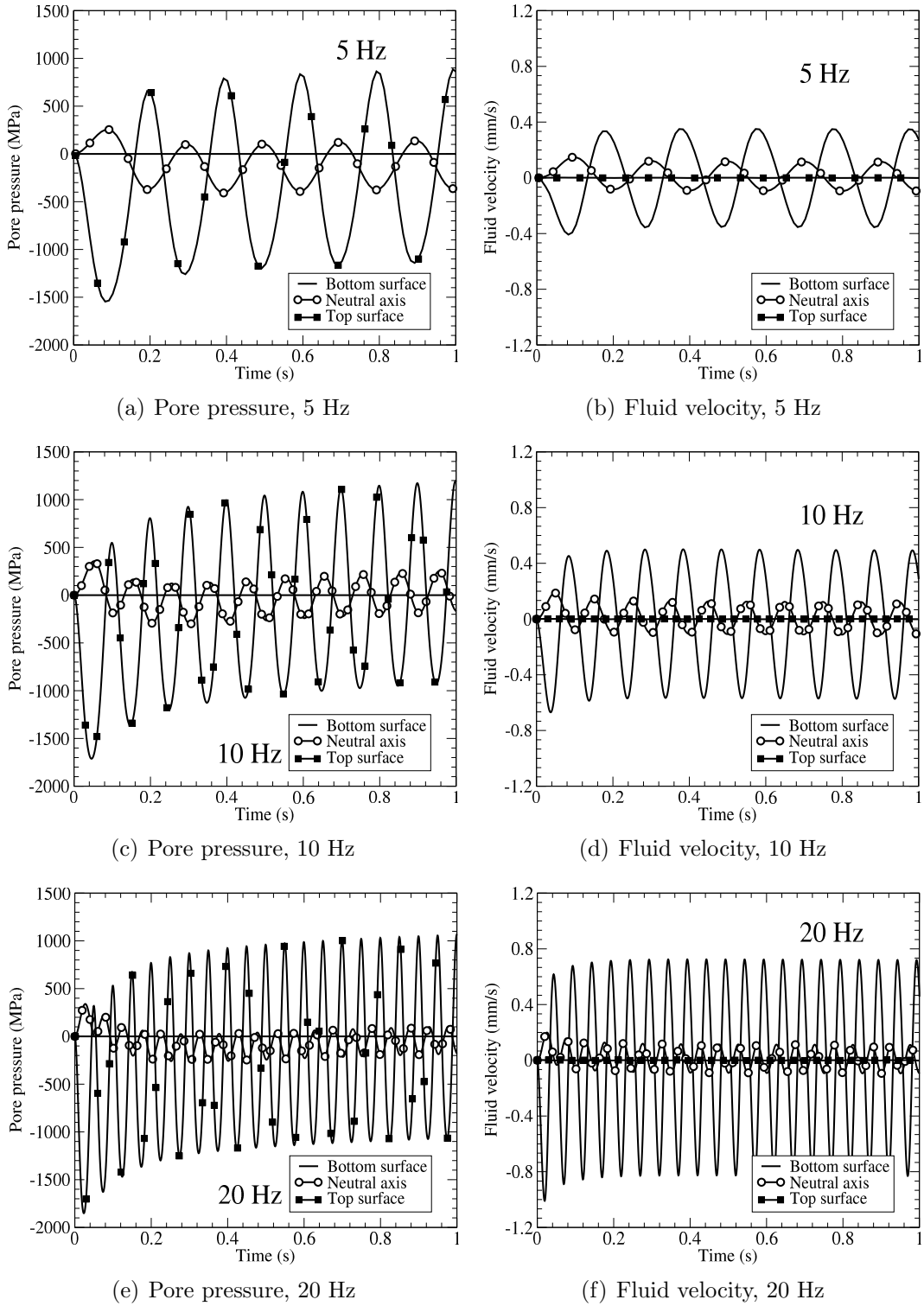


Figure 4.5: Plots of pore pressure (a, c, and e) and fluid flow velocity (b, d, f) solutions as a function of time for several frequencies.

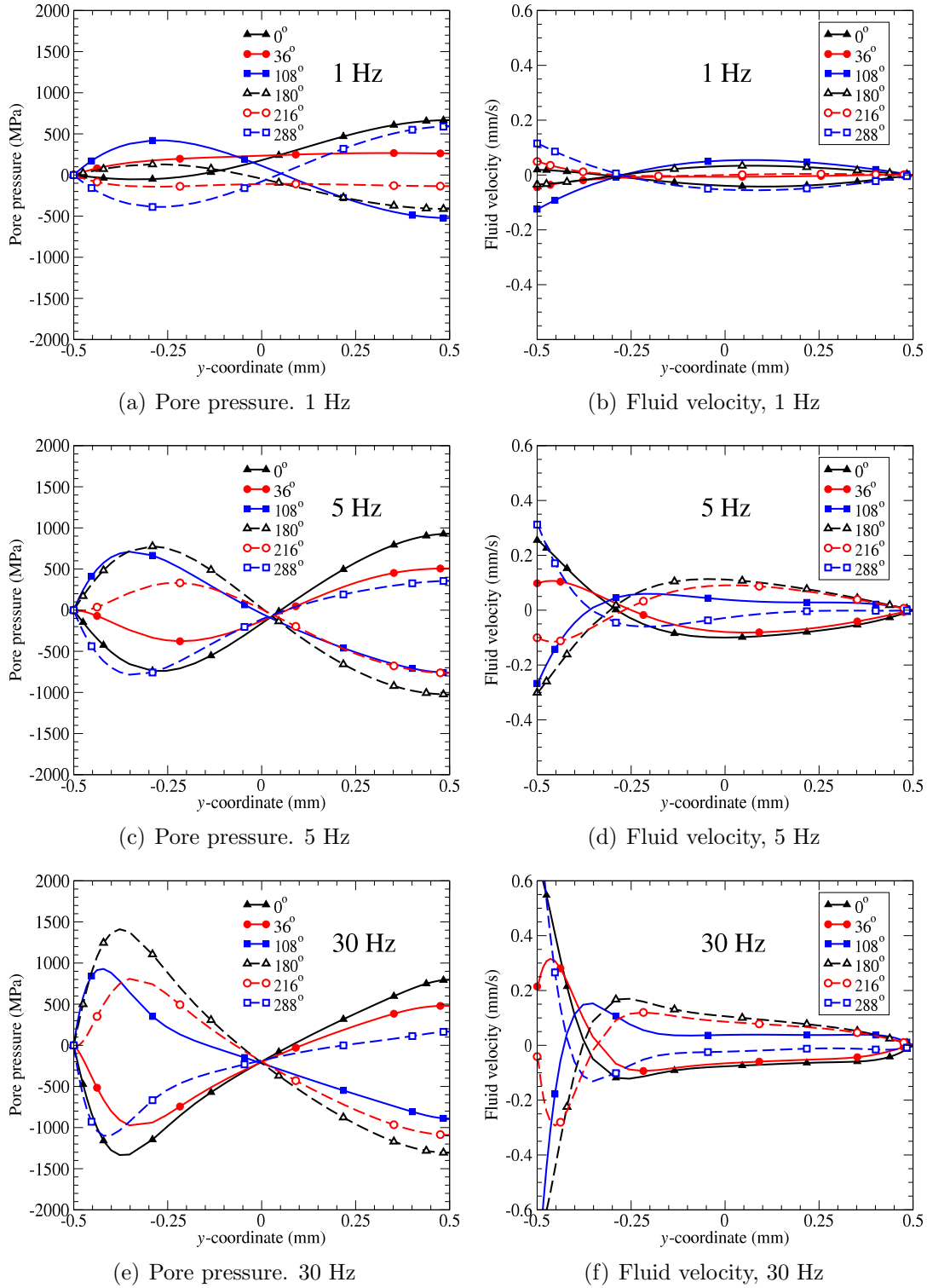


Figure 4.6: Plots of pore pressure (a, c, e) and fluid flow velocity (b, d, f) solutions across the cross-section at different points of time in a loading cycle shown here as the phase angle for 1 Hz, 5 Hz and 30 Hz.

Figures 4.6(a)-4.6(f) show the pore pressure and velocity profiles through the thickness for frequencies of 1 Hz, 5 Hz and 30 Hz, at different phase angles in a single load cycle after the initial transient. At 1 Hz (Figure 4.6(a)), the slope of the pore pressure distribution reaches zero midway between the bottom surface and the neutral axis, indicating zero flow velocity there. Figure 4.6(b) shows the corresponding fluid velocity, which shows that the velocity on the bottom surface is of higher magnitude and has a smooth variation throughout the profile. The zero velocity solution is observed at the same location throughout the loading period.

At 5 Hz, (Figures 4.6(c) and 4.6(d)), the pore pressure remains very close to zero at the neutral axis throughout the load cycle. The location of the zero-slope in the pressure solution (zero velocity by Darcy's law) varies with time of loading, and the velocity is nearly uniform over much of the beam cross-section above the neutral axis for parts of the load cycle. These effects are more pronounced at 30 Hz (Figures 4.6(e) and 4.6(f)). There is a very large variation in pore pressure near the bottom surface, which then falls to almost zero at the neutral axis. The velocity is also very large near the bottom surface, and then nearly uniform (and much smaller) over most of the remainder of the cross-section.

These results are summarized in Figures 4.7 and 4.8, which compare the pore pressure and fluid velocity through the thickness for different frequencies. As the frequency increases, the variation in pore pressure increases near the bottom surface, and becomes close to linear between $y = -0.3$ mm to $y = 0.3$ mm of the thickness. It should also be noted that the magnitude of pore pressure near the top surface (where the pressure gradient is zero) increases for lower frequencies (between 1-10 Hz), but starts decreasing for frequencies greater than 10 Hz. Figure 4.8 shows that for frequencies less than 10 Hz, the flow is significant to about 50% of the thickness, i.e., up to the neutral axis, whereas at higher frequencies, even though the velocity at the bottom surface increases, the flow velocity becomes very small before reaching 25% of the thickness.

The dissipation energy, given by the time integral of the dissipation potential in Eq. (4.11),

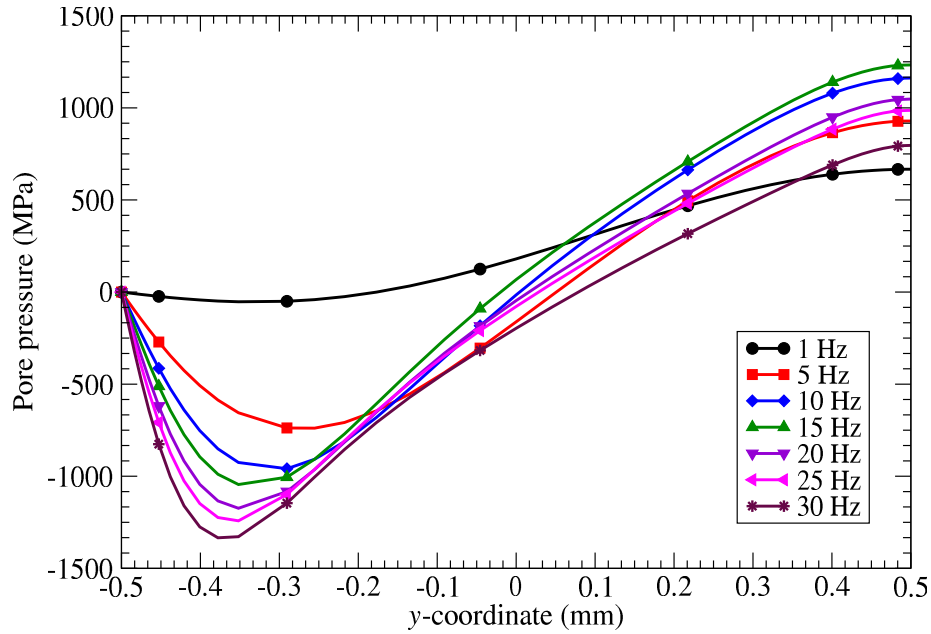


Figure 4.7: Plot of pore pressure solution across the cross-section for different frequencies at the end of a loading cycle.

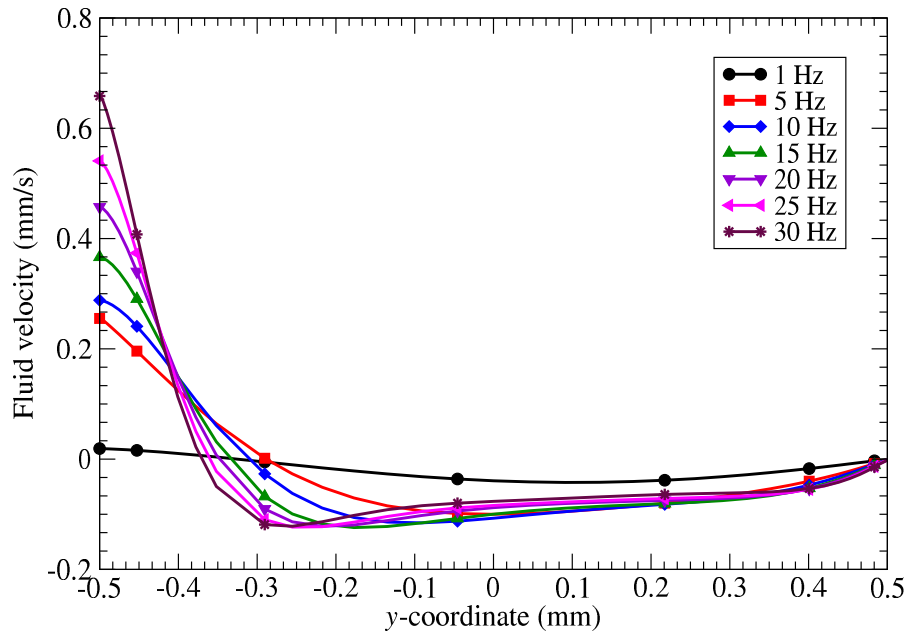


Figure 4.8: Plot of the fluid velocity pressure solution across the cross-section for different frequencies at the end of a loading cycle.

was then computed from the solutions, extrapolated to 100 s using the results at the end of the initial transient. Figure 4.9 shows that the dissipation energy in the beam follows a similar trend as the fluid velocity. The dissipation energy at the bottom surface increases

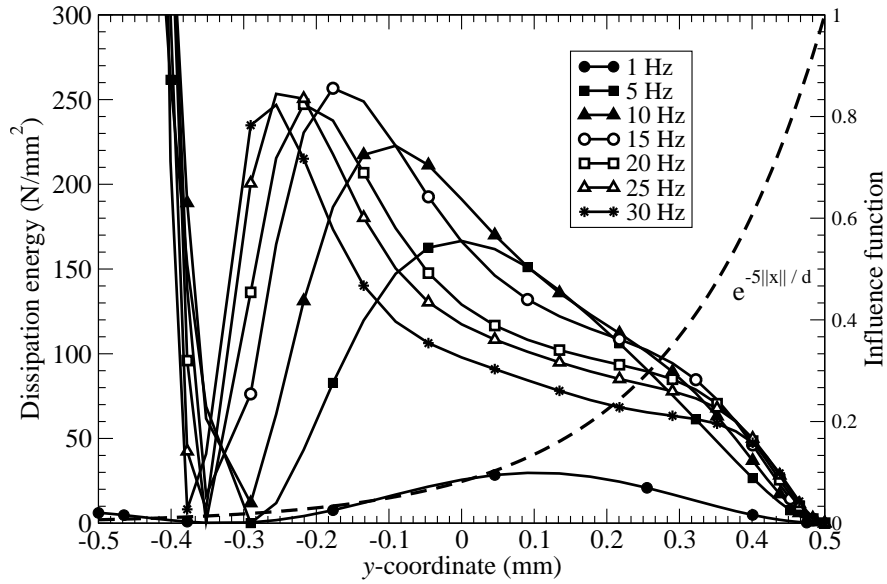


Figure 4.9: Plot of dissipation energy (time integral of dissipation potential) as a function of the y -coordinate for different frequencies. Also superimposed on the plot is the influence functions. We chose $\exp(-5||x||/d)$ as the influence function for our simulations. We have zoomed in on the plot to show the variation in the dissipation energy inside the beam, and so the very high values near $y = -0.5$ mm are cut off.

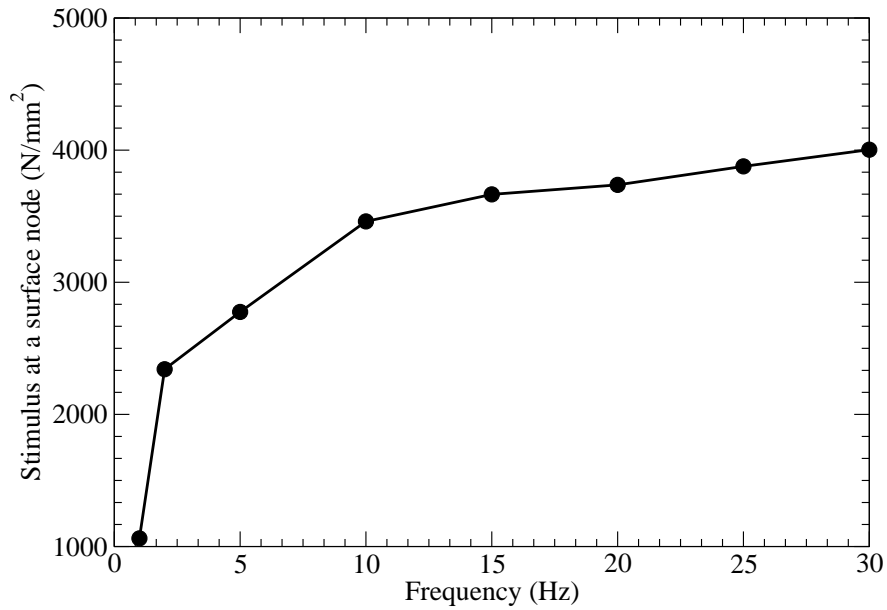


Figure 4.10: Plot of dissipation energy stimulus at a surface node as a function of loading frequency.

with frequency. Since the bottom surface is farthest from the top surface, where significant adaptation occurs, we hypothesize that the effect of the dissipation energy at the bottom

surface will be very small on the adaptation at the top. To implement this hypothesis in our analysis, we choose the exponential weighting function described earlier and illustrated in Figure 4.10.

Figure 4.10 shows the adaptation stimulus (computed from Eq. 4.12) at a top surface node at the midspan of the beam, as a function of the frequency of loading. We can see clearly that the adaptation stimulus increases rapidly with frequency at first, then saturates at about 10 Hz. We then used the growth law in Eq. (3.1) to simulate growth of the

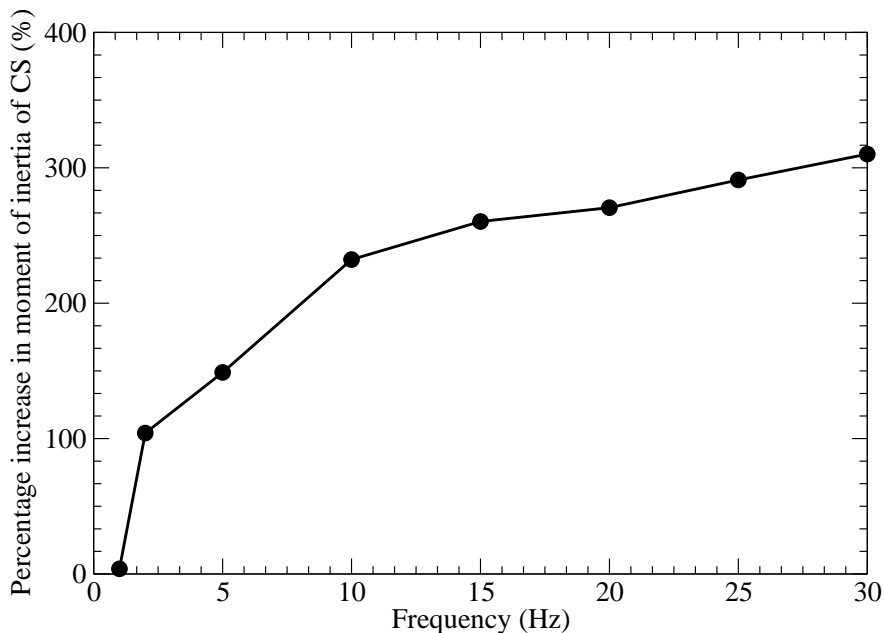


Figure 4.11: Plot showing the percentage increase in the moment of inertia of the rectangular cross section due to adaptation as a function of frequency.

rectangular beam for two growth-timesteps, using a sensitivity $\phi_{ref} = 1000 \text{ N/mm}^2$ and a gain $A = 0.01 \text{ mm}^3/\text{N}/\text{timestep}$. We express the results as the percentage change in the moment of inertia of the rectangular cross section due to the adaptation. Figure 4.11 shows that the growth depends on frequency in a manner similar to the stimulus. This behavior is typical of that observed in numerous experiments [3–5; 15], especially in the work of Warden and Turner [4].

4.5 Discussion

We have proposed the dissipation energy of the poroelastic flow induced by mechanical loading as the stimulus to trigger adaptation in cortical bone. We implemented this model using a simple rectangular beam because: (a) the geometry is simple, so the pressure and velocity solutions can be understood in greater detail; (b) and the problem has an analytical solution, which makes it easier to understand the role of the various physical parameters.

We found that at low frequencies, the flow first increases with loading frequency, but that as the frequency increases beyond 10 Hz, the amount of fluid that penetrates through the thickness of the beam decreases. This means that at the higher frequencies, there is less flow seen by the osteocytes in the lacunae, and hence they experience smaller shear and drag forces. The dissipation energy represents the work done by these forces, and thus at higher frequencies, the dissipation energy stimulus saturates, and hence one can expect the adaptation to saturate as well. Weinbaum and co-workers [43; 59; 60] analyzed the shear and drag forces on the osteocyte cell membrane due to the poroelastic flow at the cellular level, and showed that the amplification of the strains on the osteocyte cell membrane has a similar frequency response to our results for dissipation energy stimulus and increase in moment of inertia (see Figures 4.10 and 4.11). The proposed dissipation energy stimulus is thus qualitatively consistent with the strain amplification hypothesis and experimental observations.

We have also proposed the use of a “zone of influence” in order to simulate the effect of the distributed network of osteocytes and their communication. Osteocytes have been identified as the mechanosensory cells in the cortical bone. It has not yet been proven experimentally whether the osteocytes communicate with each other, and if so, what their mode of communication is. A series of experiments by Turner and coworkers on the expression of the sclerostin protein in response to *in vivo* mechanical loading suggest that there is such osteocyte communication, and also suggest the existence of a zone of influence. Robling

et al [55] found that the expression of sclerostin protein, which is found almost exclusively in the osteocyte cells, is reduced considerably by *in vivo* mechanical loading, and further that the reduction in the sclerostin expression is much greater in the regions experiencing higher strains. The most pertinent observation from their experiments for our work is that the expression of sclerostin is reduced on a group of osteocyte cells near the regions where higher bone growth is observed. This is a possible indication of the existence of a zone of influence within a real bone. Further experimental investigation of the expression of sclerostin at the scale of the osteocyte network could be used to inform the model as to the shape of the zone of influence.

4.6 Summary

We have performed poroelastic analyses on a simplified geometric model using the commercial FE software ABAQUS. We investigated the variations in pore pressure and fluid velocity with time and location in the beam, and their dependence on frequency. Based on these results, we propose the use of the dissipation energy as a mechanical stimulus for adaptation that can accommodate the effect of frequency. We also included the effect of non-locality of mechanotransduction of osteocytes present in the lacunae in the cortical bone through the use of a zone of influence. The dissipation energy stimulus evaluated in this manner is shown to increase linearly with frequency in the low frequency range, and saturate at the higher frequency range. The implementation of the poroelastic material model and the dissipation energy stimulus can be integrated seamlessly into our framework to simulate adaptation response on cortical bone. In the next chapter, we extend the poroelastic material model and the dissipation energy stimulus to the actual rat ulna FE model, and to quantitatively validate the numerical model by comparing with experimental observations.

Chapter 5

Poroelasticity – Rat ulna model

5.1 Introduction

We proposed a new dissipation energy based stimulus for adaptation in Chapter 4, and implemented it for a rectangular beam model. In this chapter, we apply this stimulus on the rat ulna FE model that we developed from μ -CT images in Chapter 2. We use the experimental results of Robling *et al* [1] corresponding to the 9 N load at 2Hz applied for 360 cycles as a single bout to calibrate the growth law parameters. We use those parameters to investigate the effect of frequency on the rat ulna model. We compare the results of the adaptation simulations qualitatively to the results of Warden and Turner [4].

5.2 Methods

We briefly review here the details of the rat ulna FE mesh from Chapter 2. The rat ulna mesh consists of 37,949 parabolic tetrahedral elements and 62,782 nodes. A 9 N compressive load is applied at 2 Hz, the same as from Chapter 3, with a cosine based time variation as $P = (P_0/2)(1 - \cos \omega t)$. The boundary conditions for fluid flow are anatomically based (similar to Chapter 4) so that the endosteal (inner) surface is fully permeable and periosteal (outer) surface is fully impermeable.

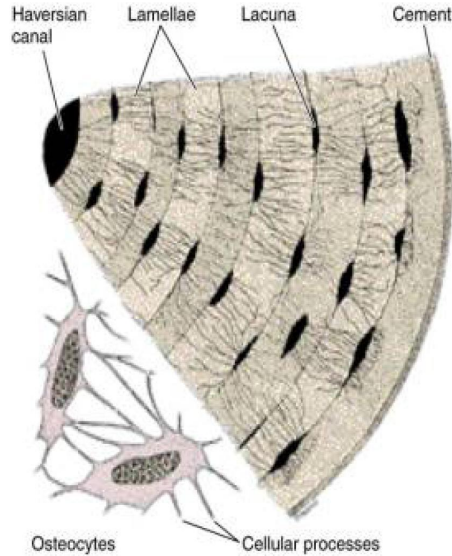


Figure 5.1: Schematic of part of the Haversian system showing the lacuno-canalicular porosity consisting of lacunae and canaliculi. Note the radially oriented canalicular channels connecting the lacunae [7].

5.2.1 Orthotropic permeability

In Chapter 4, we assumed that the permeability was isotropic. In the rectangular beam, the fluid velocity solution depends only on the y -coordinate, but not on x and z . Thus anisotropy in the permeability did not come into play. The geometry of rat ulna will lead to both radial and azimuthal flow under bending. Canalicular channels are oriented along the radial direction (see Figure 5.1), so it is reasonable to assume that the flow along the radial direction is dominant compared to the flow in the circumferential and longitudinal directions. We, therefore, model the rat ulna as a transversely isotropic medium for permeability.

In order to implement the transversely isotropic permeability in ABAQUS, we define a local coordinate system to determine the material orientation in each element [38]. To compute the centerline of the rat ulna model, we generate images of cross-sections of the rat ulna, similar to the moment of inertia calculation described in Chapter 2. We then compute either the centroid of the area (when the section is solid) or the centroid of the inner medullary canal (when the section is hollow). The locus of these centroids forms

the centerline of the bone model, and is stored as a data file. The radial direction in each element is calculated as the vector from the centerline at the same longitudinal location to the centroid of the element. The axial direction is defined as the vector along the centerline. The azimuthal direction is then computed as the cross product of the axial and radial vectors. The subroutine used to perform these calculations can be found in Appendix C. A permeability tensor κ (of units m^2) with $\kappa_{rr} = 3 \times 10^{-20}$, $\kappa_{\theta\theta} = 3 \times 10^{-22}$, $\kappa_{zz} = 3 \times 10^{-22}$, $\kappa_{ij} = 0, i \neq j$ is used in the poroelastic analysis.

5.2.2 Zone of influence

The form of dissipation energy based stimulus from Chapter 4 is used, which is of the following form:

$$\phi_i = \frac{\int_V \left(\int_0^t \frac{1}{2} n_p \mathbf{v}^{fl} \cdot \kappa^{-1} \cdot n_p \mathbf{v}^{fl} dt \right) f(\|x\|/R) dV}{\int_V f(\|x\|/R) dV}. \quad (5.1)$$

We again use the spherical zone of influence (ZOI) with the influence function $f(\|x\|/R)$, where R is the radius of ZOI. The ulna mesh is not as fine as the rectangular beam mesh in Chapter 4. Due to the coarseness of the mesh, the fluid velocity solution has some error on the outer surface. Because of this computational error, we change the exponential influence function used in Chapter 4 to $f(\|x\|/R) = (\|x\|/R)^3 \exp(-11\|x\|/R)$, in order to make the contribution of the surface nodes close to zero.

5.2.3 Simulation details

The various material parameters used for the rat ulna simulations are tabulated in Table 5.1, which are similar to properties used for the rectangular beam. The hydraulic diffusivity corresponding to the permeability given in Table 5.1 is $c = 0.7801 \text{ mm}^2/\text{s}$. The characteristic time of diffusion for the rat ulna can then be estimated to be of the order of a second corresponding to the maximum thickness of rat ulna (which is about 0.8mm).

| Property | Value | Units |
|-------------------------------|---------------------|--------------|
| Young's modulus of bone E | 17.0 | GPa |
| Fluid bulk modulus K_f | 2.3 | GPa |
| Solid bulk modulus K_s | 23.0 | GPa |
| Porosity n_p | 0.05 | - |
| Drained Poisson's ratio ν | 0.3 | - |
| Intrinsic permeability k | 3×10^{-20} | m^2 |

Table 5.1: Values of different poroelastic parameters used for the rat ulna in the analysis

We used the poroelastic solver in ABAQUS [38], which employs an unconditionally stable backward Euler integration scheme to solve the poroelastic governing equations, similar to the rectangular beam problem. The accuracy of the solution still depends on the size of the timestep. We performed a convergence study to establish the timestep size as well, and we found that accurate results were obtained for $\Delta t < 0.5h^2/c$, where h is the characteristic length of the smallest element. For the grid and properties used in the simulations reported below, this criterion gives $\Delta t = 2 \times 10^{-3}$ s. So we use timesteps equal to or less than 2×10^{-3} s in all the simulations described below.

5.3 Results

We performed a series of growth simulations on the rat ulna model using the poroelastic material model, and we present those results here. Firstly, we solved the poroelastic problem for the baseline case of 9 N compressive load at a frequency of 2 Hz. This frequency corresponds to a Fourier number Fo of 9.5. All the poroelastic analyses described below were performed on the supercomputing cluster Abe on the National Center for Supercomputing Applications (NCSA) at UIUC. Each poroelastic iteration of the the growth simulation typically takes about 8 hours on 8 processors on the Abe cluster.

Figure 5.2(a) shows the velocity fields at the midshaft cross-section of the ulna at a point in the loading cycle. One can see that the fluid flows in from the top of the cross-section to the center, and then flows out from the center to the bottom showing that fluid flows out

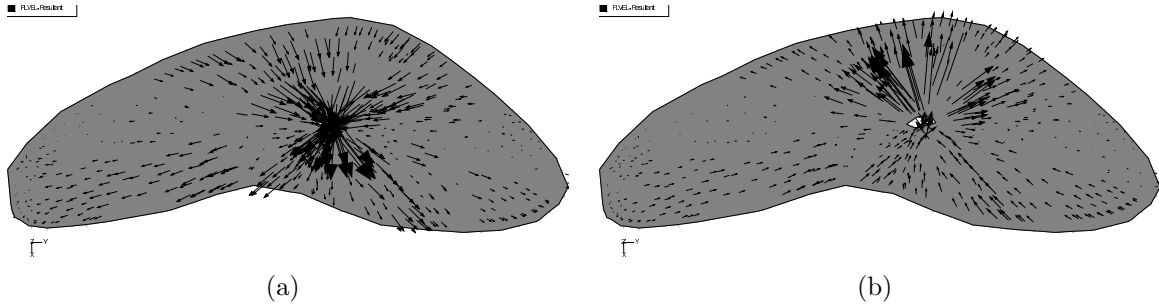


Figure 5.2: Fluid velocity fields for frequency of 2 Hz corresponding to (a) loading cycle, and (b) unloading cycle.

in compressive regions and flows in to tensile regions. The flow is primarily radial because of the dominance of the radial permeability in the orthotropic permeability tensor. Flow moves in and out of the medullary canal through the endosteal surface because of the zero pressure boundary condition imposed there. Very little flow can be seen at the periosteal surface because of the applied zero flow boundary condition. We can also see very little flow near the neutral axis due to small pressure gradients in that region. Figure 5.2(b) shows the velocity field at a point in the unloading cycle. Fluid flow has changed directions in the unloading cycle since the compressive and tensile regions are interchanged.

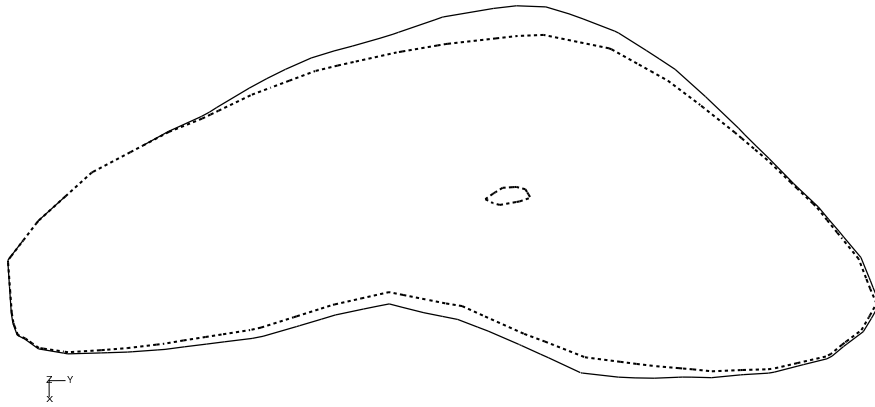


Figure 5.3: Comparison of the cross-sections between the original and grown rat ulna at the midshaft from our simulations using the dissipation energy stimulus. Compare this to the results from the experiments of Robling *et al* [1] in Chapter 3.

Using the results from the poroelastic analysis, we performed growth simulations for the 2 Hz case. We used the following baseline values for the growth law parameters: gain $A = 4$

$\text{m}^3/\text{N}/\text{timestep}$ and threshold-sensitivity $\phi_{ref} = 3 \times 10^{-2} \text{ J}/\text{m}^3$. These parameter values were selected based solely on computational experiments, in order to produce significant growth within four growth-timesteps. Figure 5.3 shows the comparison of the ulna midshaft section before and after growth from the poroelastic growth simulation. We can see that the simulation can predicts reasonably well the shape of the cross-section when compared to the experimental results (refer Figure 3.3(b) in Chapter 4).

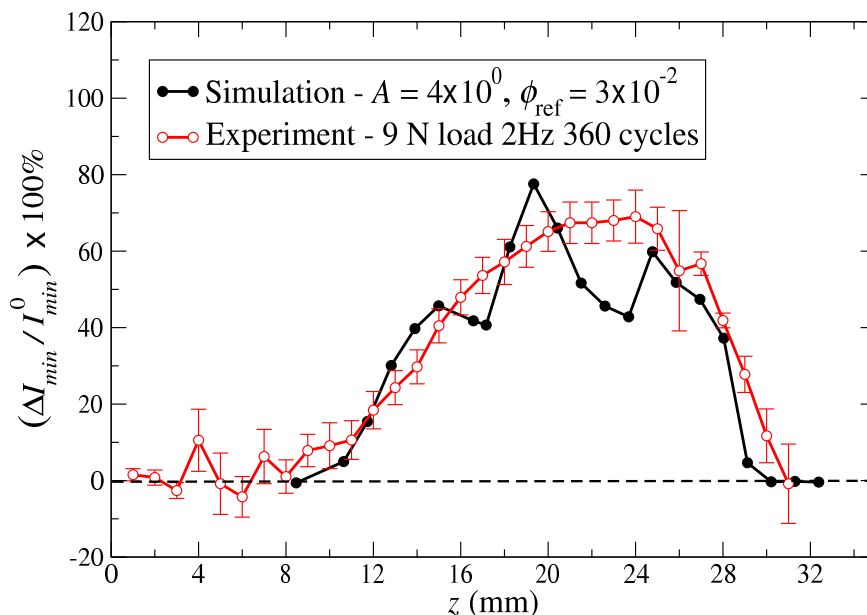


Figure 5.4: Plot showing the increase in the I_{min} due to adaptation from the numerical model compared to the experimental results of Robling *et al* [1].

For the chosen growth parameter values, we computed the percentage increase in the minimum moment of inertia I_{min} due to the growth along the length of the ulna. These results are compared with the experimental results in the plot in Figure 5.3. We can see that the simulation results match the results from the experiment closely. There are marked differences in the plot compared to the results from an elastic model in Chapter 3, such as the existence of peaks and valleys. We will elaborate more on these discrepancies in the next section.

We used the established values for the gain and threshold-sensitivity parameters to investigate the effect of different frequencies of loading on the rat ulna model. Figure 5.5 shows

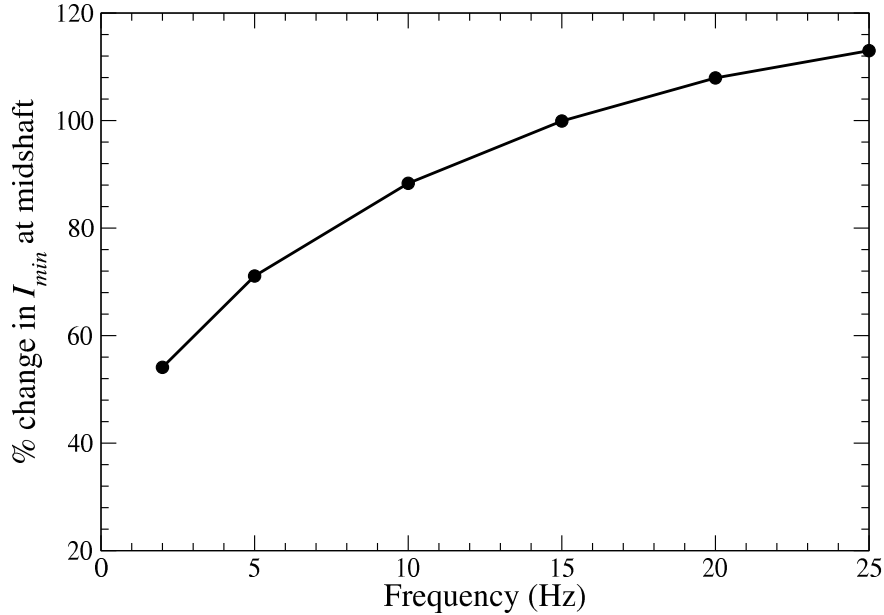


Figure 5.5: Plot showing the percentage increase in the moment of inertia of the rat ulna midshaft cross-section due to adaptation as a function of frequency.

the percentage increase in I_{min} at the midshaft cross-section for applied frequencies of 2, 5, 10, 15, 20, and 25 Hz. We see that the growth response is non-linear with respect to the loading frequency similar to a number of experimental results [3; 5; 15], specifically those of Warden and Turner [4].

5.4 Discussion

We extended the use of dissipation energy stimulus, developed in Chapter 4, to the rat ulna model to simulate cortical bone adaptation. We calibrated the growth law parameters using the experimental results through quantitative comparison of the bone growth. We also investigated the effect of varying frequencies on the rat ulna growth response.

We selected the values for the gain and threshold-sensitivity parameters from computational experiments. Note that the threshold-sensitivity value used for the poroelastic analysis is much smaller (by three orders of magnitude) than that is used in the elastic case. This is physically consistent since the magnitude of dissipation energy is usually much lower than

the magnitude of strain energy density.

There can be more than one combination of growth law parameters that can fit the experimental results. More experiments are needed to establish biophysically consistent values for the gain and threshold-sensitivity parameters. Experiments that concentrate on the mechanotransduction of the osteocyte cell body and establish the threshold fluid flow that trigger a mechanical response, can help us determine the value of ϕ_{ref} . Experiments that focus on computing accurately the bone apposition rates can aid in computing the value of the gain parameter A . We also note that, similar to the elastic material model, changing the value of ϕ_{ref} can simulate the effect of multiple bouts of loading on bone adaptation.

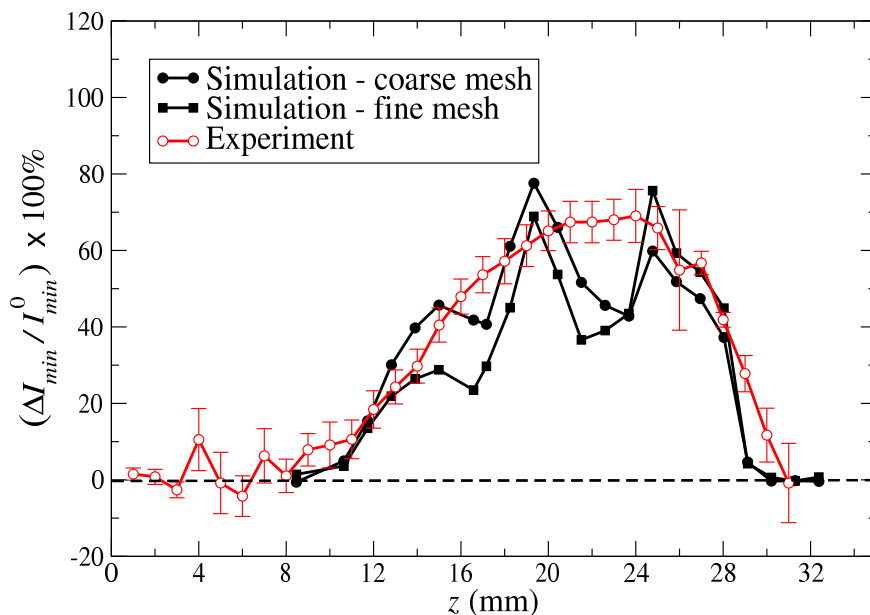


Figure 5.6: Plot showing the increase in the I_{min} due to adaptation from the numerical model for the coarse and fine mesh compared to the experimental results of Robling *et al* [1].

Comparing the increase in I_{min} from the poroelastic analysis to the elastic solution in Chapter 3, we can observe that the plot in Figure 5.3 is not as smooth, and with well-defined valleys and peaks. One of the reasons for these discrepancies maybe due to the fact that the FE mesh of the ulna is not fine enough and that can introduce some inaccuracies in the fluid velocity results. We performed a mesh convergence study by looking at the adaptation

response on a FE mesh of about 280,000 nodes and 180,000 parabolic tetrahedral elements. Since there are more elements across the ulna for this mesh, the applied zero flow boundary condition on the periosteal surface is better resolved and the velocity solutions are more accurate than those found for the coarser mesh.

However, the peaks and valleys in the adaptation solution remained as seen from Figure 5.6. This fine mesh is computationally very intensive so that it takes about 20 hours for each iteration in the growth simulation. Since not much improvement was gained with respect to the growth results, we use the coarse ulna mesh for all the analyses. Other possible reasons maybe are the size and shape of the zone of influence, and the influence function of the nodes inside the zone of influence. But it is not possible to determine the influence of these parameters without experimental investigation on the zone of influence.

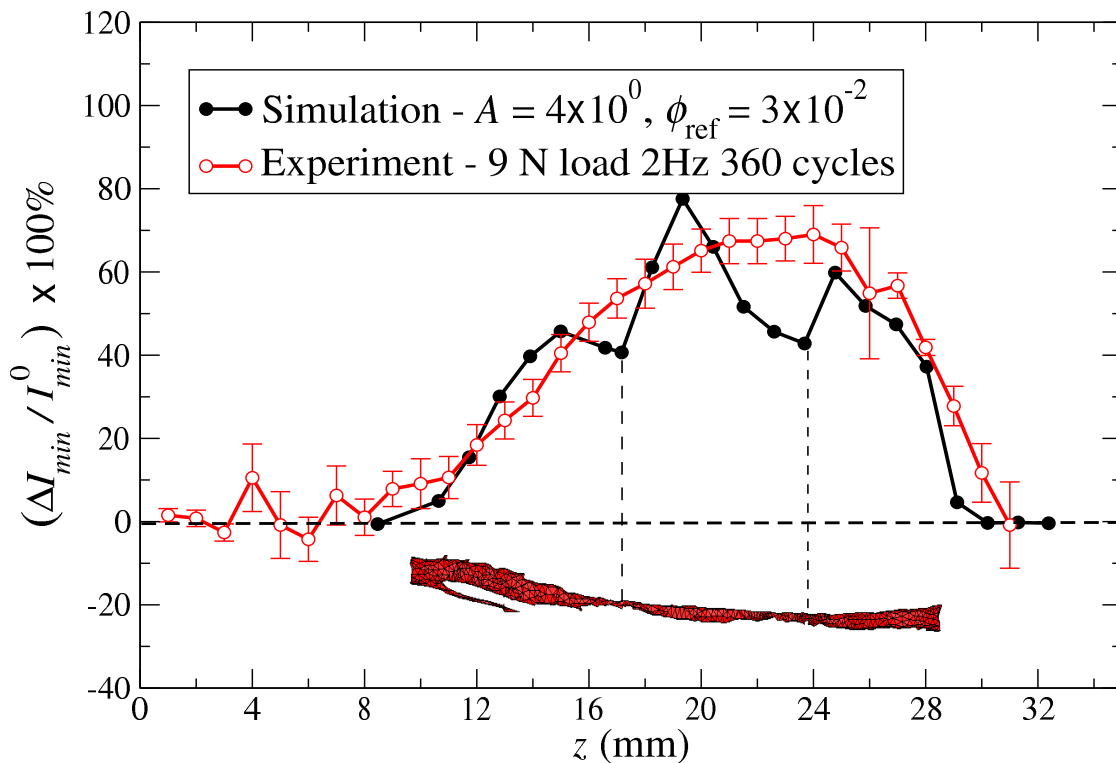


Figure 5.7: Plot showing the increase in the I_{min} due to adaptation from the numerical model with the FE mesh of the intermedullary canal to show the correlation between the size of the intermedullary canal and the valleys in the I_{min} plot.

We found that there is a correlation between the locations of the peaks and valleys

in the increase in I_{min} plot and the size of the intermedullary canal. Figure 5.7 shows the percentage increase in I_{min} plot superimposed with the FE mesh of the intermedullary canal. We can clearly see that the valleys in the I_{min} plot occur at places where the size of the intermedullary canal is reduced. This correlation implies that the increase in I_{min} computed from the adaptation simulation reflects the detail of the geometry of the particular rat ulna under investigation. The experimental results of Robling *et al* [1] is the average of growth occurred on 13 different rat samples, which could probably have smoothed out the peaks and valleys in the individual rats. This is an interesting observation since the adaptation simulation using an elastic material model did not reflect the effect of the geometry of intermedullary canal and thus resulted in a smooth growth response. More analyses will be carried out to identify the effect of the intermedullary canal on the adaptation results.

Finally, the concept of ZOI is new and there are no other experimental evidences of its existence. The experiments of Prof. Turner's group focusing on the expression of sclerostin in osteocytes when subjected to applied loading are a great initial step in establishing the existence of a ZOI [55]. More experiments are needed to decipher the role of the zone of influence on adaptation, especially the size, shape, and the functional form of the influence of the nodes inside the zone.

Chapter 6

Conclusions and Future Work

We have developed a numerical framework that can simulate and predict cortical bone adaptation under different loading conditions. We developed a method to generate finite element (FE) models of actual rat ulna from μ -CT images. The external adaptation process was implemented in the model by moving the surface nodes of the FE mesh based on an evolution law characterized by two parameters: gain A and threshold-sensitivity ϕ_{ref} .

Cortical bone was modeled firstly as an elastic material. Loading from experiments of Robling *et al* [1] was applied on the FE model, and the elastic boundary value problem was solved. The surface nodes were displaced along the normal direction according to the local strain energy density to simulate growth. We showed that the effect of splitting the loading cycles into different number of bouts on bone adaptation can be incorporated in the numerical model by changing the only the threshold-sensitivity but not the rate of adaptation. We also showed that the threshold-sensitivity parameter can quantify the mechanosensitivity of the osteocytes by comparing the numerical results to those from the experiments.

We modeled cortical bone as a poroelastic material to account for the interstitial fluid flow in order to incorporate frequency into the numerical framework. We developed a dissipation energy based growth stimulus for the poroelastic material model. We proposed and implemented the concept of a “zone of influence” near each surface point, in order to incorporate the non-locality in the mechanotransduction of osteocytes present in the lacunae. We showed that this stimulus can predict the non-linear dependence of bone adaptation on frequency of applied load using the FE models of both simple rectangular beam and the actual rat ulna. Similar non-linear adaptation frequency response also has been observed in

numerous experiments.

We have successfully implemented a numerical framework that can predict the effect of magnitude, frequency, and number of bouts of applied loading on cortical bone adaptation. The effectiveness of the model depends on the existence of good quantitative experimental results, which can help calibrate the numerical model parameters. More focused experiments are needed to quantify adaptation, similar to the change in I_{min} all along the length of ulna for different number of bouts and frequencies of loading. We have calibrated the growth law parameters for the elastic material model based on the experimentally observed values of strain energy density in the ulna. Similar calibration was not possible for the case of the dissipation energy density stimulus, because similar data for a range of frequencies were not available. We need to develop experimental procedures to determine the threshold viscous fluid flow and corresponding dissipation energy on the osteocyte cells.

Once the growth law parameters are calibrated using such experimental results, the numerical model can be extended to incorporate the effect of other loading parameters such as number of loadcycles, time period of rest between bouts of loading etc. The model, as developed, is sufficiently general that any kind of geometry and mesh can be used for analysis. Once the parameters that can include the effect of the different loading parameters on adaptation are established, the numerical model can be extended to solve the inverse problem of determining the optimal loading parameters that will result in the desired bone shape and strength. Such studies will be extremely helpful to understand and prescribe exercise regimens to athletes (for desired bone strength), and osteoporotic patients and astronauts (to counter bone loss and help strengthen bone).

Appendix A

Bone adaptation code

The cortical bone adaptation code is written in FORTRAN. The main program is listed below. It calls several other subroutines, which are listed after the main program.

```
PROGRAM ADAPT_CORTICAL_BONE

C-----
C
C   CORTICAL BONE ADAPTATION PROGRAM
C   WRITTEN BY NATARAJAN CHENNIMALAI KUMAR, JUNE 7, 2010
C
C-----
C
C   PROGRAM TO IMPLEMENT THE NUMERICAL FRAMEWORK DEVELOPED IN
C   THE THESIS. THIS CODE FOLLOWS THE BONE GROWTH ALGORITHM:
C   (A) READ THE FE MESH AND LOADING
C   (B) PERFORM FE ANALYSIS
C   (C) READ RESULTS FILE
C   (D) DISPLACE SURFACE IN NORMAL DIRECTION BASED ON RESULTS
C   (E) CREATE NEW MODEL AND CONTINUE LOOP BY GOING TO STEP (A)
C
C-----
C   DEFINE ALL THE VARIABLES WITH RESPECT TO THE FE MODEL
C
C   ARRAYS FOR: NODE NUMBER, ELEMENT NUMBER, NODAL COORDINATES,
C   LM - ELEMENT CONNECTIVITY, SURFACE NODE, SURFACE NODE NEIGHBORS,
C   NODE NORMALS, AND NORMAL DISPLACEMENTS (B, B_TILDE, B_HAT)
C-----
      INTEGER NODE_NUMBER(100000), EL_NUMBER(100000)
      INTEGER LM(100000,20), INV_NODE(100000)
      INTEGER SURFACE_NODE(20000), NODAL_CONN(100000,21)
```

```

INTEGER SURFACE_NEIGHBOR(20000,30)
INTEGER N_NODES, N_ELEMS, N_SURF_NODES, N_LINES
REAL COORD(100000,3), SURF_DISP(20000,3), DISP(100000,3)
REAL ENER(100000)
REAL NODE_NORMAL(20000,3)
REAL B(20000), B_TILDE(20000), B_HAT(20000)
C-----
C CHARACTER ARRAYS FOR FILE NAMES AND TEXT TO BE READ FROM FILE
C-----
CHARACTER*120 TEXT_VERB(50000)
CHARACTER*43 FILE_DUMMY_NAME
CHARACTER*48 FILE_NAME
CHARACTER*45 INPUT_FILE_NAME
CHARACTER*45 DATA_FILE_NAME
CHARACTER*48 NEW_INP_FILE_NAME
CHARACTER*3 FILE_NUMBER
C-----
C GROWTH LAW VARIABLES: TIMESTEP, GLOBAL WEIGHT FOR SMOOTHING,
C PHI_REF (SENSITIVITY), PROP_CONSTANT (GAIN)
C INIT_TIME - INITIAL GROWTH TIMESTEP,
C N_TIME_STEPS - NUMBER OF GROWTH TIMESTEPS NEEDED
C-----
INTEGER STIMULUS, EL_TYPE, N_SMOOTH
INTEGER TIMESTEP, INIT_TIME, N_TIME_STEPS
REAL GLOBAL_WEIGHT, PROP_CONSTANT, PHI_REF
C-----
C POROELASTIC VARIABLES: PERMEABILITY - PERM, SPECIFIC WEIGHT
C FREQUENCY, T_COMPLETE - TIME FOR COMPLETE ANALYSIS (FOR INTEGRATION)
C-----
REAL PERM, T_COMPLETE, SP_WT, FREQUENCY
C-----
C ZONE OF INFLUENCE VARIABLES: ARRAYS CONTAINING NODES
C AND ELEMENTS INSIDE THE ZONE
C-----
INTEGER N_ZONE_NODES(20000)
INTEGER ZONE_OF_INFLUENCE(20000,2000)
INTEGER ZONE_OF_INFLUENCE_EL(20000,2000), N_ZONE_EL(20000)

INTEGER DUMMY_N_SURF, DUMMY_N_NODE, EOF, ERROR

```

```

C-----
GLOBAL_WEIGHT=0.5
N_SMOOTH=1

STIMULUS = 0
C STIMULUS IS A FLAG INDICATING WHAT IS THE STIMULUS REQUIRED TO BE READ
C STIMULUS = 0 MEANS STRAIN ENERGY DENSITY OR VELOCITY (POROELASTIC)
C = 1 MEANS STRESS/STRAIN (IE., TENSORIAL STIMULUS,
C WITH SIG33 OR E33)

EL_TYPE=1
C EL_TYPE IS ELEMENT TYPE INDICATING WHETHER IT IS A HEX OR TET
C EL_TYPE = 1 FOR A PARABOLIC TETRAHEDRAL ELEMENT
C = 2 FOR A LINEAR HEXAHEDRAL ELEMENT
C = 3 FOR A PARABOLIC HEXAHEDRAL ELEMENT
C LINEAR TETS ARE NOT TO BE USED IN THE ANALYSIS

OPEN(UNIT=19,FILE='REMODEL_INPUT.INP',STATUS='OLD')

C-----
C REMODEL_INPUT.INP FILE IS AN INPUT FILE CONTAINING THE FOLLOWING DETAILS:
C INPUT FILENAME (WITHOUT.INP EXTENSION)
C ELASTIC OR POROELASTIC MATERIAL FLAG (0-ELASTIC, 1-POROELASTIC)
C PROP_CONSTANT (GAIN A), PHI_REF
C INIT_TIME (STARTING TIMESTEP), N_TIME_STEPS (TOTAL NO OF TIMESTEPS)
C IF POROELASTIC MATERIAL, INPUT COMPLETE TIME OF SIMULATION,
C FREQUENCY OF LOADING, PERMEABILITY AND SPECIFIC WEIGHT OF FLUID
C-----
READ(19,*) INPUT_FILE_NAME
READ(19,*) MATERIAL_FLAG
IF(MATERIAL_FLAG.EQ.1) THEN
  READ(19,*) T_COMPLETE, FREQUENCY, PERM, SP_WT
ENDIF
READ(19,*) PROP_CONSTANT, PHI_REF
READ(19,*) INIT_TIME, N_TIME_STEPS

WRITE(*,*) 'PHI REF IS: ', PHI_REF

```

```
WRITE(*,*) T_COMPLETE, FREQUENCY, PERM
CLOSE(19)
```

C-----

```
IF(INIT_TIME.EQ.1) THEN
  FILE_NAME=TRIM(INPUT_FILE_NAME) // '.INP'
ELSE
  WRITE(FILE_NUMBER,130) INIT_TIME

  FILE_NAME=TRIM(INPUT_FILE_NAME)
$      // '- ' // FILE_NUMBER // '.INP'

ENDIF
```

C TRIM IS AN INTRINSIC FUNCTION TO TRIM TRAILING BLANKS IN A STRING

C-----

C CALL THE SUBROUTINE TO READ THE ABAQUS INPUT FILE

C-----

```
CALL READ_ABQ_INP(FILE_NAME, EL_TYPE, COORD,
$  NODE_NUMBER, N_NODES, N_ELEMS, EL_NUMBER, LM, N_DES_NODE,
$  N_SLICES, DESIGN_NODES, NODAL_CONN, SURFACE_NEIGHBOR,
$  N_SURF_NODES, SURFACE_NODE, NODE_NORMAL,
$  INV_NODE, N_ZONE_NODES, ZONE_OF_INFLUENCE,
$  N_ZONE_EL, ZONE_OF_INFLUENCE_EL, TEXT_VERB, N_LINES)
```

```
DUMMY_N_SURF=N_SURF_NODES
```

```
DUMMY_N_NODE=N_NODES
```

```
WRITE(*,*) 'ELEMENT NUMBER ', EL_NUMBER(10)
```

```
WRITE(*,*) 'N_SURF_NODES: ', N_SURF_NODES
```

```
WRITE(*,*) 'N_NODES: ', N_NODES
```

```
WRITE(*,*) 'N_ZONE_NODES: ', N_ZONE_NODES(1)
```

C-----

C LOOP THROUGH TIMESTEPS

C-----

```
DO TIMESTEP=INIT_TIME, N_TIME_STEPS
```

```
  WRITE(FILE_NUMBER,130) TIMESTEP
```

```

FILE_DUMMY_NAME=TRIM(INPUT_FILE_NAME)
$      // '- ' // FILE_NUMBER

WRITE(*,*) FILE_DUMMY_NAME

DATA_FILE_NAME=TRIM(FILE_DUMMY_NAME)//'.DAT'

IF(TIMESTEP.EQ.1) THEN

    CALL SYSTEM('CP ' // FILE_NAME // ' ' //
$          TRIM(FILE_DUMMY_NAME) //''.INP')

ENDIF

C-----
C      IF THE DATA FILE FROM ABAQUS ANALYSIS DOES NOT EXIST AT
C      TIMESTEP 1, CALL ABAQUS TO COMPLETE THE ANALYSIS
C-----

OPEN(UNIT=11, FILE=DATA_FILE_NAME, STATUS='OLD',
$     ERR=300, IOSTAT=EOF)
GOTO 400

300  ERROR=1

WRITE(*,*) 'ERR IS ', ERR
IF(ERROR.GE.0) THEN

    CALL SYSTEM('/USR/APPS/CSM/ABAQUS6.7-1/COMMANDS/ABQ671
$          JOB=' // TRIM(FILE_DUMMY_NAME)
$          //' USER=ORIENT_UPDATE_TET.F CPUS=8 INTER')

ENDIF

400  CLOSE(11)

WRITE(*,*) TRIM(FILE_DUMMY_NAME)//'.DAT'

C-----

```

```

C      IF ELASTIC MATERIAL, READ THE DATA FILE, AND COMPUTE THE
C      NORMAL SURFACE DISPLACEMENT BASED ON STRAIN ENERGY DENSITY
C-----
      IF(MATERIAL_FLAG.EQ.0) THEN
          CALL READ_ABQ_DAT(DATA_FILE_NAME, ENER, N_NODES, STIMULUS)

          CALL SURFACE_DISPLACEMENT_COMPUTE_SMOOTH(N_SURF_NODES,
$           SURFACE_NODE, N_NODES, INV_NODE, ENER,SURFACE_NEIGHBOR,
$           COORD, B, B_TILDE, B_HAT, PHI_REF, PROP_CONSTANT,
$           GLOBAL_WEIGHT, N_SMOOTH)
      ELSE
C-----
C      IF MATERIAL IS POROELASTIC, READ THE DATA FILE AND COMPUTE
C      NORMAL SURFACE DISPLACEMENT USING DISSIPATION ENERGY STIMULUS
C-----
          CALL READ_ABQ_PORO_DAT(FILE_DUMMY_NAME, ENER,
$           PERM, FREQUENCY, T_COMPLETE, STIMULUS, SP_WT)

          CALL SURFACE_DISPLACEMENT_COMPUTE_SMOOTH_PORO(N_SURF_NODES,
$           SURFACE_NODE, N_NODES, INV_NODE, ENER,SURFACE_NEIGHBOR,
$           COORD, B, B_TILDE, B_HAT, PHI_REF, PROP_CONSTANT,
$           GLOBAL_WEIGHT, N_SMOOTH,
$           N_ZONE_NODES, ZONE_OF_INFLUENCE,
$           N_ZONE_EL, ZONE_OF_INFLUENCE_EL, LM, EL_TYPE)

      ENDIF

C-----
C      COMPUTE THE SURFACE DISPLACEMENT ALONG THE COORDINATE DIRECTIONS
C-----
      DO I=1, N_SURF_NODES

          SURF_DISP(I,1)=NODE_NORMAL(I,1)*
$           B_HAT(I)
          SURF_DISP(I,2)=NODE_NORMAL(I,2)*
$           B_HAT(I)
          SURF_DISP(I,3)=NODE_NORMAL(I,3)*
$           B_HAT(I)
      ENDDO

```



```

C-----
C      WRITE THE INTERMEDIATE ABAQUS INPUT FILE TO COMPUTE THE
C      INNER NODES DISPLACEMENT FOR A SMOOTH MESH
C-----

OPEN(UNIT=20, FILE='SMOOTHING.INP',STATUS='REPLACE')
M=1
DO I=1,N_LINES
  WRITE(20,100) TEXT_VERB(I)
  IF(INDEX(TEXT_VERB(I),'*NODE').GT.0) THEN
    DO J=1,N_NODES
      WRITE(20,50) NODE_NUMBER(J), COORD(J,1), COORD(J,2),
$          COORD(J,3)
      ENDDO
      WRITE(20,100) '*****'
    ENDIF
    IF(INDEX(TEXT_VERB(I),'*ELEMENT').GT.0) THEN
      WRITE(*,*) 'WRITING ELEMENTS'
      IF(EL_TYPE.EQ.1) THEN
        WRITE(20,100) '*ELEMENT, TYPE=C3D10M,ELSET=FINE-MESH'
      ELSE IF(EL_TYPE.EQ.2) THEN
        WRITE(20,100) '*ELEMENT, TYPE=C3D8'
      ELSE IF(EL_TYPE.EQ.3) THEN
        WRITE(20,100) '*ELEMENT, TYPE=C3D20'
      ENDIF
      DO J=1,N_ELEMS
        IF(EL_TYPE.EQ.3) THEN
          WRITE(20,60) J, (LM(J,K), K=1,20)
        ELSE IF(EL_TYPE.EQ.1) THEN
          WRITE(20,61) J, (LM(J,K), K=1,10)
        ENDIF
      ENDDO
      WRITE(20,100) '*****'
    ENDIF
    IF(INDEX(TEXT_VERB(I),'*NSET').GT.0 .AND. INDEX(TEXT_VERB(I),
$      'SURFACE').GT.0) THEN
      DO J=1, N_SURF_NODES, 8
        WRITE(20,45) (SURFACE_NODE(K), K=J, J+7)
      ENDDO
      WRITE(20,100) '*****'
    ENDIF
  ENDIF
ENDIF

```

```

        IF(INDEX(TEXT_VERB(I),'*MATERIAL').GT.0 .OR.
$       INDEX(TEXT_VERB(I),'*MATERIAL').GT.0) THEN
        WRITE(20,100)'*ELASTIC'
        WRITE(20,140) 1.700, 0.3
        WRITE(20,100)'*****'
        WRITE(20,100)'*STEP'
        WRITE(20,100)'*STATIC, DIRECT'
        WRITE(20,100)'*BOUNDARY'
        WRITE(20,100)'INNER-SURFACE, 1, 3,'
C       WRITE(20,100)'FRONT_FACE, 1, 3'
C       WRITE(20,100)'BACK_FACE, 1, 3'
C       WRITE(20,100)'LEFT_FACE, 1, 1'
C       WRITE(20,100)'LEFT_FACE, 3, 3'
C       WRITE(20,100)'RIGHT_FACE, 1, 1'
C       WRITE(20,100)'RIGHT_FACE, 3, 3'
        DO J=1, N_SURF_NODES
            WRITE(20,120) (SURFACE_NODE(J)), 1, 1,
$                SURF_DISP(J,1)
            WRITE(20,120) (SURFACE_NODE(J)), 2, 2,
$                SURF_DISP(J,2)
            WRITE(20,120) (SURFACE_NODE(J)), 3, 3,
$                SURF_DISP(J,3)
        ENDDO
C       WRITE(20,*) 'OTHER, 1, 3,'
        WRITE(20,100)'*      OUTPUT, FIELD, VARIABLE=PRESELECT'
        WRITE(20,100)'*NODE OUTPUT'
        WRITE(20,100)'CF, RF, U'
        WRITE(20,100)'*ELEMENT OUTPUT'
        WRITE(20,100)'E, S, ENER'
        WRITE(20,100)'*NODE PRINT'
        WRITE(20,100)'U'
        WRITE(20,100)'*END STEP'
        GOTO 200
    ENDIF
ENDDO

200  CLOSE(20)

```

C-----

```

C      SOLVE THE ELASTIC BVP BY CALLING ABAQUS
C-----
      CALL SYSTEM('CP SMOOTHING.INP SMOOTHING-DUMMY.INP')

      CALL SYSTEM("/USR/APPS/CSM/ABAQUS6.7-1/COMMANDS/ABQ671
$      CPUS=8 JOB=SMOOTHING USER=ORIENT_UPDATE_TET.F INTER")

      DO I=1,N_NODES
          DISP(I,1:3)=(/0,0,0/)
      ENDDO

C-----
C      READ THE INTERMEDIATE ANALYSIS DATA FILE TO COMPUTE THE
C      DISPLACEMENT AT ALL THE NODES IN THE MESH
C-----
      CALL READ_ABQ_DAT_DISP(INV_NODE, DISP)

      DO I=1, N_NODES
          COORD(I,1)=COORD(I,1)+DISP(I,1)
          COORD(I,2)=COORD(I,2)+DISP(I,2)
          COORD(I,3)=COORD(I,3)+DISP(I,3)
      ENDDO

      WRITE(FILE_NUMBER,130) TIMESTEP+1

      FILE_DUMMY_NAME=TRIM(INPUT_FILE_NAME)
$      // '- ' // FILE_NUMBER

      NEW_INP_FILE_NAME=TRIM(FILE_DUMMY_NAME) //' .INP'
      WRITE(*,*) NEW_INP_FILE_NAME

C-----
C      WRITE THE ABAQUS INPUT FILE FOR THE NEW MESH FOR THE NEXT
C      GROWTH TIMESTEP
C-----
      OPEN(UNIT=10, FILE=NEW_INP_FILE_NAME,STATUS='NEW')
      M=1
      DO I=1,N_LINES

```

```

WRITE(10,100) TEXT_VERB(I)
IF(INDEX(TEXT_VERB(I),'*NODE').GT.0) THEN
  DO J=1,N_NODES
    WRITE(10,50) NODE_NUMBER(J), COORD(J,1), COORD(J,2),
$      COORD(J,3)
  ENDDO
  WRITE(10,*) '*****'
ENDIF
IF(INDEX(TEXT_VERB(I),'*ELEMENT').GT.0) THEN
  DO J=1,N_ELEMS
    IF(EL_TYPE.EQ.3) THEN
      WRITE(10,60) J, (LM(J,K), K=1,20)
    ELSE IF(EL_TYPE.EQ.1) THEN
      WRITE(10,61) J, (LM(J,K), K=1,10)
    ENDIF
  ENDDO
  WRITE(10,*) '*****'
ENDIF
IF((INDEX(TEXT_VERB(I),'*NSET').GT.0
$   .OR.INDEX(TEXT_VERB(I),'*NSET').GT.0)
$   .AND. INDEX(TEXT_VERB(I),
$   'SURFACE').GT.0) THEN
  DO J=1, N_SURF_NODES, 8
    WRITE(10,45) (SURFACE_NODE(K), K=J, J+7)
  ENDDO
ENDIF
ENDDO
CLOSE(10)

```

```
CALL SYSTEM('CP SMOOTHING.INP DUMMY.INP')
```

```

C-----
C   CALL THE ABAQUS SYSTEM COMMAND TO SOLVE THE ABAQUS INPUT FILE
C   OF THE NEW UPDATED MESH
C-----

```

```

CALL SYSTEM('/USR/APPS/CSM/ABAQUS6.7-1/COMMANDS/ABQ671
$   JOB=' // TRIM(FILE_DUMMY_NAME)
$   //' CPUS=8 USER=ORIENT_UPDATE_TET.F INTER')

```

ENDDO

```
C-----  
C      FORMAT STATEMENTS  
C-----  
40    FORMAT(I3,' ',I3,' ',I3,' ',I3,' ',I3,' ',I3,' ',I3,  
      $      ', ',I3)  
45    FORMAT(8(I6,' '))  
50    FORMAT(I7,' ',F16.8,' ',F16.8,' ',F16.5)  
60    FORMAT(20(I6,' '),I6)  
61    FORMAT(10(I6,' '),I6)  
80    FORMAT(120A1)  
100   FORMAT(A)  
110   FORMAT(6(15(I4,' ')/),12(I4,' '))  
120   FORMAT(I6,' ',I3,' ',I3,' ',E15.5)  
130   FORMAT(I3.3)  
140   FORMAT(F15.5,' ',F10.5)
```

END PROGRAM

```
SUBROUTINE READ_ABQ_INP(FILE_NAME, EL_TYPE, COOR, NODE_NO, N_NODE,  
$   N_ELEMS, EL_NO, LM, N_DES_NODE, N_SLICES,  
$   DESIGN_NODES, NODAL_CONN, SURFACE_NEIGHBOR,  
$   N_SURF_NODES, SURFACE_NODE, NODE_NORMAL,  
$   INV_NODE, N_ZONE_NODES, ZONE_OF_INFLUENCE,  
$   N_ZONE_EL, ZONE_OF_INFLUENCE_EL, TEXT_OUT, N_LINES)
```

```
CHARACTER TEXT(120), SLICE(80)  
CHARACTER*120 TEXT_OUT(50000)  
CHARACTER*120 LINE, SLICEDATA  
EQUIVALENCE(TEXT, LINE)  
EQUIVALENCE(SLICE,SLICEDATA)  
CHARACTER*48 FILE_NAME  
INTEGER N_NODE, N_ELEMS,N_SLICES, N_SURF_NODES, EL_TYPE  
INTEGER NODE_NO(100000), EL_NO(100000), LM(100000,20)  
INTEGER SURFACE_NODE(20000), NODAL_CONN(100000,21)  
INTEGER SURFACE_NEIGHBOR(20000,30), INV_NODE(100000)
```

```

INTEGER DESIGN_NODES(600,30)
REAL COOR(100000,3), NODE_NORMAL(20000,3)
REAL DVEC(3), NORM_MAG, AMAG, BMAG, CMAG, DMAG, NORMALA_MAG
REAL NORMALA1, NORMALA2, NORMALA3, NORMALB1, NORMALB2, NORMALB3
INTEGER EOF, ERR_CODE, DUMMY, N_DES_NODE, DUMMY_NODE
INTEGER SURF_NODE_INDEX, NODE_PER_EL, EL_INDEX, A
INTEGER DUMMY_NODE_NORMAL(5), DUMMY_NODE_NORMAL_EL(5)
INTEGER NODE_BLOCK(8000,35), BLOCK, BLOCK_INDEX(35)
INTEGER ZONE_OF_INFLUENCE(20000, 1000), ZONE_INDEX
INTEGER ZONE_OF_INFLUENCE_EL(20000, 1000)
INTEGER N_ZONE_NODES(20000), N_ZONE_EL(20000)

```

```

C-----
C   SUBROUTINE TO READ THE ABAQUS INPUT FILE AND OTHER PRE-PROCESSING
C   IN THIS SUBROUTINE, THE OUTER SURFACE NEIGHBORS ARE IDENTIFIED,
C   SURFACE NORMALS ARE COMPUTED AND THE ZONE OF INFLUENCE IS
C   DETERMINED
C-----

```

```

      I=1
      J=1
      K=1
      DUMMY=1
      EL_INDEX=1
      SURF_NODE_INDEX=1
      ERR_CODE=0
      A=1
      DO I=1,35
        BLOCK_INDEX(I)=1
      ENDDO
      BLOCK=1
      I=1

      IF(EL_TYPE.EQ.1) THEN
        NODE_PER_EL=10
      ELSE IF(EL_TYPE.EQ.2) THEN
        NODE_PER_EL=8
      ELSE IF(EL_TYPE.EQ.3) THEN
        NODE_PER_EL=20
      ENDIF

```

```

C-----
C   OPEN THE ABAQUS INPUT FILE
C-----
      OPEN(UNIT=10,FILE=FILE_NAME,STATUS='OLD',IOSTAT=EOF)
      WRITE(*,*) 'OPENED FILE', EOF
      WRITE(*,*) FILE_NAME
      DO WHILE(EOF.EQ.0)
        DUMMY=1
40      READ(10,910, END=30) TEXT
        TEXT_OUT(J)=LINE
        J=J+1
C-----
C   READ THE NODES AND NODAL COORDINATES
C-----
      IF(INDEX(LINE,'*NODE').GT.0) THEN
        N_LINE_BEFORE_NODE=J
        WRITE(*,*) 'ENCOUNTERED NODE'
        NODE_INDEX=1
        DO WHILE(ERR_CODE.EQ.0)
          READ(10,*,ERR=40,IOSTAT=ERR_CODE) NODE_NO(I),
$          COOR(I,1), COOR(I,2), COOR(I,3)
          IF(I.LE.4660) THEN
            WRITE(*,*) I,NODE_NO(I),COOR(I,1),COOR(I,2),COOR(I,3)
          ENDIF
          INV_NODE(NODE_NO(I))=I
C-----
C   SPLIT THE NODES INTO BLOCKS BASED ON THE Z-COORDINATE
C   TO SIMPLIFY THE SEARCH PROCESS TO DETERMINE THE ZONE OF INFLUENCE
C-----
          BLOCK=INT(COOR(I,3)/100)+1
          NODE_BLOCK(BLOCK_INDEX(BLOCK),BLOCK)=NODE_NO(I)
          BLOCK_INDEX(BLOCK)=BLOCK_INDEX(BLOCK)+1
          I=I+1
          NODE_INDEX=NODE_INDEX+1
        ENDDO
      ENDIF
C-----
C   READ THE ELEMENT NUMBERS AND CONNECTIVITY TABLE
C-----
      IF(INDEX(LINE,'*ELEMENT').GT.0) THEN

```

```

N_LINE_BEFORE_ELEMENT=J
WRITE(*,*) 'ENCOUNTERED ELEMENT ', ERR_CODE
EL_INDEX=1
DO WHILE(ERR_CODE.GE.0)
  READ(10,*, ERR=40, IOSTAT=ERR_CODE) EL_NO(EL_INDEX),
$      (LM(EL_INDEX,K), K=1,NODE_PER_EL)
  A=1
  DO I=1,NODE_PER_EL
    DO A=2,21
      IF(NODAL_CONN(LM(EL_INDEX,I),A).EQ.0) THEN
        NODAL_CONN(LM(EL_INDEX,I),A)=EL_INDEX
        EXIT
      ENDIF
    ENDDO
  ENDDO
  EL_INDEX=EL_INDEX+1
ENDDO
ENDIF
C-----
C   READ THE PERIOSTEAL SURFACE NODES ON WHICH GROWTH WILL BE ACTIVATED
C-----
  IF((INDEX(LINE,'*NSET').GT.0 .OR. INDEX(LINE,'*NSET').GT.0)
$   .AND. INDEX(LINE,'SURFACE').GT.0) THEN
  WRITE(*,*) 'ENCOUNTERED SURFACE'

  DO WHILE(ERR_CODE.GE.0)
    READ(10,*,ERR=40, IOSTAT=ERR_CODE)
$      (SURFACE_NODE(A),
$      A=SURF_NODE_INDEX,SURF_NODE_INDEX+7)
    DO I=SURF_NODE_INDEX, SURF_NODE_INDEX+7
      NODAL_CONN(SURFACE_NODE(I),1)=1
    ENDDO
    SURF_NODE_INDEX=SURF_NODE_INDEX+8
    IF(ERR_CODE.LT.0) THEN
      WRITE(*,*) 'EXITING SURFACE READ'
    ENDIF
  ENDDO

ENDIF
ENDDO

```



```

30  N_NODE=NODE_INDEX-1
    N_LINES=J-1
    N_ELEMS=EL_INDEX-1

    DO I=1, SURF_NODE_INDEX-1
      IF(SURFACE_NODE(I).EQ.0) THEN
        EXIT
      ENDIF
    ENDDO
    N_SURF_NODES=I-1
    N_INNER_SURF_NODES=K-1
    WRITE(*,*) N_NODE, N_LINES, N_ELEMS, SURF_NODE_INDEX, N_SURF_NODES
    K=0

```

```

C-----
C   CALL THE SUBROUTINE TO COMPUTE THE NORMALS AT THE SURFACE
C   NODES TO DETERMINE THE DIRECTION OF GROWTH
C-----

```

```

    CALL DETERMINE_NORMAL(N_SURF_NODES, SURFACE_NODE, COOR, LM,
$     INV_NODE, NODAL_CONN, NODE_PER_EL, NODE_NORMAL)

```

```

C-----
C   DETERMINE THE SURFACE NODE NEIGHBORS BASED ON THE
C   ELEMENT CONNECTIVITY
C-----

```

```

    DO I=1, N_SURF_NODES
      DUMMY=1
      DO J=1,30
        SURFACE_NEIGHBOR(I,J)=0
      ENDDO
      DO A=2,21
        IF(A.EQ.2) THEN
          NEIGHBOR_PRE_STEP=0
        ELSE
          NEIGHBOR_PRE_STEP=NEIGHBOR_INDEX
        ENDIF
        IF(NODAL_CONN(SURFACE_NODE(I),A).NE.0) THEN
          DO J=1,NODE_PER_EL
            DUMMY_NODE=LM(NODAL_CONN(SURFACE_NODE(I),A),J)

```

```

                IF(NODAL_CONN(DUMMY_NODE,1).EQ.1 .AND.
$                DUMMY_NODE.NE.SURFACE_NODE(I)) THEN
                DUMMY=0
                DO K=1,30
                    IF(SURFACE_NEIGHBOR(I,K).NE.DUMMY_NODE) THEN
                        DUMMY=DUMMY+1
                    ENDIF
                    IF(SURFACE_NEIGHBOR(I,K).EQ.0.AND.
$                    DUMMY.EQ.K) THEN
                        SURFACE_NEIGHBOR(I,K)=DUMMY_NODE
                        NEIGHBOR_INDEX=K
                        EXIT
                    ENDIF
                ENDDO
            ENDIF
        ENDDO
    ELSE
        EXIT
    ENDIF

1100    ENDDO

        ENDDO

        WRITE(*,*) 'N_SURF_NODES INSIDE READ: ', N_SURF_NODES

C-----
C    CALL THE SUBROUTINE TO DETERMINE THE ZONE OF INFLUENCE
C-----
        CALL DETERMINE_ZONE_OF_INFLUENCE(N_NODE, NODE_NO, COOR,
$        N_SURF_NODES, SURFACE_NODE, INV_NODE, NODAL_CONN,
$        NODE_BLOCK, BLOCK_INDEX,
$        ZONE_OF_INFLUENCE, N_ZONE_NODES,
$        ZONE_OF_INFLUENCE_EL, N_ZONE_EL)

        CLOSE(10)

10    FORMAT(15(I6,', '))
910   FORMAT(120A1)
940   FORMAT(I8, 3F16.12)

```

```

960 FORMAT(I3, 2A1)
950 FORMAT(I6, ', ', F10.5, ', ', F10.5, ', ', F10.5)

```

```

END SUBROUTINE

```

```

*****

```

```

SUBROUTINE READ_ABQ_DAT(FILE_NAME, ENER, N_NODES, STIMULUS)

```

```

CHARACTER*45 FILE_NAME
INTEGER DUMMY(6000), EL_NO(600), N_NODES
REAL ENER(100000), SENER(10000,8)
CHARACTER*80 LINE
CHARACTER TEXT(80)
EQUIVALENCE(TEXT, LINE)
INTEGER EOF, ERR_CODE, DES_INDEX, AB, STIMULUS

```

```

C-----
C SUBROUTINE TO READ THE STRAIN ENERGY DENSITY STIMULUS
C FROM THE ABAQUS DATA FILE
C-----

```

```

DES_INDEX=1
AB=1
ERR_CODE=0
EOF=0

```

```

WRITE(*,*) 'STIMULUS: ', STIMULUS
OPEN(UNIT=10, FILE=TRIM(FILE_NAME), STATUS='OLD', IOSTAT=EOF)
WRITE(*,*) 'OPENED FILE: ', EOF, ' ', TRIM(FILE_NAME)

```

```

DO WHILE(EOF.EQ.0)
40 READ(10,910, END=30, ERR=30) TEXT
IF(INDEX(LINE, ' E L E M E N T O U T P U T').GT.0) THEN
WRITE(*,*) 'ENCOUNTERED OUTPUT'
DO I=1,10
READ(10,910) TEXT
WRITE(*,*) TEXT
ENDDO
I=1
J=1

```

```

        WRITE(*,*) 'ERR_CODE: ', ERR_CODE
        DO WHILE(ERR_CODE.GE.0)
            IF(STIMULUS.EQ.1) THEN
                READ(10,*,ERR=40,IOSTAT=ERR_CODE) EL_NO(I),
$                DUMMY(I), DUMMY1, DUMMY2, SENER(I,J), DUMMY3,
$                DUMMY4, DUMMY5
            ELSE
                READ(10,*, ERR=40, IOSTAT=ERR_CODE) DUMMY1,
$                ENER(I)
            ENDIF
            I=I+1
        ENDIF
        IF(INDEX(LINE,'ERROR').GT.0) THEN
            WRITE(*,*) 'ERROR IN DAT FILE !!'
            CALL ABORT()
        ENDIF
    ENDDO

```

```

30  N_LINES=I-1
    WRITE(*,*) N_LINES
    WRITE(*,*) 'OUTSIDE READING'

```

```

    CLOSE(10)
    WRITE(*,*) 'FILE CLOSED'

```

```

910  FORMAT(80A1)

```

```

    RETURN
    END SUBROUTINE

```

```

*****
SUBROUTINE READ_ABQ_DAT_DISP(INV_NODE, DISP)

    INTEGER DUMMY(6000),EL_NO(600)
    REAL DISP(100000,3)
    CHARACTER*80 LINE
    CHARACTER TEXT(80)
    EQUIVALENCE(TEXT,LINE)
    INTEGER EOF, ERR_CODE, DES_INDEX,AB, STIMULUS, I
    INTEGER NODE_NO, INV_NODE(100000)

```

```

C-----
C   SUBROUTINE TO READ THE DISPLACEMENTS FROM THE INTERMEDIATE
C   DATA FILE IN ORDER TO GENERATE A SMOOTH MESH
C-----

ERR_CODE=0
EOF=0

OPEN(UNIT=10,FILE='SMOOTHING.DAT',STATUS='OLD',IOSTAT=EOF)
WRITE(*,*) 'OPENED FILE: ',EOF,' ',FILE_NAME

IF(EOF.NE.0) THEN
  CALL SYSTEM("/USR/APPS/CSM/ABAQUS6.7-1/COMMANDS/ABQ671
$   CPUS=8 JOB=SMOOTHING USER=ORIENT_UPDATE_TET.F INTER")
ENDIF

OPEN(UNIT=10,FILE='SMOOTHING.DAT',STATUS='OLD',IOSTAT=EOF)
WRITE(*,*) 'OPENED FILE: ',EOF,' ',FILE_NAME

DO WHILE(EOF.EQ.0)
40  READ(10,910, END=30, ERR=30) TEXT

  IF(INDEX(LINE,' N O D E   O U T P U T').GT.0) THEN
    WRITE(*,*) 'ENCOUNTERED OUTPUT'
    DO I=1,10
      READ(10,910) TEXT
      WRITE(*,*) TEXT
    ENDDO
    I=1
    J=1
    WRITE(*,*) 'ERR_CODE: ', ERR_CODE
    DO WHILE(ERR_CODE.GE.0)

      READ(10,*, ERR=40, IOSTAT=ERR_CODE), NODE_NO,
$         DISP(INV_NODE(NODE_NO),1),
$         DISP(INV_NODE(NODE_NO),2),
$         DISP(INV_NODE(NODE_NO),3)

      I=I+1
    ENDDO
  ENDIF
ENDWHILE

```

```

        ENDIF
        IF(INDEX(LINE,'ERROR').GT.0) THEN
            WRITE(*,*) 'ERROR IN DAT FILE !!'
            CALL ABORT()
        ENDIF
    ENDDO

30    N_LINES=I-1
        WRITE(*,*) N_LINES
        WRITE(*,*) 'OUTSIDE READING'

        CLOSE(10)
        WRITE(*,*) 'FILE CLOSED'

910   FORMAT(80A1)
920   FORMAT(3E10.5)

        RETURN
    END SUBROUTINE

```

```

SUBROUTINE SURFACE_DISPLACEMENT_COMPUTE_SMOOTH(N_SURF_NODES,
$    SURFACE_NODE, N_NODES, INV_NODE, ENER, SURFACE_NEIGHBOR,
$    COORD, B, B_TILDE, B_HAT, PHI_REF, PROP_CONSTANT,
$    GLOBAL_WEIGHT, N_SMOOTH)

```

```

    INTEGER N_SURF_NODES,SURFACE_NODE(20000), N_NODES,INV_NODE(100000)
    INTEGER SURFACE_NEIGHBOR(20000,30), N_SMOOTH
    REAL ENER(100000), COORD(100000,3), B(20000), B_TILDE(100000)
    REAL B_HAT(100000), B_DUMMY(100000), PHI_REF, PROP_CONSTANT
    REAL GLOBAL_WEIGHT
    INTEGER I, J, K
    INTEGER SMOOTH_NODE, DIST_DUMMY
    REAL DISTANCE, DISTANCE_SUM, WEIGHTING_SUM

```

I=1

```

C-----
C    COMPUTE THE SURFACE NODE DISPLACEMENT USING THE

```

C GROWTH LAW

```
C-----  
  
DO I=1, N_SURF_NODES  
  IF((ENER(INV_NODE(SURFACE_NODE(I)))-PHI_REF).GT.0.00) THEN  
    B(INV_NODE(SURFACE_NODE(I)))=PROP_CONSTANT*  
$      (ENER(INV_NODE(SURFACE_NODE(I)))-PHI_REF)  
  ELSE  
    B(INV_NODE(SURFACE_NODE(I)))=0.0  
  ENDIF  
  B_DUMMY(INV_NODE(SURFACE_NODE(I)))=  
$      B(INV_NODE(SURFACE_NODE(I)))  
  
ENDDO
```

```
C-----  
C LOOP THROUGH THE SURFACE NODES TO SMOOTH OUT THE  
C SURFACE NODE DISPLACEMENT BASED ON THE SURFACE NEIGHBORS  
C-----
```

```
DO N_SMOOTH=1,1  
  DO I=1, N_SURF_NODES  
    SMOOTH_NODE=SURFACE_NODE(I)  
    DISTANCE_SUM=0.0  
    WEIGHTING_SUM=0.0  
    CURRENT_STIMULUS=B_DUMMY(INV_NODE(SURFACE_NODE(I)))  
    DIST_DUMMY=0  
    DO J=1,30  
      IF(SURFACE_NEIGHBOR(I,J).NE.0) THEN  
        NEIGHBOR_NODE=SURFACE_NEIGHBOR(I,J)  
        DISTANCE=SQRT(  
$          (COORD(INV_NODE(SMOOTH_NODE),1)-  
$          COORD(INV_NODE(NEIGHBOR_NODE),1))**2+  
$          (COORD(INV_NODE(SMOOTH_NODE),2)-  
$          COORD(INV_NODE(NEIGHBOR_NODE),2))**2+  
$          (COORD(INV_NODE(SMOOTH_NODE),3)-  
$          COORD(INV_NODE(NEIGHBOR_NODE),3))**2)  
  
        DISTANCE_SUM=DISTANCE_SUM+DISTANCE  
        WEIGHTING_SUM=WEIGHTING_SUM+  
$          B_DUMMY(INV_NODE(NEIGHBOR_NODE))/DISTANCE
```

```

        DIST_DUMMY=DIST_DUMMY+1

        ELSE
            EXIT
        ENDIF
    ENDDO

    B_TILDE(INV_NODE(SURFACE_NODE(I)))=
$      WEIGHTING_SUM/DISTANCE_SUM
    B_HAT(INV_NODE(SURFACE_NODE(I)))=
$      GLOBAL_WEIGHT*CURRENT_STIMULUS
$      +(1-GLOBAL_WEIGHT)*WEIGHTING_SUM/DISTANCE_SUM

    B(INV_NODE(SURFACE_NODE(I)))=
$      B_TILDE(INV_NODE(SURFACE_NODE(I)))

    ENDDO

    DO I=1,N_SURF_NODES
        B_DUMMY(INV_NODE(SURFACE_NODE(I)))=
$      B(INV_NODE(SURFACE_NODE(I)))
    ENDDO
ENDDO

END SUBROUTINE

```

```

SUBROUTINE DETERMINE_NORMAL(N_SURF_NODES, SURFACE_NODE,
$   COOR, LM, INV_NODE, NODAL_CONN, NODE_PER_EL, NODE_NORMAL)

INTEGER LM(100000,20), N_SURF_NODES, NODE_PER_EL, EL_TYPE
INTEGER SURFACE_NODE(20000), NODAL_CONN(100000,21)
INTEGER INV_NODE(100000)
REAL NODE_NORMAL(20000,3), DET1
REAL AVEC(3), BVEC(3), COOR(100000,3)
REAL AMAG, BMAG, NORMALA_MAG, NORM_MAG
REAL NORMALA1, NORMALA2, NORMALA3
INTEGER DUMMY_NODE_NORMAL(10), DUMMY_NODE_NORMAL_EL(10)

```



```

INTEGER EL_INDEX, NORMAL_INDEX, NODE1, NODE2, NODE3
INTEGER SURF_NODE_EL_INDEX, I, J
INTEGER DUMMY_NODE, K, A, M, NODEA, NODEB, NODEC

```

```

C-----
C   SUBROUTINE TO COMPUTE THE NORMALS AT THE SURFACE NODES
C-----

```

```

DO I=1, N_SURF_NODES
  DO K=1,3
    AVEC(K)=0.0
    BVEC(K)=0.0
    NODE_NORMAL(I,K)=0.00
  ENDDO

```

```

  ENDDO
  DO A=2,21
    DO J=1,5
      DUMMY_NODE_NORMAL(J)=0
      DUMMY_NODE_NORMAL_EL(J)=0
    ENDDO

```

```

    NORMAL_INDEX=1
    EL_INDEX=1

```

```

    IF(NODAL_CONN(SURFACE_NODE(I),A).GE.1) THEN

```

```

C-----
C   NORMAL COMPUTATION SEGMENT
C   GO THROUGH THE ALL THE NODES OF AN ELEMENT
C-----

```

```

    NODE1=SURFACE_NODE(I)

```

```

    DO J=1, NODE_PER_EL
      DUMMY_NODE=LM(NODAL_CONN(SURFACE_NODE(I),A),J)
      IF(DUMMY_NODE.EQ.NODE1) THEN
        SURF_NODE_EL_INDEX=J
      ENDIF

```

```

    IF(NODAL_CONN(DUMMY_NODE,1).EQ.1.AND.
$      DUMMY_NODE.NE.SURFACE_NODE(I)) THEN
      DUMMY_NODE_NORMAL(NORMAL_INDEX)=DUMMY_NODE
      DUMMY_NODE_NORMAL_EL(EL_INDEX)=J
      NORMAL_INDEX=NORMAL_INDEX+1
      EL_INDEX=EL_INDEX+1

```

```

        ENDIF
    ENDDO

    IF(NORMAL_INDEX.EQ.1) THEN
        GOTO 1100
    ENDIF

    IF(NORMAL_INDEX-1.LE.2) THEN
        GOTO 1100
    ENDIF

C-----
C   TO COMPUTE THE NORMAL FOR A SURFACE NODE, THE SURFACE NODE
C   IS TAKEN AS NODE1. GO THROUGH THE NODAL NEIGHBORS TO
C   DETERMINE TWO OTHER NODES THAT FORM A TRIANGULAR PLANE.
C   USE THOSE TWO NODES TO COMPUTE TWO VECTORS A AND B,
C   A=(NODE2-NODE1), B=(NODE3-NODE1), AND THEN COMPUTE A CROSS B
C-----

        NODE1=SURFACE_NODE(I)
        NODE2=0
        NODE3=0
        DO K=1,3
            AVEC(K)=0.0
            BVEC(K)=0.0
        ENDDO

C-----
C   CASE 1: TETRAHEDRAL ELEMENT MESH
C-----

        IF(NODE_PER_EL.EQ.10) THEN

C-----
C   GO THROUGH THE NODAL NEIGHBORS TO FIND OUT NODE2 AND NODE3
C-----

        IF(SURF_NODE_EL_INDEX.EQ.1) THEN
            IF( ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
                $           .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7)
                $           .EQ.1) THEN
                NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
                NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
            ENDIF

```

```

IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
ENDIF

ELSE IF(SURF_NODE_EL_INDEX.EQ.2) THEN
      IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$          .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6)
$          .EQ.1) THEN
              NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
              NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
      ENDIF
      IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$          .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
$          .EQ.1) THEN
              NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
              NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
      ENDIF
      IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$          .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
$          .EQ.1) THEN
              NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
              NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
      ENDIF

ELSE IF(SURF_NODE_EL_INDEX.EQ.3) THEN
      IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$          .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7)
$          .EQ.1) THEN
              NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
              NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
      ENDIF

```

```

IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
ENDIF

ELSE IF(SURF_NODE_EL_INDEX.EQ.4) THEN
  IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
ENDIF
  IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
ENDIF
  IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
ENDIF

ELSE IF(SURF_NODE_EL_INDEX.EQ.5) THEN
  IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
ENDIF

```

```

IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
ENDIF

ELSE IF(SURF_NODE_EL_INDEX.EQ.6) THEN
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
ENDIF
ELSE IF(SURF_NODE_EL_INDEX.EQ.7) THEN
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
ENDIF
ELSE IF(SURF_NODE_EL_INDEX.EQ.8) THEN
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7) .EQ. 1
$
$
    .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
    .EQ.1) THEN
    NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
    NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
ENDIF

```

```

IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
ENDIF
ELSE IF (SURF_NODE_EL_INDEX.EQ.9) THEN
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,5) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),5)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,10)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),10)
ENDIF

ELSE IF (SURF_NODE_EL_INDEX.EQ.10) THEN
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,6) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,9)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),9)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),6)
ENDIF
IF (ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,7) .EQ. 1
$      .AND. ISMEMBER(DUMMY_NODE_NORMAL_EL,EL_INDEX-1,8)
$      .EQ.1) THEN
      NODE2=LM(NODAL_CONN(SURFACE_NODE(I),A),7)
      NODE3=LM(NODAL_CONN(SURFACE_NODE(I),A),8)
ENDIF

```

ENDIF

```

C-----
C   WE USED PARABOLIC HEXES ONLY FOR RECTANGULAR BEAM, SO
C   SURFACE NORMALS ARE JUST ALONG THE Y DIRECTION
C-----

```

ELSE IF (NODE_PER_EL.EQ.20) THEN

```

CONTINUE

ENDIF

IF(NODE2.EQ.0.OR.NODE3.EQ.0) THEN
  GOTO 1100
ENDIF

NODEA=NODE1
NODEB=NODE2
NODEC=NODE3

C-----
C   ONCE THE NODES ARE DETERMINED, THE POSITION VECTORS ARE COMPUTED
C-----

  NODE1=INV_NODE(SURFACE_NODE(I))
  NODE2=INV_NODE(NODEB)
  NODE3=INV_NODE(NODEC)

C-----
C   COMPUTE THE CROSS PRODUCT
C-----

900    DET1=COOR(NODE1,1)*(COOR(NODE2,2)*COOR(NODE3,3)-
$      COOR(NODE3,2)*COOR(NODE2,3))-COOR(NODE1,2)*
$      (COOR(NODE2,1)*COOR(NODE3,3)-COOR(NODE3,1)
$      *COOR(NODE2,3))+COOR(NODE1,3)*(COOR(NODE2,1)
$      *COOR(NODE3,2)-COOR(NODE3,1)*COOR(NODE2,2))

      AVEC(1)=COOR(NODE2,1)-COOR(NODE1,1)
      AVEC(2)=COOR(NODE2,2)-COOR(NODE1,2)
      AVEC(3)=COOR(NODE2,3)-COOR(NODE1,3)
      AMAG = SQRT(AVEC(1)**2+AVEC(2)**2+AVEC(3)**2)

      BVEC(1)=COOR(NODE3,1)-COOR(NODE1,1)
      BVEC(2)=COOR(NODE3,2)-COOR(NODE1,2)
      BVEC(3)=COOR(NODE3,3)-COOR(NODE1,3)
      BMAG = SQRT(BVEC(1)**2+BVEC(2)**2+BVEC(3)**2)

      NORMALA1=(AVEC(2)*BVEC(3)-AVEC(3)*BVEC(2))
      NORMALA2=(AVEC(3)*BVEC(1)-AVEC(1)*BVEC(3))

```

```
NORMALA3=(AVEC(1)*BVEC(2)-AVEC(2)*BVEC(1))
```

```
NORMALA_MAG=SQRT(NORMALA1**2+NORMALA2**2+NORMALA3**2)
```

```
C-----  
C   NORMALIZE THE CROSS PRODUCT TO COMPUTE THE NORMAL VECTOR  
C-----
```

```
          NODE_NORMAL(I,1)=NODE_NORMAL(I,1)+  
$          NORMALA1/NORMALA_MAG  
          NODE_NORMAL(I,2)=NODE_NORMAL(I,2)+  
$          NORMALA2/NORMALA_MAG  
          NODE_NORMAL(I,3)=NODE_NORMAL(I,3)+  
$          NORMALA3/NORMALA_MAG  
      ENDIF  
1100  ENDDO  
      NORM_MAG=SQRT(NODE_NORMAL(I,1)**2+NODE_NORMAL(I,2)**2+  
$      NODE_NORMAL(I,3)**2)  
      IF(NORM_MAG.NE.0) THEN  
          NODE_NORMAL(I,1)=NODE_NORMAL(I,1)/NORM_MAG  
          NODE_NORMAL(I,2)=NODE_NORMAL(I,2)/NORM_MAG  
          NODE_NORMAL(I,3)=NODE_NORMAL(I,3)/NORM_MAG  
      ENDIF  
  
      IF(NODE_PER_EL.EQ.20) THEN  
          NODE_NORMAL(I,1)=0.0  
          NODE_NORMAL(I,2)=1.0  
          NODE_NORMAL(I,3)=0.0  
      ENDIF  
  ENDDO  
  
END SUBROUTINE
```

```
*****
```

```
  SUBROUTINE DETERMINE_ZONE_OF_INFLUENCE(N_NODES, NODE_NO, COOR,  
$  N_SURFACE_NODES, SURFACE_NODE, INV_NODE, NODAL_CONN,  
$  NODE_BLOCK, BLOCK_INDEX,  
$  ZONE_OF_INFLUENCE, N_ZONE_NODES,  
$  ZONE_OF_INFLUENCE_EL, N_ZONE_EL)
```



```

C-----
C   SUBROUTINE TO DETERMINE THE NODES AND ELEMENTS
C   INSIDE THE ZONE OF INFLUENCE
C-----

      INTEGER, INTENT(IN):: N_NODES, N_SURFACE_NODES, INV_NODE(100000)
      INTEGER, INTENT(IN):: BLOCK_INDEX(35), NODAL_CONN(100000, 21)
      INTEGER, INTENT(IN):: SURFACE_NODE(20000), NODE_NO(100000)
      INTEGER, INTENT(IN):: NODE_BLOCK(8000,35)
      INTEGER, INTENT(OUT):: ZONE_OF_INFLUENCE(20000,1000)
      INTEGER, INTENT(OUT):: N_ZONE_NODES(20000)
      INTEGER, INTENT(OUT):: ZONE_OF_INFLUENCE_EL(20000,1000)
      INTEGER, INTENT(OUT):: N_ZONE_EL(20000)
      INTEGER ZONE_INDEX, ZONE_EL_INDEX, DUMMY_EL_INDEX
      INTEGER I, J, K, L, M, DUMMY
      REAL, INTENT(IN):: COOR(100000,3)
      REAL SPHERE_RADIUS, DISTANCE

      SPHERE_RADIUS=50.0
      ZONE_INDEX=1
      ZONE_EL_INDEX=0
      DUMMY_EL_INDEX=1

      WRITE(*,*) 'SPHERE RADIUS: ', SPHERE_RADIUS
      WRITE(*,*) 'N_SURFACE_NODES: ', N_SURFACE_NODES
      WRITE(*,*) 'COORD(1,1): ', COOR(1,1)

      DO I=1, N_SURFACE_NODES

C-----
C   LOOP THROUGH THE SURFACE NODES AND DETERMINE WHICH
C   BLOCK THE NODE IS IN. THIS IS DONE TO MAKE THE SEARCH
C   ALGORITHM EFFICIENT
C-----

      BLOCK=INT(COOR(INV_NODE(SURFACE_NODE(I)), 3)/100)+1
      ZONE_INDEX=1
      DUMMY_EL_INDEX=1
      DO J=BLOCK-1,BLOCK+1

```

```

C     SEARCH WITHIN ONE BLOCK AHEAD AND ONE BLOCK BEFORE
C-----
      DO K=1, BLOCK_INDEX(J)
        NODE1=INV_NODE(SURFACE_NODE(I))
        NODE2=INV_NODE(NODE_BLOCK(K, J))
        DISTANCE=SQRT((COOR(NODE1, 1)-COOR(NODE2, 1))**2+
$           (COOR(NODE1, 2)-COOR(NODE2, 2))**2+
$           (COOR(NODE1, 3)-COOR(NODE2, 3))**2)
C-----
C     STORE THE NODES THAT ARE WITHIN A THE RADIUS IN AN ARRAY
C-----
      IF(DISTANCE.LE.SPHERE_RADIUS .AND. DISTANCE.GT.0.00
$       .AND. NODAL_CONN(NODE_BLOCK(K, J), L).NE.1) THEN
        ZONE_OF_INFLUENCE(I, ZONE_INDEX)=NODE_BLOCK(K, J)
        ZONE_INDEX=ZONE_INDEX+1
C-----
C     STORE THE ELEMENTS ATTACHED TO THE NODES WITHIN THE ZONE
C-----
      DO L=2, 10
        IF(NODAL_CONN(NODE_BLOCK(K, J), L).NE.1) THEN
          IF(NODAL_CONN(NODE_BLOCK(K, J), L).NE.0) THEN
            DUMMY=0
            DO M=1, DUMMY_EL_INDEX-1
              IF(ZONE_OF_INFLUENCE_EL(I, M)
$                .NE.NODAL_CONN(NODE_BLOCK(K, J), L)) THEN
                DUMMY=DUMMY+1
              ENDIF
            ENDDO
            IF(DUMMY.EQ.DUMMY_EL_INDEX-1) THEN
              ZONE_OF_INFLUENCE_EL(I, DUMMY_EL_INDEX)=
$                NODAL_CONN(NODE_BLOCK(K, J), L)
              DUMMY_EL_INDEX=DUMMY_EL_INDEX+1
            ENDIF
          ENDIF
        ENDIF
      ENDDO
    ENDIF
  ENDDO
ENDDO

```

```

      N_ZONE_NODES(I)=ZONE_INDEX-1
      N_ZONE_EL(I)=DUMMY_EL_INDEX-1
ENDDO

```

```

10  FORMAT(15(I6,', '))
    END SUBROUTINE

```

```

      SUBROUTINE READ_ABQ_PORO_DAT(DUMMY_FILE_NAME, ENER,
$      PERM, FREQ, T_COMPLETE, STIMULUS, SP_WT)

```

```

C-----
C      SUBROUTINE TO COMPUTE THE TIME INTEGRAL OF THE
C      DISSIPATION POTENTIAL BY READING THE ABAQUS
C      DATA FILE FROM THE FE ANALYSIS
C-----

```

```

      CHARACTER*43 FILE_NAME, DUMMY_FILE_NAME, STA_FILE_NAME
      CHARACTER DUMMY_STRING(27)
      CHARACTER DUMMY_STRING2(37)

      INTEGER, INTENT(IN):: STIMULUS
      REAL, INTENT(IN):: T_COMPLETE, FREQ, PERM, SP_WT
      REAL VEL_M(100000,5000)
      REAL ENER(100000), TIME(5000)
      REAL DUMMY_ENER(5000)
      REAL SUM, PI
      CHARACTER*80 LINE
      CHARACTER TEXT(80)
      EQUIVALENCE(TEXT,LINE)
      INTEGER EOF, ERR_CODE, EOF1
      INTEGER A, TIME_MARKER, N_TIME, DUMMY1, DUMMY_SURF, N_TIME_STEPS
      REAL T_PERIOD, T_TOTAL, T_REM, POROSITY
      REAL ABQ_TIME_INT
      REAL DUMMY_TIME(600), DUMMY_INTEG(600), DUMMY_INT, STEP(5000)
      INTEGER DUMMY, DUMMY2, DUMMY3, DUMMY4
      REAL DUMMY5, DUMMY6

```

```

PI=3.141592653589793
AB=1
ERR_CODE=0
EOF=0
EOF1=0
A=1
SUM=0
POROSITY=0.05

WRITE(*,*) 'N_NODES: ', N_NODES

T_PERIOD=1/FREQ

C-----
C   T_COMPLETE - TOTAL TIME FOR WHICH DISSIPATION POTENTIAL TO BE
C   INTEGRATED, FREQ - FREQUENCY, PERM - PERMEABILITY
C   ALL INPUT PARAMETERS FROM THE INPUT FILE
C-----

WRITE(*,*) T_COMPLETE, FREQ, PERM

WRITE(*,*) 'N_SURF_NODES IN DAT READ: ', N_SURF_NODES
DUMMY_SURF=N_SURF_NODES
WRITE(*,*) 'STIMULUS: ', STIMULUS

FILE_NAME=TRIM(DUMMY_FILE_NAME) // '.DAT'

STA_FILE_NAME=TRIM(DUMMY_FILE_NAME) // '.STA'

WRITE(*,*) TRIM(DUMMY_FILE_NAME), STA_FILE_NAME

C-----
C   OPEN FIRST THE STA FILE TO READ THE TIMESTEP SIZE
C-----

OPEN(UNIT=12, FILE=STA_FILE_NAME, STATUS='OLD', IOSTAT=EOF1)

DO I=1,5
  READ(12,910) TEXT
  WRITE(*,*) TEXT

```

```

ENDDO
DO WHILE(EOF1.EQ.0)
  READ(12,*, ERR=50, IOSTAT=ERR_CODE)
$     DUMMY, A, DUMMY1, DUMMY2, DUMMY3, DUMMY4, DUMMY5,
$     DUMMY6,
$     STEP(A)

  IF(A.EQ.1) THEN
    TIME(A)=STEP(A)
  ELSE
    TIME(A)=TIME(A-1)+STEP(A)
  ENDIF
  WRITE(*,*) A, TIME(A), STEP(A)
ENDDO

50 DO I=1,A
  WRITE(*,*) STEP(I), TIME(I)
ENDDO

CLOSE(12)

N_TIME_STEPS=A
A=1

C-----
C   OPEN THE DAT FILE TO READ THE FE RESULTS FOR EACH TIMESTEP
C-----

OPEN(UNIT=10,FILE=FILE_NAME,STATUS='OLD',IOSTAT=EOF)
WRITE(*,*) 'OPENED FILE: ',EOF,' ',FILE_NAME

DO WHILE(EOF.EQ.0)
40  READ(10,910, END=30, ERR=30) TEXT
  IF(INDEX(LINE,'INCREMENT').GT.0 .AND.
$   INDEX(LINE,'SUMMARY').GT.0) THEN
    DO I=1,4
      READ(10,910) TEXT
    ENDDO
    READ(LINE,920,ERR=40,IOSTAT=ERR_CODE) DUMMY_STRING,
$   DUMMY_TIME(A),

```

```

$          DUMMY_STRING2
          WRITE(*,*) A, TIME(A)
          A=A+1
        ENDIF
        IF(INDEX(LINE,' E L E M E N T   O U T P U T').GT.0) THEN
          DO I=1,10
            READ(10,910) TEXT
          ENDDO
          I=1
          J=1
          DO WHILE(ERR_CODE.GE.0)
            IF(STIMULUS.EQ.1) THEN
              WRITE(*,*) 'STIMULUS NOT SUPPORTED'

              ELSE

C-----
C   READ ONLY THE MAGNITUDE OF VELOCITY FOR COMPUTATION
C   OF THE DISSIPATION POTENTIAL
C-----
              READ(10,*, ERR=40, IOSTAT=ERR_CODE) DUMMY1,
$              DUMMY2, DUMMY3, VEL_M(I,A-1),
$              DUMMY2, DUMMY3, DUMMY4
            ENDIF

            I=I+1

          ENDDO
        ENDIF
        IF(INDEX(LINE,'ERROR').GT.0) THEN
          WRITE(*,*) 'ERROR IN DAT FILE !!'
          CALL ABORT()
        ENDIF
      ENDDO

30  N_LINES=I-1
      WRITE(*,*) 'N_LINES:', N_LINES
      WRITE(*,*) 'OUTSIDE READING'

      WRITE(*,*) 'A = ',A
      CLOSE(10)

```

```

WRITE(*,*) 'FILE CLOSED'

IF(TIME(A-1)-TIME(A-2).LE.1E-6) THEN
  A=A-1
ENDIF

C-----
C   T_TOTAL = TOTAL TIME PERIOD FOR WHICH FE ANALYSIS IS DONE
C   T_PERIOD = TIME PERIOD OF LOADING (1/FREQUENCY)
C   T_COMPLETE = COMPLETE TIME PERIOD OF EXPERIMENT FOR WHICH
C               INTEGRATION SHOULD BE PERFORMED
C-----

T_TOTAL=TIME(A-1)
WRITE(*,*) T_PERIOD, T_TOTAL, T_COMPLETE, N_LINES

DO K=1, N_LINES
C-----
C   COMPUTE DISSIPATION POTENTIAL=1/2*1/PERMEABILITY*V^2*POROSITY
C-----
  DO I=1, A-1
    DUMMY_ENER(I)=0.5*SP_WT/PERM*POROSITY**2*VEL_M(K,I)**2
  ENDDO

C-----
C   COMPUTE THE INTEGRAL OF THE DISSIPATION POTENTIAL
C   ABQ_TIME_INT - MEANS THE TIME INTEGRAL UPTO THE
C   TIME OF ANALYSIS USING ABAQUS
C-----
  CALL INTEGRATE(TIME, DUMMY_ENER, A-1, ABQ_TIME_INT)

  IF(ABS(T_TOTAL-T_COMPLETE).LE.1E-4) THEN

    ENER(K)=ABQ_TIME_INT

C-----
C   IF THE TIME OF EXPERIMENT IS GREATER THAN THE TIME PERIOD
C   OF ANALYSIS USING ABAQUS, USE THE LAST CYCLE OF LOADING
C   AND EXTRAPOLATE IT TO COMPUTE THE DISSIPATION ENERGY
C-----

```

```

ELSE IF(T_COMPLETE.GT.T_TOTAL) THEN

    DO J=A-1,1,-1
        IF(ABS(TIME(J)-(T_TOTAL-T_PERIOD)).LE.MAXVAL(STEP)) THEN
            TIME_MARKER=J
            GOTO 10
        ENDIF
    ENDDO

10    N_PERIOD=(A-1)-TIME_MARKER
    T_REM=(T_COMPLETE-T_TOTAL)
    N_TIME=CEILING(T_REM/T_PERIOD)

    DO I=1, N_PERIOD
        DUMMY_TIME(I)=TIME(TIME_MARKER+I)-
$         TIME(TIME_MARKER)

        DUMMY_INTEG(I)=DUMMY_ENER(TIME_MARKER+I)
    ENDDO

    CALL INTEGRATE(DUMMY_TIME, DUMMY_INTEG, N_PERIOD,
$         DUMMY_INT)

    ENER(K)=(N_TIME*DUMMY_INT+ABQ_TIME_INT)

ELSE IF(T_COMPLETE.LT.T_TOTAL) THEN
C-----
C   IF THE TIME OF EXPERIMENT IS LESS THAN THE TIME PERIOD
C   OF ANALYSIS USING ABAQUS, COMPUTE THE DISSIPATION
C   ENERGY ONLY UPTO THE TIME OF EXPERIMENT
C-----
    DO J=1, A
        IF(ABS(TIME(J)-(T_COMPLETE)).LE.MAXVAL(STEP)) THEN

            TIME_MARKER=J

            GOTO 60
        ENDIF
    ENDDO

```



```

60      DO I=1, TIME_MARKER

          DUMMY_TIME(I)=TIME(I)
          DUMMY_INTEG(I)=DUMMY_ENER(I)
        ENDDO

          CALL INTEGRATE(DUMMY_TIME, DUMMY_INTEG, TIME_MARKER,
$          DUMMY_INT)

          ENER(K)=DUMMY_INT

        ENDIF
      ENDDO

20     WRITE(*,*) 'LAST STATEMENT'
      WRITE(*,*) 'N_NODES: ', N_NODES, 'N_SURF_NODES: ', N_SURF_NODES
      RETURN

910    FORMAT(80A1)
920    FORMAT(27A1, E9.3, 37A1)
930    FORMAT(I3,I3,4I3,2F6.5, E8.6)
940    FORMAT(40(E10.4,TR4))

      END SUBROUTINE

```

```

SUBROUTINE INTEGRATE(X, Y, N, INTEGRAL)

```

```

REAL X(1000), Y(1000), INTEGRAL
REAL X0, X1, X2, Y0, Y1, Y2
INTEGER I, J, N

```

```

C-----
C   SUBROUTINE TO COMPUTE THE INTEGRAL OF
C   A FUNCTION DEFINED AT UNEQUAL INTERVALS
C-----

```

```

INTEGRAL=0.0

DO I=1,N-1,2

    X0=X(I)
    X1=X(I+1)
    X2=X(I+2)

    Y0=Y(I)
    Y1=Y(I+1)
    Y2=Y(I+2)

    IF(I+2.GT.N) THEN
        X0=X(I-1)
        X1=X(I)
        X2=X(I+1)

        Y0=Y(I-1)
        Y1=Y(I)
        Y2=Y(I+1)

        INT_01=(Y1-Y0)/(X1-X0)
        INT_12=(Y2-Y1)/(X2-X1)
        INT_012=(INT_12-INT_01)/(X2-X0)

        INTEGRAL=INTEGRAL+Y0*(X2-X1)+INT_01/2*
$           ((X2-X0)**2-(X1-X0)**2)
$           +((X2**3)/3-(X2**2)*(X1+X0)/2+X0*X1*X2-(X1**3)/3
$           +(X1**2)*(X1+X0)/2-X0*X1**2)*INT_012

    ELSE
        IF((X2-X1).NE.0) THEN
            INTEGRAL=INTEGRAL+(X2-X0)*(Y0+(X2-X0)
$                *(Y1-Y0)/(2*(X1-X0)))+
$                (2*X2**2-X0*X2-X0**2+3*X0*X1-3*X1*X2)/6
$                *((Y2-Y1)/(X2-X1)-(Y1-Y0)/(X1-X0))

        ENDIF

    ENDIF

```

```

ENDDO
RETURN
END SUBROUTINE

```

```

SUBROUTINE SURFACE_DISPLACEMENT_COMPUTE_SMOOTH_PORO(N_SURF_NODES,
$ SURFACE_NODE, N_NODES, INV_NODE, ENER, SURFACE_NEIGHBOR,
$ COORD, B, B_TILDE, B_HAT, PHI_REF, PROP_CONSTANT,
$ GLOBAL_WEIGHT, N_SMOOTH,N_ZONE_NODES, ZONE_OF_INFLUENCE,
$ N_ZONE_EL, ZONE_OF_INFLUENCE_EL, LM, EL_TYPE)

```

```

C-----
C SUBROUTINE TO COMPUTE THE SURFACE NODAL DISPLACEMENT
C BASED ON THE DISSIPATION ENERGY STIMULUS
C-----

```

```

INTEGER, INTENT(IN):: N_SURF_NODES, N_NODES
INTEGER, INTENT(IN):: SURFACE_NODE(20000), INV_NODE(100000)
INTEGER, INTENT(IN):: SURFACE_NEIGHBOR(20000,30)
INTEGER, INTENT(IN):: N_ZONE_NODES(20000), EL_TYPE,
$ ZONE_OF_INFLUENCE(20000,1000), LM(100000,20)
INTEGER, INTENT(IN):: ZONE_OF_INFLUENCE_EL(20000, 1000),
$ N_ZONE_EL(20000)
REAL ENER(100000), COORD(100000,3)
REAL B(20000), B_TILDE(20000)
REAL B_HAT(20000), B_DUMMY(20000)
REAL, INTENT(IN):: PHI_REF, PROP_CONSTANT
REAL ENER_ZONE(20000), ZONE_DISTANCE, INV_DIST_SUM
REAL GLOBAL_WEIGHT, INT_PT, INT_PT1, INT_PT2
REAL DISTANCE_SCALE, EXP_WEIGHT
INTEGER I, J, K, N_SMOOTH, KL, ELEMENT_NUM
INTEGER SMOOTH_NODE, DIST_DUMMY, N_INTPTS, EL_NODE(20)
REAL DISTANCE, DISTANCE_SUM, WEIGHTING_SUM, INTPTS(8,3)
REAL SHAPE(20), DSDPSI(20,3), JACOBIAN, X(3), JACOBIAN_MATRIX(3,3)
REAL ENER_INT_PT, DIST_ZONE, WEIGHT(8), X_NODE(3), SCALE, CUTOFF
REAL SPHERE_RADIUS

```

```

C-----
C IDENTIFY THE INTEGRATION POINTS FOR VOLUME INTEGRAL

```

C USING THE FE FORMULATION

C-----

```
IF(EL_TYPE.EQ.3) THEN
  N_INT_PTS=1
```

C-----

C CHOICE OF EITHER ONE OR EIGHT INTEGRATION POINTS FOR
C THE PARABOLIC HEX ELEMENT

C-----

```
NODE_PER_EL=20
IF(N_INT_PTS.EQ.8) THEN
  INT_PT=SQRT(1.0/3.0)
  INT_PTS(1,1:3)=(-INT_PT, -INT_PT, -INT_PT/)
  INT_PTS(2,1:3)=(-INT_PT, -INT_PT, INT_PT/)
  INT_PTS(3,1:3)=(-INT_PT, INT_PT, -INT_PT/)
  INT_PTS(4,1:3)=(-INT_PT, INT_PT, INT_PT/)
  INT_PTS(5,1:3)=(/INT_PT, -INT_PT, -INT_PT/)
  INT_PTS(6,1:3)=(/INT_PT, -INT_PT, INT_PT/)
  INT_PTS(7,1:3)=(/INT_PT, INT_PT, -INT_PT/)
  INT_PTS(8,1:3)=(/INT_PT, INT_PT, INT_PT/)
  WEIGHT(1:8)=1.00
ELSE IF(N_INT_PTS.EQ.1) THEN
  INT_PTS(1,1:3)=(/0, 0, 0/)
  WEIGHT(1)=8
ENDIF
ELSE IF(EL_TYPE.EQ.1) THEN
  N_INT_PTS=1
```

C-----

C CHOICE OF EITHER ONE OR FOUR INTEGRATION POINTS FOR
C THE PARABOLIC TETRAHEDRAL ELEMENT

C-----

```
NODE_PER_EL=10
IF(N_INT_PTS.EQ.4) THEN
  INT_PT1=(5-SQRT(5.0))/20
  INT_PT2=(5+3*SQRT(5.0))/20
  INT_PTS(1,1:3)=(/INT_PT1, INT_PT1, INT_PT2/)
  INT_PTS(2,1:3)=(/INT_PT1, INT_PT2, INT_PT1/)
  INT_PTS(3,1:3)=(/INT_PT2, INT_PT1, INT_PT1/)
```

```

        INT_PTS(4,1:3)=(/INT_PT1, INT_PT1, INT_PT1/)
        WRITE(*,*) INT_PT1, INT_PT2
        WRITE(*,*) ((INT_PTS(AB,AC), AC=1,3),AB=1,4)
        WEIGHT(1:4)=1.0/24
    ELSE IF(N_INT_PTS.EQ.1) THEN
        INT_PTS(1,1:3)=(/0.25, 0.25, 0.25/)
        WEIGHT(1)=1.0/6
    ENDIF
ENDIF
WRITE(*,*) 'N_INT_PTS: ', N_INT_PTS

C-----
C   LOOP THROUGH THE SURFACE NODES
C-----

    DO I=1, N_SURF_NODES

        NODE1=INV_NODE(SURFACE_NODE(I))
        ENER_ZONE(I)=0.0
        DIST_ZONE=0.0

C-----
C   LOOP THROUGH THE ELEMENTS IN THE ZONE OF INFLUENCE
C   FOR EACH SURFACE NODE
C-----

        DO J=1, N_ZONE_EL(I)
            DO M=1, N_INT_PTS
                CALL SHAPE_FUNCTION(EL_TYPE, INT_PTS(M,1),
$                   INT_PTS(M,2), INT_PTS(M,3), SHAPE, DSDPSI)

                X(1:3)=(/0, 0, 0/)
                ENER_INT_PT=0.0
                DO KL=1,3
                    DO KM=1,3
                        JACOBIAN_MATRIX(KL,KM)=0.0
                    ENDDO
                ENDDO
                ELEMENT_NUM=ZONE_OF_INFLUENCE_EL(I,J)
                DO K=1,NODE_PER_EL

```

```

EL_NODE(K)=LM(ZONE_OF_INFLUENCE_EL(I,J),K)
DO KL=1, 3
  X_NODE(KL)=COORD(INV_NODE(EL_NODE(K)),KL)
  X(KL)=X(KL)+SHAPE(K)*X_NODE(KL)
ENDDO

```

```

IF(ENER(ELEMENT_NUM).LE.0.0) THEN
  WRITE(*,*) 'NEGATIVE ENERGY:',
$      ELEMENT_NUM,
$      ENER(ELEMENT_NUM)
  ENER(ELEMENT_NUM)=
$      ABS(ENER(ELEMENT_NUM))
ENDIF

```

```

C-----
C   SUM THE ENERGIES AT THE INTEGRATION POINTS FOR EACH ELEMENT
C-----

```

```

$      ENER_INT_PT=ENER_INT_PT+
$      ENER(ELEMENT_NUM)*SHAPE(K)

```

```

$      JACOBIAN_MATRIX(1,1)=JACOBIAN_MATRIX(1,1)+
$      DSDPSI(K,1)*X_NODE(1)
$      JACOBIAN_MATRIX(1,2)=JACOBIAN_MATRIX(1,2)+
$      DSDPSI(K,1)*X_NODE(2)
$      JACOBIAN_MATRIX(1,3)=JACOBIAN_MATRIX(1,3)+
$      DSDPSI(K,1)*X_NODE(3)

```

```

$      JACOBIAN_MATRIX(2,1)=JACOBIAN_MATRIX(2,1)+
$      DSDPSI(K,2)*X_NODE(1)
$      JACOBIAN_MATRIX(2,2)=JACOBIAN_MATRIX(2,2)+
$      DSDPSI(K,2)*X_NODE(2)
$      JACOBIAN_MATRIX(2,3)=JACOBIAN_MATRIX(2,3)+
$      DSDPSI(K,2)*X_NODE(3)

```

```

$      JACOBIAN_MATRIX(3,1)=JACOBIAN_MATRIX(3,1)+
$      DSDPSI(K,3)*X_NODE(1)
$      JACOBIAN_MATRIX(3,2)=JACOBIAN_MATRIX(3,2)+
$      DSDPSI(K,3)*X_NODE(2)

```

```

          JACOBIAN_MATRIX(3,3)=JACOBIAN_MATRIX(3,3)+
$          DSDPSI(K,3)*X_NODE(3)

          ENDDO

C-----
C   COMPUTE THE JACOBIAN AT THE INTEGRATION POINTS
C-----

          JACOBIAN=JACOBIAN_MATRIX(1,1)*
$          (JACOBIAN_MATRIX(2,2)*JACOBIAN_MATRIX(3,3)
$          -JACOBIAN_MATRIX(2,3)*JACOBIAN_MATRIX(3,2))
$          -JACOBIAN_MATRIX(1,2)*
$          (JACOBIAN_MATRIX(2,1)*JACOBIAN_MATRIX(3,3)
$          -JACOBIAN_MATRIX(2,3)*JACOBIAN_MATRIX(3,1))
$          +JACOBIAN_MATRIX(1,3)*
$          (JACOBIAN_MATRIX(2,1)*JACOBIAN_MATRIX(3,2)
$          -JACOBIAN_MATRIX(3,1)*JACOBIAN_MATRIX(2,2))

C-----
C   COMPUTE THE DISTANCE FROM THE SURFACE NODE TO THE
C   ELEMENT INTEGRATION POINT
C-----

          DISTANCE=SQRT((X(1)-COORD(NODE1,1))**2
$          +(X(2)-COORD(NODE1,2))**2
$          +(X(3)-COORD(NODE1,3))**2)

          SPHERE_RADIUS=45.0

          DISTANCE_SCALE=DISTANCE/SPHERE_RADIUS

C-----
C   COMPUTE THE EXPONENTIAL INFLUENCE FUNCTION WEIGHT
C   BASED ON THE DISTANCE
C-----

          EXP_WEIGHT=100*(DISTANCE_SCALE)**3
$          *EXP(-11*DISTANCE_SCALE)

C-----
C   COMPUTE THE VOLUME INTEGRAL OF THE DISSIPATION ENERGY

```

```

C      STIMULUS OVER THE ZONE OF INFLUENCE
C-----
          ENER_ZONE(I)=ENER_ZONE(I)+ENER_INT_PT*
$          JACOBIAN*WEIGHT(M)*EXP_WEIGHT
          DIST_ZONE=DIST_ZONE+JACOBIAN*WEIGHT(M)
$          *EXP_WEIGHT

          ENDDO
ENDDO

          ENER_ZONE(I)=ENER_ZONE(I)/DIST_ZONE

          WRITE(*,*) 'FINAL OUTPUT: ', (SURFACE_NODE(I)), ENER_ZONE(I)

          ENDDO
C-----
C      COMPUTE THE SURFACE NODE DISPLACEMENT USING THE
C      GROWTH LAW
C-----
          DO I=1, N_SURF_NODES
              IF((ENER_ZONE(I)-PHI_REF).GT.0.00) THEN
                  B(I)=PROP_CONSTANT*
$                  (ENER_ZONE(I)-PHI_REF)
              ELSE
                  B(I)=0.0
              ENDIF
              B_DUMMY(I)=
$              B(I)

          ENDDO

C-----
C      LOOP THROUGH THE SURFACE NODES TO SMOOTH OUT THE
C      SURFACE NODE DISPLACEMENT BASED ON THE SURFACE NEIGHBORS
C-----
          DO N_SMOOTH=1,1
              DO I=1, N_SURF_NODES
                  SMOOTH_NODE=SURFACE_NODE(I)
                  DISTANCE_SUM=0.0
                  WEIGHTING_SUM=0.0

```



```

CURRENT_STIMULUS=B_DUMMY(I)
DIST_DUMMY=0
DO J=1,30
  IF(SURFACE_NEIGHBOR(I,J).NE.0) THEN
    NEIGHBOR_NODE=SURFACE_NEIGHBOR(I,J)
    DISTANCE=SQRT(
$      (COORD(INV_NODE(SMOOTH_NODE),1)-
$      COORD(INV_NODE(NEIGHBOR_NODE),1))**2+
$      (COORD(INV_NODE(SMOOTH_NODE),2)-
$      COORD(INV_NODE(NEIGHBOR_NODE),2))**2+
$      (COORD(INV_NODE(SMOOTH_NODE),3)-
$      COORD(INV_NODE(NEIGHBOR_NODE),3))**2)

    DISTANCE_SUM=DISTANCE_SUM+1/DISTANCE
    WEIGHTING_SUM=WEIGHTING_SUM+
$      B_DUMMY(I)/DISTANCE
    DIST_DUMMY=DIST_DUMMY+1

  ELSE
    EXIT
  ENDIF
ENDDO

  B_TILDE(I)=
$    WEIGHTING_SUM/DISTANCE_SUM
  B_HAT(I)=
$    GLOBAL_WEIGHT*CURRENT_STIMULUS
$    +(1-GLOBAL_WEIGHT)*WEIGHTING_SUM/DISTANCE_SUM

  B(I)=
$    B_HAT(I)
ENDDO

DO I=1,N_SURF_NODES
  B_DUMMY(I)=
$    B(I)
ENDDO
ENDDO

DO I=1,N_SURF_NODES

```

```

        WRITE(*,*) (SURFACE_NODE(I)),
$         COORD(INV_NODE(SURFACE_NODE(I)),3),
$         B_HAT(I)
    ENDDO

```

```

END SUBROUTINE

```

```

*****

```

```

SUBROUTINE SHAPE_FUNCTION(EL_TYPE, PSI, ETA, ZETA, SHAPE, DSDPSI)

```

```

    INTEGER, INTENT(IN):: EL_TYPE
    REAL, INTENT(IN):: PSI, ETA, ZETA
    REAL SHAPE(20), DSDPSI(20,3)
    INTEGER I, J, K
    REAL COORD(20,3)

```

```

C-----
C   SUBROUTINE TO COMPUTE THE SHAPE FUNCTIONS AND ITS
C   DERIVATIVES FOR THE DIFFERENT ELEMENT TYPES
C   EL_TYPE=3 MEANS PARABOLIC HEX
C-----

```

```

    IF(EL_TYPE.EQ.3) THEN

```

```

C-----
C   NODAL COORDINATES OF THE PARABOLIC HEXAHEDRAL ISOPARAMETRIC ELEMENT
C-----

```

```

        COORD(1,1:3)=(/-1, -1, -1/)
        COORD(2,1:3)=(/1, -1, -1/)
        COORD(3,1:3)=(/1, 1, -1/)
        COORD(4,1:3)=(/-1, 1, -1/)
        COORD(5,1:3)=(/-1, -1, 1/)
        COORD(6,1:3)=(/1, -1, 1/)
        COORD(7,1:3)=(/1, 1, 1/)
        COORD(8,1:3)=(/-1, 1, 1/)

```

```

        COORD(9,1:3)=(/0, -1, -1/)

```

```

COORD(10,1:3)=(/1, 0, -1/)
COORD(11,1:3)=(/0, 1, -1/)
COORD(12,1:3)=(/-1, 0, -1/)

```

```

COORD(13,1:3)=(/0, -1, 1/)
COORD(14,1:3)=(/1, 0, 1/)
COORD(15,1:3)=(/0, 1, 1/)
COORD(16,1:3)=(/-1, 0, 1/)

```

```

COORD(17,1:3)=(/-1, -1, 0/)
COORD(18,1:3)=(/1, -1, 0/)
COORD(19,1:3)=(/1, 1, 0/)
COORD(20,1:3)=(/-1, 1, 0/)

```

```

C-----
C   SHAPE FUNCTIONS AND DERIVATIVES OF THE PARABOLIC HEX
C-----

```

```

DO I=1,8
  SHAPE(I)=(1+PSI*COORD(I,1))*(1+ETA*COORD(I,2))*
$         (1+ZETA*COORD(I,3))*(PSI*COORD(I,1)+ETA*COORD(I,2)
$         +ZETA*COORD(I,3)-2)/8

  DSDPSI(I,1)=((COORD(I,1))*(1+ETA*COORD(I,2))*
$         (1+ZETA*COORD(I,3))*(PSI*COORD(I,1)+ETA*COORD(I,2)
$         +ZETA*COORD(I,3)-2)+(1+PSI*COORD(I,1))*
$         (1+ETA*COORD(I,2))*(1+ZETA*COORD(I,3))*COORD(I,1))/8

  DSDPSI(I,2)=((1+PSI*COORD(I,1))*(COORD(I,2))*
$         (1+ZETA*COORD(I,3))*(PSI*COORD(I,1)+ETA*COORD(I,2)
$         +ZETA*COORD(I,3)-2)+(1+PSI*COORD(I,1))*
$         (1+ETA*COORD(I,2))*(1+ZETA*COORD(I,3))*COORD(I,2))/8

  DSDPSI(I,3)=((1+PSI*COORD(I,1))*(1+ETA*COORD(I,2))*
$         (COORD(I,3))*(PSI*COORD(I,1)+ETA*COORD(I,2)
$         +ZETA*COORD(I,3)-2)+(1+PSI*COORD(I,1))*
$         (1+ETA*COORD(I,2))*(1+ZETA*COORD(I,3))*COORD(I,3))/8

ENDDO

```

```

DO I=9,15,2
  SHAPE(I)=(1-PSI**2)*(1+ETA*COORD(I,2))*(1+ZETA*COORD(I,3))/4

  DSDPSI(I,1)=-2*PSI*(1+ETA*COORD(I,2))*(1+ZETA*COORD(I,3))/4
  DSDPSI(I,2)=(1-PSI**2)*(COORD(I,2))*(1+ZETA*COORD(I,3))/4
  DSDPSI(I,3)=(1-PSI**2)*(1+ETA*COORD(I,2))*(COORD(I,3))/4
ENDDO
DO I=10,16,2
  SHAPE(I)=(1+PSI*COORD(I,1))*(1-ETA**2)*(1+ZETA*COORD(I,3))/4

  DSDPSI(I,1)=COORD(I,1)*(1-ETA**2)*(1+ZETA*COORD(I,3))/4
  DSDPSI(I,2)=(1+PSI*COORD(I,1))*(-2*ETA)*
$      (1+ZETA*COORD(I,3))/4
  DSDPSI(I,3)=(1+PSI*COORD(I,1))*(1-ETA**2)*(COORD(I,3))/4
ENDDO
DO I=17,20
  SHAPE(I)=(1+PSI*COORD(I,1))*(1+ETA*COORD(I,2))*(1-ZETA**2)/4

  DSDPSI(I,1)=COORD(I,1)*(1+ETA*COORD(I,2))*(1+ZETA**2)/4
  DSDPSI(I,2)=(1+PSI*COORD(I,1))*(COORD(I,2))*(1+ZETA**2)/4
  DSDPSI(I,3)=(1+PSI*COORD(I,1))*(1+ETA*COORD(I,2))*
$      (-2*ZETA)/4
ENDDO

```

```

C-----
C   SHAPE FUNCTIONS AND DERIVATIVES FOR THE PARABOLIC
C   TETRAHEDRAL ISOPARAMETRIC ELEMENT
C-----

```

```

ELSE IF(EL_TYPE.EQ.1) THEN
  SHAPE(1)=2*(1-PSI-ETA-ZETA)*(0.5-PSI-ETA-ZETA)

  DSDPSI(1,1:3)=2*(-1)*(0.5-PSI-ETA-ZETA)+2*(1-PSI-ETA-ZETA)*(-1)

  SHAPE(2)=2*PSI*(PSI-0.5)

  DSDPSI(2,1)=2*PSI+2*(PSI-0.5)
  DSDPSI(2,2)=0.0
  DSDPSI(2,3)=0.0

```

$$\text{SHAPE}(3)=2*\text{ETA}*(\text{ETA}-0.5)$$

$$\text{DSDPSI}(3,2)=2*\text{ETA}+2*(\text{ETA}-0.5)$$

$$\text{DSDPSI}(3,1)=0.0$$

$$\text{DSDPSI}(3,3)=0.0$$

$$\text{SHAPE}(4)=2*\text{ZETA}*(\text{ZETA}-0.5)$$

$$\text{DSDPSI}(4,3)=2*\text{ZETA}+2*(\text{ZETA}-0.5)$$

$$\text{DSDPSI}(4,2)=0.0$$

$$\text{DSDPSI}(4,1)=0.0$$

$$\text{SHAPE}(5)=4*\text{PSI}*(1-\text{PSI}-\text{ETA}-\text{ZETA})$$

$$\text{DSDPSI}(5,1)=4*(1-\text{PSI}-\text{ETA}-\text{ZETA})+4*\text{PSI}*(-1)$$

$$\text{DSDPSI}(5,2)=4*\text{PSI}*(-1)$$

$$\text{DSDPSI}(5,3)=4*\text{PSI}*(-1)$$

$$\text{SHAPE}(6)=4*\text{PSI}*\text{ETA}$$

$$\text{DSDPSI}(6,1)=4*\text{ETA}$$

$$\text{DSDPSI}(6,2)=4*\text{PSI}$$

$$\text{DSDPSI}(6,3)=0.0$$

$$\text{SHAPE}(7)=4*\text{ETA}*(1-\text{PSI}-\text{ETA}-\text{ZETA})$$

$$\text{DSDPSI}(7,2)=4*(1-\text{PSI}-\text{ETA}-\text{ZETA})+4*\text{ETA}*(-1)$$

$$\text{DSDPSI}(7,1)=4*\text{ETA}*(-1)$$

$$\text{DSDPSI}(7,3)=4*\text{ETA}*(-1)$$

$$\text{SHAPE}(8)=4*\text{ZETA}*(1-\text{PSI}-\text{ETA}-\text{ZETA})$$

$$\text{DSDPSI}(8,3)=4*(1-\text{PSI}-\text{ETA}-\text{ZETA})+4*\text{ZETA}*(-1)$$

$$\text{DSDPSI}(8,2)=4*\text{ZETA}*(-1)$$

$$\text{DSDPSI}(8,1)=4*\text{ZETA}*(-1)$$

$$\text{SHAPE}(9)=4*\text{PSI}*\text{ZETA}$$

$$\text{DSDPSI}(9,1)=4*\text{ZETA}$$

$$\text{DSDPSI}(9,2)=0.0$$

DSDPSI(9,3)=4*PSI

SHAPE(10)=4*ETA*ZETA

DSDPSI(10,1)=0.0

DSDPSI(10,2)=4*ZETA

DSDPSI(10,3)=4*ETA

ENDIF

END SUBROUTINE

Appendix B

ABAQUS Input Deck

```
**-----  
** RAT-ULNA-POROELASTIC-ORTHO-PERM.INP                JUNE 7, 2010  
**-----  
**  
**                DECK WRITTEN AND MESH GENERATED BY  
**                NATARAJAN CHENNIMALAI KUMAR  
**  
** PROBLEM: POROELASTIC ANALYSIS OF THE RAT ULNA SUBJECTED TO  
**           COMPRESSIVE LOADING OF 9 N AT 2 HZ FREQUENCY  
**  
** GEOMETRY: MICRO-CT IMAGES PROVIDED BY PROF.CHARLES TURNER  
**           DEPT. OF ORTHOPEADIC SURGERY, IUPUI, INDIANAPOLIS  
**  
** MESH:      GENERATED USING ALTAIR HYPERMESH  
**  
** 62782 NODES NOTE: Distorted elements are present as  
** 37949 ELEMENTS      a result of the curved bone geometry  
**  
**-----  
** UNITS:  length - 1E-5 m  
**         time   - s  
**         mass   - kg  
**         force  - N  
**-----  
** MATERIAL PROPERTIES - INPUT FOR ANALYSIS  
**  
** Drained elastic modulus      E      = 17          GPa  
** Drained Poisson's ratio      nu = 0.3  
** Solid Grain Bulk Modulus     Ks     = 16.29       GPa  
** Material Frame Bulk Modulus  K      = 14.16       GPa  
** Fluid Bulk Modulus           Kf     = 2.3         GPa
```

```

** Porosity                n_p      = 0.05
** Void Ratio              e          = 5.262E-02
** Fluid Specific Weight   rw        = 9.881E+03      N/m^3
** Fluid Density           rho_f     = 1.000E+03      kg/m^3
** Fluid Viscosity         mu        = 1.000E-03      Pa s
**
** PERMEABILITY VALUE
**
** Intrinsic Permeability  k          = 3.000E-20      m^2
** Hydraulic Conductivity  kappa      = 2.943E-13      m/s
**
** LOADING DETAILS
**
** Loading frequency       freq      = 2              Hz
** Loading magnitude       F          = 9              N
** Loading period          Period    = 2              s
** Timestep                dt         = 2.000E-03      s
** Number of steps         N_steps  = 1000
** Estimated run time      rtime     = 8              hrs
**
**-----
*Heading
rat-ulna-poroelastic-ortho
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
*ORIENTATION, NAME=radial_theta, SYSTEM=USER
*****
*NODE
.....node number, nodal coordinates.....
.....
*****
*Element, type=C3D10MP
.....Element number, nodal connectivity.....
.....
*****
*Elset, elset=TET-MESH, generate
    1, 37949, 1
*Nset, nset=ALLNODES, generate
    1, 62782, 1
** Section: Section-1-TET-MESH

```



```

*Solid Section, elset=TET-MESH, ORIENTATION=radial_theta, material=BONE
1.,
*****
** List of nodes on the outer surface
** where growth is activated
*****
*NSET, nset=surface
.....surface node list.....
.....
*****
*Nset, nset=constraint-node
4,
*Nset, nset=point-load-node
5323,
*****
** List of inner surface nodes for
** the application of the zero pressure
** boundary condition
*****
*Nset, nset=INNER-SURFACE
.....Inner surface node list.....
.....
*****
** List of outer surface elements for
** the application of the zero pressure
** boundary condition
*****
*Elset, elset=_Outer-surf_S1
.....outer surface element list.....
.....
*Elset, elset=_Outer-surf_S2
.....outer surface element list.....
.....
*Elset, elset=_Outer-surf_S3
.....outer surface element list.....
.....
*Elset, elset=_Outer-surf_S4
.....outer surface element list.....
.....
*****

```

```

*Surface, type=ELEMENT, name=Outer-surf
_Outer-surf_S1, S1
_Outer-surf_S2, S2
_Outer-surf_S4, S4
_Outer-surf_S3, S3
*****
*** Length unit (lu) is 10 microns or 10x10^-6 m
*** So conversion of units needed
*** k_bar goes from 1.472E-11 m/s to 1.472e-6 (lu)/s
*** sp weight 9.81E3 N/m^3 to 9.81E-12 N/lu^3
*** modulus 17 GPa to 1.7 N/lu^2
*****
**
** MATERIALS
**
*****
*Material, name=BONE
*Elastic
  1.7, 0.3
*Permeability, type=orthotropic, specific=9.81e-12
2.9430e-8, 2.9430E-12, 2.943e-08, 0.05
*Porous Bulk Moduli
1.6292,0.23
*****
**INITIAL CONDITIONS
*****
*INITIAL CONDITIONS, TYPE=RATIO
ALLNODES, 0.05, 0
*INITIAL CONDITIONS, TYPE=SATURATION
ALLNODES, 1
*INITIAL CONDITIONS, TYPE=PORE PRESSURE
ALLNODES, 0
*****
*AMPLITUDE, NAME=SINLOAD, DEFINITION=PERIODIC, VALUE=RELATIVE
1, 12.56637, 0.0, 0.5
-0.5, 0.0
*****
** STEP: Step-1
*****
*STEP, INC = 5000, NAME = Step-1

```

```

rat-ulna-poroelastic
*Soils, consolidation, end=PERIOD, utol=1e+10
2.0e-3, 2.0, 2.0e-3, 2.0e-3,
*SFLOW
Outer-surf, Q, 0, 0
**bottom_face, Q2, 0, 0
**HNAME LOADCOL          1 HM_Load_Cols_1
**HWCOLOR LOADCOL        1      5
*CLOAD, amplitude=SINLOAD
      5323, 1, 0.2554
      5323, 2, 0.2087
      5323, 3, -8.9940
**HNAME LOADCOL          2 constraints
**HWCOLOR LOADCOL        2      3
*BOUNDARY
      4, 1, 3, 0.0
24894, 1, 3, 0.0
      5323, 1, 2, 0.0
INNER-SURFACE, 8, 8,
*EL PRINT, position=centroid
SENER, POR, FLVEL
*OUTPUT, FIELD, VARIABLE = PRESELECT, FREQUENCY = 2
*Node OUTPUT
CF, RF, U, POR
*ELEMENT OUTPUT
E, S, ENER, FLVEL
*END STEP
*****

```



```

C FIND ELEMENT CENTROID, ZMIN AND ZMAX
DO N=1,4
  DO I=1,3
    CENTROID(I) = CENTROID(I) + CNODES(I,N)
  ENDDO
  ZMIN = MIN(ZMIN,CNODES(3,N))
  ZMAX = MAX(ZMAX,CNODES(3,N))
ENDDO
DO I=1,3
  CENTROID(I) = CENTROID(I)/4
ENDDO
IF(DEBUG)THEN
  WRITE(*,*)'CENTROID: ',CENTROID,' ZMIN,ZMAX',ZMIN,ZMAX
ENDIF
C GET CENTERLINE LOCATION FROM FINDCL, THEN COMPUTE FUNDAMENTAL VECTOR
CALL FINDCL(CENTROID(3),CENTERLINE,CENTERLINE_1)
IF(DEBUG)THEN
  WRITE(*,*)'CENTERLINE: ',CENTERLINE
ENDIF
TSUM = ZERO
DO J=1,3
  T(J,1) = CENTERLINE(J) - CENTROID(J)
  TSUM = TSUM + T(J,1)**2
ENDDO
DO J=1,3
  T(J,1) = T(J,1)/SQRT(TSUM)
ENDDO
C CONSTRUCT PUTATIVE AXIS FROM THE CENTERLINE LOCATION
DO J=1,3
  T(J,3) = CENTERLINE_1(J) - CENTERLINE(J)
  TSUM = TSUM + T(J,3)**2
ENDDO
DO J=1,3
  T(J,3) = T(J,3)/SQRT(TSUM)
ENDDO
C NOW GET THE REST OF THE VECTORS BY CROSS PRODUCTS
CALL CROSS_PROD(T(1,1),T(1,3),T(1,2),IERR)
IF(IERR.NE.0)THEN
  WRITE(*,*)'CROSS PRODUCT FAILED - BAILING OUT'
  RETURN

```

```

ENDIF
CALL CROSS_PROD(T(1,1),T(1,2),T(1,3),IERR)
C DO I=1,3
C  WRITE(*,*) (T(I,J), J=1,3)
C ENDDO
RETURN
END
SUBROUTINE CROSS_PROD(A,B,C,IERR)
C
C ROUTINE TO COMPUTE THE CROSS PRODUCT C = A X B
C THE RESULT IS NORMALIZED
C
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
DIMENSION A(3),B(3),C(3)
C
C(1) = A(2)*B(3) - A(3)*B(2)
C(2) = A(3)*B(1) - A(1)*B(3)
C(3) = A(1)*B(2) - A(2)*B(1)
CNORM = SQRT(C(1)**2 + C(2)**2 + C(3)**2)
IF(CNORM.LT.1.D-8)THEN
  WRITE(*,*)'ERROR IN CROSS PRODUCT: CNORM < MINIMUM'
  IERR=1
ELSE
  DO I=1,3
    C(I) = C(I)/CNORM
  ENDDO
  IERR = 0
ENDIF
RETURN
END
SUBROUTINE FINDCL(ZCTR,CENTERLINE,CENTERLINE_1)
C
C ROUTINE TO FIND THE APPROPRIATE AVERAGE LOCATION OF THE BONE
C CENTERLINE BY SEARCHING ALONG THE CENTERLINE PATH.
C
IMPLICIT DOUBLE PRECISION (A-H, O-Z)
DIMENSION CENTERLINE(3),CENTERLINE_1(3), CTR_MIN(3),CTR_MAX(3)
DIMENSION CLVECTOR(3,317)
LOGICAL FIRST
SAVE CLVECTOR,IPTR,FIRST,IDATA

```

```

DATA FIRST, IDATA, TWO /.TRUE., 0, 2.0D0/
IPTR=1
C
IF(FIRST)THEN
  FIRST = .FALSE.
  OPEN(21, FILE="/HOME/AC/CHENNIMA/CENTERLINE.DAT", STATUS='OLD')
10  IDATA = IDATA+1
  READ(21, *, END=20, ERR=20) (CLVECTOR(I, IDATA), I=1, 3)
C  WRITE(*, *) (CLVECTOR(I, IDATA), I=1, 3)
  GO TO 10
20  IDATA = IDATA - 1
  CLOSE(21)
ENDIF
C SEARCH DATAFILE
30 IF (CLVECTOR(3, IPTR) .LE. ZCTR .AND. CLVECTOR(3, IPTR+1) .GE. ZCTR) THEN
  CENTERLINE(1) = (CLVECTOR(1, IPTR)+CLVECTOR(1, IPTR+1))/TWO
  CENTERLINE(2) = (CLVECTOR(2, IPTR)+CLVECTOR(2, IPTR+1))/TWO
  CENTERLINE(3) = (CLVECTOR(3, IPTR)+CLVECTOR(3, IPTR+1))/TWO
  CENTERLINE_1(1)=CLVECTOR(1, IPTR+1)
  CENTERLINE_1(2)=CLVECTOR(2, IPTR+1)
  CENTERLINE_1(3)=CLVECTOR(3, IPTR+1)
  RETURN
ELSEIF(ZCTR.LT.CLVECTOR(3, IPTR))THEN
  IPTR = IPTR-1
  IF(IPTR.LE.0)THEN
    WRITE(*, *) 'SOMETHING IS WRONG. ZCTR IS LESS THAN MINIMUM',
      +
      ' CENTERLINE DATA'
    CENTERLINE(1) = (CLVECTOR(1, 1)+CLVECTOR(1, 2))/TWO
    CENTERLINE(2) = (CLVECTOR(2, 1)+CLVECTOR(2, 2))/TWO
    CENTERLINE(3) = (CLVECTOR(3, 1)+CLVECTOR(3, 2))/TWO
    RETURN
  ENDIF
  GO TO 30
ELSE
  IPTR = IPTR+1
  IF(IPTR.GE.IDATA)THEN
    WRITE(*, *) 'SOMETHING IS WRONG. ZCTR IS LARGER THAN MAXIMUM',
      +
      ' CENTERLINE DATA'
    CENTERLINE(1) = (CLVECTOR(1, IDATA-1)+CLVECTOR(1, IDATA))/TWO
    CENTERLINE(2) = (CLVECTOR(2, IDATA-1)+CLVECTOR(2, IDATA))/TWO

```

```
CENTERLINE(3) = (CLVECTOR(3, IDATA-1)+CLVECTOR(3, IDATA))/TWO
C  WRITE(*,*) ZCTR, (CENTERLINE(I), I=1,3)
  RETURN
ENDIF
GO TO 30
ENDIF
40 CONTINUE
END
```


References

- [1] A. G. Robling, F. M. Hinant, D. B. Burr, and C. H. Turner. Improved bone structure and strength after long-term mechanical loading is greatest if loading is separated into short bouts. *Journal of Bone and Mineral Research*, 17(8):1545–1554, 2002.
- [2] A. G. Robling, A. B. Castillo, and C. H. Turner. Biomechanical and molecular regulation of bone remodeling. *Annual Review of Biomedical Engineering*, 8:455–498, 2006.
- [3] Y. F. Hsieh and C. H. Turner. Effects of loading frequency on mechanically induced bone formation. *Journal of Bone Mineral Research*, 16:918–924, 2001.
- [4] S. J. Warden and C. H. Turner. Mechanotransduction in cortical bone is most efficient at loading frequencies of 5-10 hz. *Bone*, 34(2):261–270, 2004.
- [5] D. B. Burr, A. G. Robling, and C. H. Turner. Effects of biomechanical stress on bones in animals. *Bone*, 30(5):781–786, 2002.
- [6] A. G. Robling, D. B. Burr, and C. H. Turner. Recovery periods restore mechanosensitivity to dynamically loaded bone. *Journal of Experimental Biology*, 204(19):3389–3399, 2001.
- [7] L. C. Junqueira and J. Carnerio. *Basic Histology*. Lange McGraw-Hill, 10th ed., New York, 2003.
- [8] H. M. Frost. Skeletal structural adaptations to mechanical usage (satmu): 2. redefining wolff’s law: the remodeling problem. *The Anatomical Record*, 226(4):414–422, 1990.
- [9] H. M. Frost. Skeletal structural adaptations to mechanical usage (satmu): 1. redefining wolff’s law: The bone modeling problem. *The Anatomical Record*, 226(4):403–413, 1990.
- [10] Stephen C. Cowin. *Bone Mechanics Handbook*. CRC Press, New York,, 2001.
- [11] A. D. Leblanc, V. S. Schneider, H. J. Evans, D. A. Engelbretson, and J. M. Krebs. Bone mineral loss and recovery after 17 weeks of bed rest. *Journal of Bone Mineral Research*, 5:843–850, 1990.
- [12] M. B. Schaffler, K. Choi, and C. Milgrom. Aging and matrix microdamage accumulation in human compact bone. *Bone*, 17(6):521–525, 1995.
- [13] M. Devas. *Stress Fractures*. Churchill-Livingston, New York, 1975.

- [14] C. H. Turner. Three rules for bone adaptation to mechanical stimuli. *Bone*, 23(5):399–407, 1998.
- [15] L. E. Lanyon, A. E. Goodship, C. J. Pye, and J. H. MacFie. Mechanically adaptive bone remodelling. *Journal of Biomechanics*, 15(3):141–154, 1982.
- [16] C. T. Rubin and L. E. Lanyon. Regulation of bone mass by mechanical strain magnitude. *Calcified Tissue International*, 37(4):411–417, 1985.
- [17] A. G. Robling, D. B. Burr, and C. H. Turner. Partitioning a daily mechanical stimulus into discrete loading bouts improves the osteogenic response to loading. *Journal of Bone and Mineral Research*, 15(8):1596–1602, 2000.
- [18] L. E. Lanyon and C. T. Rubin. Static vs dynamic loads as an influence on bone remodelling. *Journal of Biomechanics*, 17(12):897–905, 1984.
- [19] M. P. Akhter, D. M. Raab, C. H. Turner, D. B. Kimmel, and R. R. Recker. Characterization of in vivo strain in the rat tibia during external application of a four-point bending load. *Journal of Biomechanics*, 25(10):1241–1246, 1992.
- [20] C. Rubin, A. S. Turner, S. Bain, C. Mallinckrodt, and K. McLeod. Low mechanical signals strengthen long bones. *Nature*, 412(6847):603–604, 2001.
- [21] S. C. Cowin and D. H. Hegedus. Bone remodeling i: theory of adaptive elasticity. *Journal of Elasticity*, 6(3):313–325, 1976.
- [22] P. Fridez, L. Rakotomanana, A. Terrier, and P. F. Leyvraz. Three dimensional model of external bone adaptation. *Computer Methods in Biomechanics and Biomedical Engineering 2*, pages 189–196, 1998.
- [23] R. Huiskes, H. Weinans, H. J. Grootenboer, M. Dalstra, B. Fudala, and T. J. Slooff. Adaptive bone-remodeling theory applied to prosthetic-design analysis. *Journal of Biomechanics*, 20(11-12):1135–1150, 1987.
- [24] R. Huiskes, R. Ruimerman, G. H. van Lenthe, and J. D. Janssen. Effects of mechanical forces on maintenance and adaptation of form in trabecular bone. *Nature*, 405(6787):704–706, 2000.
- [25] P. J. Prendergast and D. Taylor. Prediction of bone adaptation using damage accumulation. *Journal of Biomechanics*, 27(8):1067–1076, 1994.
- [26] H. Weinans, R. Huiskes, and H. J. Grootenboer. The behavior of adaptive bone-remodeling simulation models. *Journal of Biomechanics*, 25(12):1425–1441, 1992.
- [27] B. P. McNamara, P. J. Prendergast, and D. Taylor. Prediction of bone adaptation in the ulnar-osteotomized sheep’s forelimb using an anatomical finite element model. *Journal of Biomedical Engineering*, 14(3):209–216, 1992.

- [28] D. R. Carter, T. E. Orr, and D. P. Fyhrie. Relationships between loading history and femoral cancellous bone architecture. *Journal of Biomechanics*, 22(3):231–244, 1989.
- [29] D. R. Carter, M. C. H. Van der Meulen, and G. S. Beaupré. Mechanical factors in bone growth and development. *Bone*, 18(1, Supplement 1):S5–S10, 1996.
- [30] M. Doblaré and J. M. García. Anisotropic bone remodelling model based on a continuum damage-repair theory. *Journal of Biomechanics*, 35(1):1–17, 2002.
- [31] C. T. Rubin and L. E. Lanyon. Regulation of bone formation by applied dynamic loads. *Journal of Bone and Joint Surgery*, 66(3):397–402, 1984.
- [32] G. S. Beaupré, T. E. Orr, and D. R. Carter. An approach for time-dependent bone modeling and remodeling - theoretical development. *Journal of Orthopedic Research*, 8(5):651–661, 1990.
- [33] S. C. Cowin and K. Firoozbakhsh. Bone remodeling of diaphysial surfaces under constant load: Theoretical predictions. *Journal of Biomechanics*, 14(7):471–484, 1981.
- [34] G. Martinez, J. M. García Aznar, M. Doblaré, and M. Cerrolaza. External bone remodeling through boundary elements and damage mechanics. *Mathematics and Computers in Simulation*, 73(1-4 SPEC ISS):183–199, 2006.
- [35] M. C. H. van der Meulen, G. S. Beaupré, and D. R. Carter. Mechanobiologic influences in long bone cross-sectional growth. *Bone*, 14:635–642, 1993.
- [36] Amira. *Amira User's Guide*. 2005.
- [37] Altair Hypermesh. *Altair HyperMesh Documentation*. 2007.
- [38] ABAQUS Inc. *ABAQUS Version 6.7 Documentation*. Troy, NY., 2008.
- [39] N. Chennimalai Kumar, J. A. Dantzig, I. M. Jasiuk, A. G. Robling, and C. H. Turner. Numerical modeling of long bone adaptation due to mechanical loading: Correlation with experiments. *Annals of Biomedical Engineering*, 38(3):594–604, 2010.
- [40] D. E. Discher, P. Janmey, and Y. Wang. Tissue cells feel and respond to the stiffness of their substrate. *Science*, 310(5751):1139–1143, 2005.
- [41] F. M. Pavalko, N. X. Chen, C. H. Turner, D. B. Burr, S. Atkinson, Y. F. Hsieh, J. Qiu, and R. L. Duncan. Fluid shear-induced mechanical signaling in mc3t3-e1 osteoblasts requires cytoskeleton-integrin interactions. *American Journal of Physiology - Cell Physiology*, 275(644-646), 1998.
- [42] S. Weinbaum, S. C. Cowin, and Yu Zeng. A model for the excitation of osteocytes by mechanical loading-induced bone fluid shear stresses. *Journal of Biomechanics*, 27(3):339–360, 1994.

- [43] Y. Han, S. C. Cowin, B. M. Schaffler, and S. Weinbaum. Mechanotransduction and strain amplification in osteocyte cell processes. *Proceedings of National Academy of Sciences*, 101:16689–16694, 2004.
- [44] S. Srinivasan, D. A. Weimer, S. C. Agans, S. D. Bain, and T. S. Gross. Low-magnitude mechanical loading becomes osteogenic when rest is inserted between each load cycle. *Journal of Bone and Mineral Research*, 17(9):1613–1620, 2002.
- [45] C. H. Turner, M. R. Forwood, and M. W. Otter. Mechanotransduction in bone: Do bone cells act as sensors of fluid flow? *FASEB Journal*, 8(11):875–878, 1994.
- [46] C. Rubin, S. Judex, and M. Hadjiargyrou. Skeletal adaptation to mechanical stimuli in the absence of formation or resorption of bone. *Journal of Musculoskeletal Neuronal Interactions*, 2(3):264–267, 2002.
- [47] Y. Qin, A. Saldanha, and T. Kaplan. Oscillatory bone fluid flow and its role in initiating remodeling in the absence of matrix strain. In *2001 ASME International Mechanical Engineering Congress and Exposition, Nov 11-16 2001*, volume 51, pages 51–52, New York, NY, United States, 2001.
- [48] Yi-Xian Qin, Marilyn Cute, and Clinton Rubin. Bone morphological adaptation induced by dynamic fluid flow in the absence of matrix strain. *Annals of Biomedical Engineering*, 28(SUPPL 1):1–8, 2000.
- [49] Y. X Qin, K. J. McLeod, and C. T. Rubin. Intramedullary pressure induced fluid flow in bone. In *Proceedings of the First Joint BMES/EMBS Conference, 13-16 Oct. 1999*, volume vol.1, page 490 vol.1, Atlanta, GA, USA, 1999. Program in Biomed. Eng., State Univ. of New York, Stony Brook, NY, USA, IEEE.
- [50] M. L. Knothe Tate and U. Knothe. An ex vivo model to study transport processes and fluid flow in loaded bone. *Journal of Biomechanics*, 33(2):247–254, 2000.
- [51] M. L. Knothe Tate, R. Steck, M. R. Forwood, and P. Niederer. In vivo demonstration of load-induced fluid flow in the rat tibia and its potential implications for processes associated with functional adaptation. *Journal of Experimental Biology*, 203(18):2737–2745, 2000.
- [52] K. M. Reich, C. V. Gay, and J. A. Frangos. Fluid shear stress as a mediator of osteoblast cyclic adenosine monophosphate production. *Journal of Cellular Physiology*, 143(1):100–104, 1990.
- [53] S. R. Pollack, N. Petrov, and R. Salzstein. An anatomical model for streaming potentials in osteons. *Journal of Biomechanics*, 17(8):627–636, 1984.
- [54] R. A. Salzstein and S. R. Pollack. Electromechanical potentials in cortical bone: Ii. experimental analysis. *Journal of Biomechanics*, 20(3):271–280, 1987.

- [55] A. G. Robling, P. J. Niziolek, L. A. Baldrige, K. W. Condon, M. R. Allen, I. Alam, S. M. Mantila, J. Gluhak-Heinrich, T. M. Bellido, S. E. Harris, and C. H. Turner. Mechanical stimulation of bone in vivo reduces osteocyte expression of *sost/sclerostin*. *Journal of Biological Chemistry*, 283(9):5866–5875, 2008.
- [56] D. Zhang, S. Weinbaum, and S. C. Cowin. Electrical signal transmission in a bone cell network: The influence of a discrete gap junction. *Annals of Biomedical Engineering*, 26(4):644–659, 1998.
- [57] D. Zhang, S. Weinbaum, and S. C. Cowin. Estimates of the peak pressures in bone pore water. *Journal of Biomechanical Engineering*, 120(6):697–703, 1998.
- [58] M. L. Knothe Tate. Interstitial fluid flow. *Bone Biomechanics Handbook*, 22:1–29, 2001.
- [59] Lidan You, Stephen C. Cowin, Mitchell B. Schaffler, and Sheldon Weinbaum. A model for strain amplification in the actin cytoskeleton of osteocytes due to fluid drag on pericellular matrix. *Journal of Biomechanics*, 34(11):1375–1386, 2001.
- [60] Dajun Zhang and Stephen C. Cowin. Oscillatory bending of a poroelastic beam. *Journal of the Mechanics and Physics of Solids*, 42(10):1575–1599, 1994.
- [61] Susannah P. Fritton and Sheldon Weinbaum. Fluid and solute transport in bone: Flow-induced mechanotransduction. *Annual Review of Fluid Mechanics*, 41(1):347–374, 2009.
- [62] D. Lacroix and P. J. Prendergast. A mechano-regulation model for tissue differentiation during fracture healing: Analysis of gap size and loading. *Journal of Biomechanics*, 35(9):1163–1171, 2002.
- [63] P. J. Prendergast, R. Huiskes, and K. Soballe. Biophysical stimuli on cells during tissue differentiation at implant interfaces. *Journal of Biomechanics*, 30(6):539–548, 1997.
- [64] M. A. Biot. General theory of three-dimensional consolidation. *Journal of Applied Physics*, 12:155–164, 1941.
- [65] M. A. Biot and D. G. Willis. The elastic coefficients of the theory of consolidation. *Journal of Applied Mechanics*, 24:594–601, 1957.
- [66] Olivier Coussy. *Mechanics of porous continua*. John Wiley, England, 1995.
- [67] E. Detournay and A. H. D Cheng. Fundamentals of poroelasticity. *Comprehensive Rock Engineering: Principles, Practice and Projects*, 2:113–171, 1993.
- [68] Stephen C. Cowin. Bone poroelasticity. *Journal of Biomechanics*, 32(3):217–238, 1999.
- [69] R. Steck, P. Niederer, and M. L. Knothe Tate. A finite element analysis for the prediction of load-induced fluid flow and mechanochemical transduction in bone. *Journal of Theoretical Biology*, 220(2):249–259, 2003.

Vita

Natarajan Chennimalai Kumar was born in the city of Madurai, India. He completed his undergraduate degree in Aeronautical Engineering at Madras Institute of Technology in Chennai, India in April 2003. He moved to Urbana to pursue his Masters in Aerospace Engineering at University of Illinois at Urbana-Champaign in Spring 2004. He graduated with Masters and started PhD studies at Department of Mechanical Science and Engineering from Fall of 2006. He has authored two journal publications, and has presented at around ten conferences during his graduate study. Natarajan will be moving on to Niskayuna, New York to start his job as Mechanical Engineer at General Electric Global Research Center after the completion of his PhD.