5-2018

Mathematical Model of Interstitial Fluid Flow and Characterization of the Lacunar Canalicular System in Cortical Bone

Melanie B. Venderley
Purdue University

MATHEMATICAL MODEL OF INTERSTITIAL FLUID FLOW AND CHARACTERIZATION OF THE LACUNAR CANALICULAR SYSTEM IN CORTICAL BONE

A Thesis

Submitted to the Faculty

of

Purdue University

by

Melanie B. Venderley

In Partial Fulfillment of the

Requirements for the Degree

of

Master of Science in Biomedical Engineering

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF THESIS APPROVAL

Dr. Eric A. Nauman, Co-Chair

School of Mechanical Engineering

Dr. Russell P. Main, Co-Chair

Department of Basic Medical Sciences

Dr. Sarah Calve

School of Biomedical Engineering

Approved by:

Dr. George R. Wodicka

Head of the School Graduate Program

To my parents Joe and Sherry, my brother Tim, and my sisters Anne and Samantha for their unwavering support and encouragement.

ACKNOWLEDGMENTS

I would like to thank all those who had an impact in the development of this thesis and aided me in its completion. In particular, I would like to express my gratitude to Dr. Nauman for his mentorship, guidance, and support during my undergraduate and graduate years and for always recounting the best stories. I would also like to thank Dr. Main for his instruction and aid and Dr. Calve for always inspiring me to be a better researcher.

Additionally, I would like to acknowledge the graduate and undergraduate students in the HIRRT and PMBAM Labs. Specifically, I would not have been able to complete this work without the guidance and expertise from Roy Lycke, Xiaoyu Xu, and Taylor Lee.

Finally, I would like to express the utmost gratitude to the Weldon School of Biomedical Engineering for providing me the opportunity to work as a Graduate Teaching Assistant and to the Graduate School for awarding me the Charles C. Chappelle Fellowship, both of which provided tuition and funding. Without either of them, I would have not have had the opportunity to accomplish this work.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

SYMBOLS

A	constant of integration
a	radius of osteocyte process
B	body containing all points within mixture
$\underline{B}^s$	Left Cauchy-Green deformation tensor for the solid constituent
b	radius of canaliculus
$\underline{b}^\alpha$	body force(s) acting on constituent α
C	constant of integration
$\underline{C}$	Right Cauchy-Green deformation tensor
c^α	interconversion of mass of constituent α
Dh	diameter of horizontal fibers in the glycocalyx structure
Dv	diameter of vertical fibers in the glycocalyx structure
$\underline{D}^f$	time rate of change of the deformation tensor for the fluid constituent
E	elastic modulus
I_1	first invariant of the Right Cauchy-Green tensor, $\underline{\mathrm{C}}$
I_2	second invariant of the Right Cauchy-Green tensor, $\underline{\mathrm{C}}$
$\underline{I}$	identity tensor
J^α	determinant of deformation gradient for constituent α
$\underline{K}_o$	reference configuration
$\underline{K}_t$	current configuration
k_o	magnitude of the permeability tensor $\underline{k}$
$\underline{k}$	permeability tensor
Lh	length of horizontal fibers in the glycocalyx structure
Lv	length of vertical fibers in the glycocalyx structure

M^α	total mass of constituent α
N	Nauman number
n	total number of constituents within mixture
n_c	total number of input terms in Cotter's Method
$\underline{n}$	normal vector
P_o	reference configuration region that contains B
P_t	current configuration region that contains B
$\underline{P}$	First Piola-Kirchhoff stress tensor
p	pressure
R_o	region within reference configuration, containing S
R_t	region within current configuration, containing S
S	subset of points of B
$\underline{S}$	Second Piola-Kirchhoff stress tensor
$\underline{T}^\alpha$	stress tensor for constituent α
$\underline{t}^\alpha$	traction vector acting on constituent α
v_o	magnitude of initial velocity
$\underline{v}^\alpha$	velocity of constituent α
W	Mooney-Rivlin strain energy function
$\underline{X}$	location within reference configuration
$\underline{x}$	location within current configuration
α	constituent within mixture
β_o	coefficient in the constitutive equation for $\underline{T}^s$
β_1	coefficient in the constitutive equation for $\underline{T}^s$
β_2	coefficient in the constitutive equation for $\underline{T}^s$
ϵ	strain
η	Poisson's ratio
μ	viscosity
ρ^α	apparent density of constituent α

$\underline{\pi}^{\alpha}$	internal (within RVE) force acting on constituent α due to interaction with other constituents
ρ_T^{α}	true density of constituent α
ϕ^{α}	volume fraction of constituent α
χ^{α}	mapping of constituent α from one configuration to another
∂P_o	surface of region P_o
∂P_t	surface of region P_t
∂R_o	surface of region R_o
∂R_t	surface of region R_t

ABBREVIATIONS

AFM	atomic force microscopy
CFD	computational fluid dynamics
CLSM	confocal laser scanning microscopy
CT	computed tomography
EtOH	ethanol
FIB-SEM	focused ion beam scanning electron microscopy
FITC	fluorescein isothiocyanate
LCS	lacunar canalicular system
NBF	neutral buffered formalin
NO	nitric oxide
PG	prostaglandin
ROI	region of interest
RVE	representative volume element
SR-PNT	synchrotron X-ray phase nano-tomography
SR-μCT	synchrotron radiation micro-computed tomography
UHVEM	ultra high voltage electron microscopy

ABSTRACT

Venderley, Melanie B. M.S.B.M.E., Purdue University, May 2018. Mathematical Model of Interstitial Fluid Flow and Characterization of the Lacunar Canalicular System in Cortical Bone. Major Professors: Eric A. Nauman, Russell P. Main.

While the mechanotransductive capability of skeletal tissue has been acknowledged for decades, the exact mechanisms that enable bone to sense and respond to external stimuli have remained elusive. Numerous theories have evolved to explain this behavior, most notably those involving fluid movement through the tissue's hierarchical structure. Within mineralized bone, osteocytes reside in micro and nanoporosities, known as lacunae and canaliculi, which house the cell body and their long cellular processes, respectively. Through this lacunar-canalicular system (LCS), osteocytes form an interconnected network, which allow signaling and communication with surrounding osteocytes via gap junctions and secreted factors. It has been theorized that external loading-induced interstitial fluid movement along the cell processes results in shear stresses and/or drag forces that elicit stimulatory responses from osteocytes. While length and mineralized tissue render direct measurements inaccessible, mathematical and computational modeling have been utilized to predict these potential stimulatory mechanisms. However, assumptions regarding the presence of a glycocalyx, which is a pericellular matrix within the interstitial fluid space, are typically made despite the inability to fully characterize its structure. Thus, to investigate the importance of the possible compositions of this glycocalyx, a mathematical model of interstitial fluid flow within a canaliculus was developed, utilizing mixture theory. Resulting sensitivity analyses show that assumptions regarding the glycocalyx greatly influence the profile within the LCS, therefore affecting potential mechanotransductive signals. Additionally, confocal microscopy and a custom,

automated reconstruction algorithm, were used to generate three-dimensional renderings of confocal images to further characterize the LCS and improve computational models. Both the mathematical model and reconstruction of the LCS will enhance the development of accurate predictive models and increase understanding of bone's mechanotransductive abilities.

1. INTRODUCTION

1.1 Motivation

In the 1860s, the concept that bone responds to mechanical stimuli was introduced by Dr. Julius Wolff, and since then, the theory has become widely accepted. However, the exact mechanisms by which cells within the tissue sense and transmit these signals have yet to be fully understood [1] [2] [3]. Osteocytes, mature bone cells, are hypothesized to be the main mechanosensors of bone due to their interconnected network within the lacunar-canalicular porosity throughout bone. This irregular nanoporosity contains interstitial fluid that may be the medium by which the processes on the osteocytes are able to sense external loads [4] [5] [6] [7]. The most popular theories of mechanotransduction in bone are the following: 1) fluid shear stress that the osteocyte cell process experiences due to loading-induced flow [2] [3] [8] [9] [10] and/or 2) fluid drag that acts on the cytoskeleton of the ostecyte cell process, where the fluid drag is a resultant of the transmembrane connections to an interstitial glycoaylx [9] [11] [12] [13] [14].

Directly measuring fluid shear stress and drag, both in vitro and in vivo, is currently not an option due to the micro- and nanoscopic scales of the lacunar-canalicular system (LCS). Thus, computational and mathematical models have been developed in order to predict these variables on the surface of the osteocyte cell process. In particular, numerous theoretical, poroelastic models have been used to study the interaction between the solid matrix of mineralized bone and the interstitial fluid flow. However, the structure of the glycocalyx has typically been assumed to exist for these models, even though it has yet to be fully characterized [5] [8] [9] [10] [11] [15] [16] [17] [18]. In addition, idealized structures for the lacunar-canalicular system are commonly used [11] [15]; while a few models have been developed to show the role of the LCS's

complex geometry on strain and stress amplification, they utilize a minimal sample size due to the manual labor needed to create numerous anatomically-correct LCS models [6] [19].

Thus, it is important to understand the impact of the glycocalyx and its effects on interstitial fluid flow. Furthermore, a need exists to develop a three-dimensional reconstruction method for the LCS that is not subject to user bias through manual segmentation processes. Characterizing the LCS will enable more accurate theoretical and computational models to understand and predict bone mechanotransduction. Determining the mechanisms by which bone remodels would ultimately be beneficial for the development of clinical interventions to prevent age-related bone loss.

1.2 Objective

The first objective of this thesis was to mathematically model interstitial fluid flow within the lacunar-canalicular system of cortical bone using mixture theory. The goal of the mathematical model was to determine the impact of the glycocalyx on the flow profile within the interstitial space when an idealized canaliculus is assumed. Additionally, the second objective was to characterize the three-dimensional lacunar-canalicular structure by generating an autonomous method to reconstruct two-dimensional images obtained from confocal laser scanning microscopy.

2. BACKGROUND

2.1 Hierarchical Structure of Bone

Bone has a hierarchical structure that is optimized to support external loads and protect internal structures while also remaining as light as possible. Its organization can be compartmentalized into five different levels: (1) the molecular structure, (2) the sub-nanostructure, (3) the nanostructure, (4) the sub-microstructure and the microstructure, and (5) the macrostructure [20], as shown in Figure 2.1 [21]. As a whole, bone is composed of approximately 65% mineral, 25% organic matrix (of which 90% is Type I collagen and the remainder is noncollagenous proteins), and 10% water [4]. Part of the organic matrix, the collagen molecules form triple helices, which assemble in groups of five to create fibrils. Within the gaps between the collagen molecules in these individual fibrils, nucleation of hydroxyapatite crystals occurs, which is the mineral constituent of bone. Aggregation of the fibrils leads to the formation of a single collagen fiber [21] [4].

At the sub-microscopic level, parallel sheets of Type I collagen are organized, known as lamellae, which are each $3 - 7\mu m$ wide [20]. As shown in Figure 2.1, they concentrically wrap around a central canal, referred to as a Haversian canal. Each canal is approximately $40 - 80\mu m$ in diameter and houses vasculature and innervation [22] [23] [24] [25]. The layering of the lamellae around the Haversian canal forms the basic structural unit of bone, the osteon. The diameter of osteons has been reported to range from $100\mu m$ to $500\mu m$ [20] [22] [23] [24] [26].

At the macroscopic level, the osteons comprise two types of bone, cortical (or compact) or cancellous (or trabecular/spongy). In long bones, cortical bone forms a dense, cortical shell that encompasses an interior, marrow canal [20]. Porous cancellous bone is internally found at the ends to aid in transmitting mechanical loads to

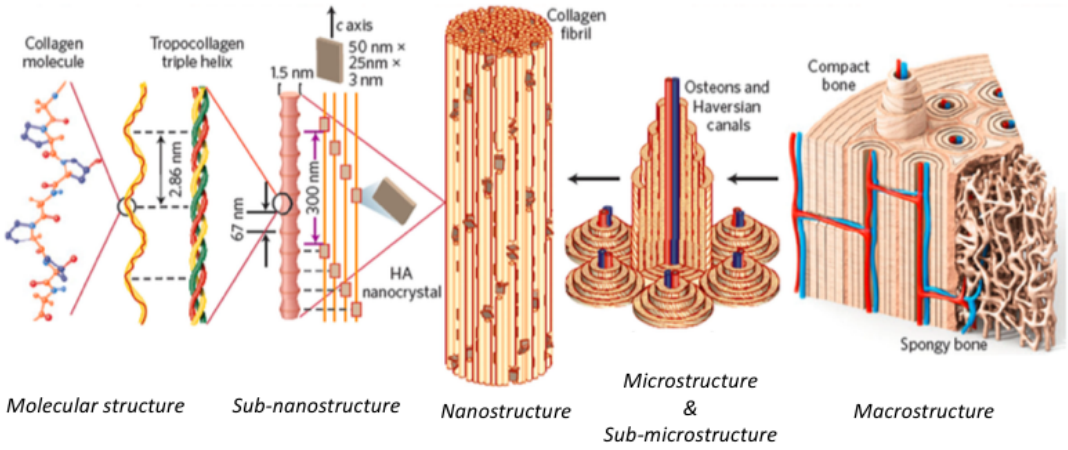


Fig. 2.1. Hierarchical structure of bone. The molecular structure consists of the collagen molecules, which form in groups of 5 to create fibrils (sub-nanostructure). Hydroxyapatite, or the bone mineral, nucleates in the gap regions between the ends of collagen molecules within the fibril. Aggregation of fibrils leads to collagen fibers, which make up the nanostructure. Parallel sheets of the collagen fibers form lamellae which create circumferential bands of bone, known as osteons (sub-microstructure and microstructure). A Haversian canal runs down the center of the osteon to provide an opening for vasculature and innervation. The osteons are then arranged to macroscopically form dense cortical bone or porous (spongy) trabeculae [4]. Image modified from Liu et al. [21].

the cortical bone. It is characterized by a rod and plate structure [4]. On the other hand, flat bones consist of two outer cortical layers with a thin, internal cancellous structure for support [20].

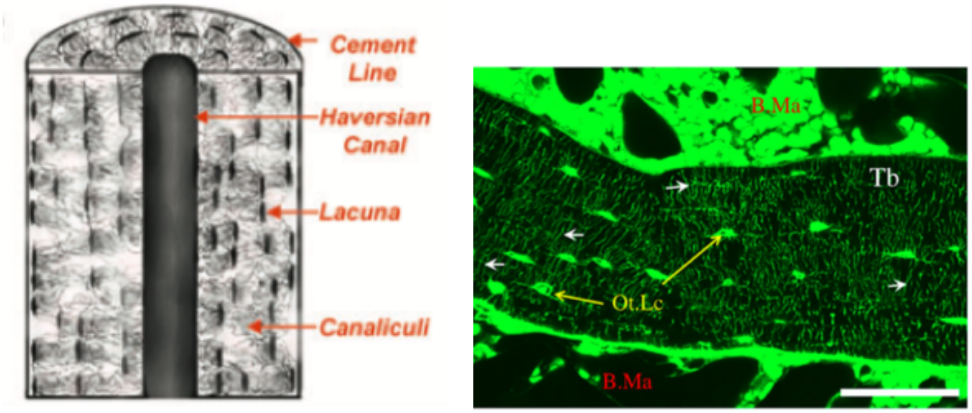
2.2 Lacunar Canalicular System

Bone is a dynamic tissue that can respond to external and internal signals to model and remodel in order to maintain its structural integrity. Three types of cells are responsible for bone maintenance: osteoclasts, osteoblasts, and osteocytes. Osteoclasts are multinuclear cells that manage bone resorption. They are essential

for removing bone for bone modeling during growth and also for bone remodeling due to microdamage. Once osteoclasts are no longer needed, they undergo apoptosis [4]. Osteoblasts, found on the surface of bone, synthesize and secrete unmineralized bone matrix, or osteoid, which they then mineralize. While 60% to 80% of osteoblasts undergo apoptosis following mineralization, the remainder either flatten and become lining cells or are entrapped within the matrix that they are actively producing [4]. A balance between the osteoclastic and osteoblastic activity is needed to ensure bone's overall structure.

The entrapped osteoblasts differentiate into the third cell type, osteocytes. Their cell bodies are contained within cavities known as lacunae while their long, dendritic processes are encompassed in thin cylindrical porosities known as canaliculi. With these processes, the osteocytes form an interconnected network, or the lacunar-canalicular system (LCS), as a means to communicate with surrounding cells. Figure 2.2(a) shows a longitudinally-sectioned osteon, illustrating the complex network that the osteocytes form within the mineralized bone [27]. Figure 2.2(b) is a trabecular section at the metaphysis in a rat tibia. Fluorescein isothiocyanate (FITC) is used to fill and visualize the LCS.

The porosity due to the LCS ranges over multiple length scales. Atomic force microscopy (AFM), confocal laser scanning microscopy (CLSM), synchrotron radiation micro-computed tomography (SR-μCT), focused ion beam scanning electron microscopy (FIB-SEM), and studies utilizing tracers, such as ferritin and horseradish peroxidase, have all been employed in order to characterize this network [29] [1] [2]. However, specimens utilized, sample location, sample preparation, and imaging technique are just a few of the differences across studies that have resulted in wide range of measured variables. The length of lacunae has been reported to range from $8 - 70\mu m$ [5] [30] [27] [31] [32], while its width ranges from $2 - 8\mu m$ [27] [32] [33]. Canaliculi are much smaller than the lacunae, with a diameter ranging from $150 - 844nm$ [27] [34] [33] [35] [36] [37] and between 41 and 115 canaliculi per lacuna [33] [37] [35] [38] [39].

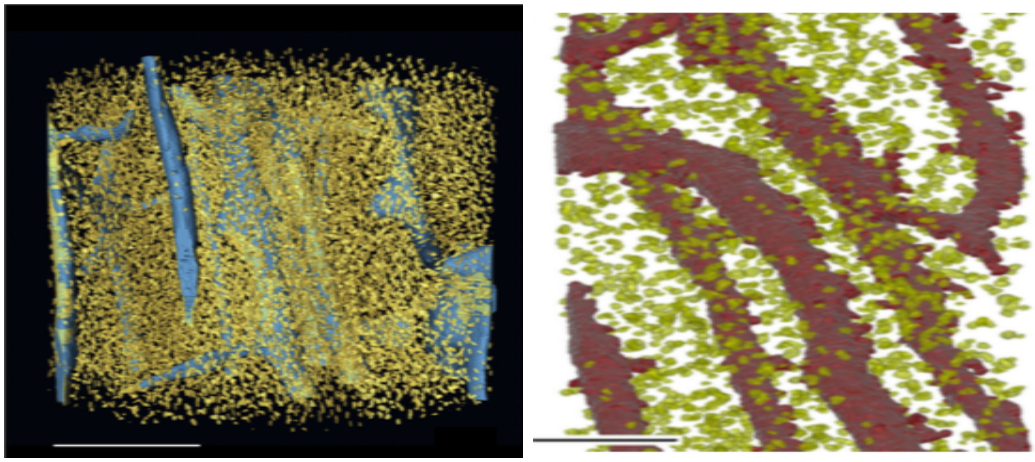


(a) (b)

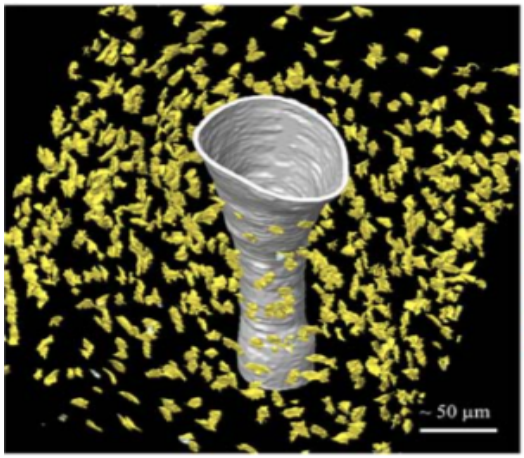
Fig. 2.2. (a) A longitudinally-sectioned osteon illustrating the networking of the lacunar-canalicular system [27]. (b) Cancellous rat tibia metaphysis stained using fluorescein isothiocyanate (FITC). Bone marrow (B.Ma), osteocyte lacunae (Ot.Lc), and trabecula (Tb) are indicated. The white arrows point to the canalicular porosity; scale bar = $130\mu m$ [28].

While many of these measurements are made from two-dimensional images, an increasing number of methods have been developed to acquire three-dimensional representations of the network. In 2004, McCreadie et al. reconstructed confocal stacks of trabecular bone from the iliac crest of women with and without osteoporotic fracture. Assuming an ellipsoidal shape for the lacunae, the volume and shape of the lacunae were determined using the principle moments of intertia and the eigenvalues of the intertia matrix [40]. Similarly, Carter et al., Palacio-Mancheno et al., and Dong et al. performed analyses using either SR-μCT or high resolution computed tomography (CT) to characterize lacunar morphology, as shown in Figure 2.3 [30] [41] [42]. More recently, Heveran et al. published a novel open-source tool to reconstruct three-dimensional lacunar morphology from CLSM images, using similar mathematical constructs as McCreadie et al. Their tool eliminates user bias and error when manually segmenting, in addition to greatly reducing the amount of time to reconstruct the confocal z-stacks. Utilizing this automatic segmentation tool, confocal images were reconstructed from cortical bone samples of adult (6-mo) and aged (24-mo) male mice, which had been stained with 1% basic fuchsin. Figure 2.4 shows the comparison between the reconstructed lacunae from the adult (a) and aged (b) mice. They ultimately were able to conclude significant age-related changes to the three-dimensional morphology, with the lacunae from the aged mice being smaller, more spherical, more oblate, and less densely populated within the mineralized bone [43].

Many additional articles have been published, which not only reconstruct the lacunar geometry, but also the much more complex canalicular structure. In 2004, Sugawara et al. stained sixteen-day-old chick calvariae with phalloidin and imaged the osteocytes using CLSM. Multiple softwares were employed in order to reconstruct and characterize the LCS. IMPARIS software was utilized to reconstruct the network for osteocyte cell body measurements (Figure 2.5(a)), NEURON TRACER software was used to trace and determine the length of the osteocyte cell processes, and SURPASS software was needed to determine surface area and volume of the osteocyte cell body and processes, combined [38]. In 2011, Schneider et al. used FIB-SEM on sections

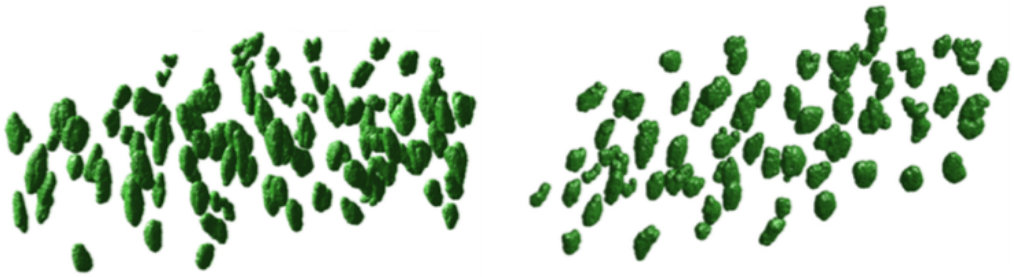


(a) (b)



(c)

Fig. 2.3. (a) Using SkyScan CT software, SR-μCT images from a human male anterior cortical bone sample were reconstructed. Lacunae are shown in gold within the region of interest (ROI) with the vascular canals in blue; scale bar=300μm [30]. (b) Reconstruction of the anterior proximal tibia of a skeletally-mature rat, using a 1-μm-resolution CT scan and Skyscan software. Vasculature is shown in red while the lacunae are in gold; scale bar=100μm [41]. (c) Three-dimensional reconstruction of a human female cortical osteon from the femoral mid-diphysis, using SR-μCT and a custom, automated method to segment and extract lacunar morphology. Haversian canal is shown in white with the surrounding lacunae in gold [42].

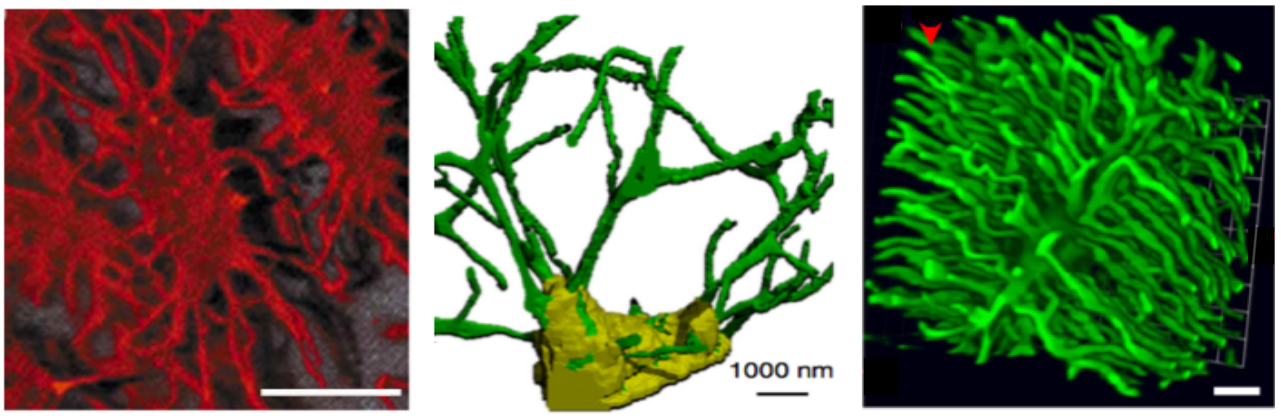


(a) (b)

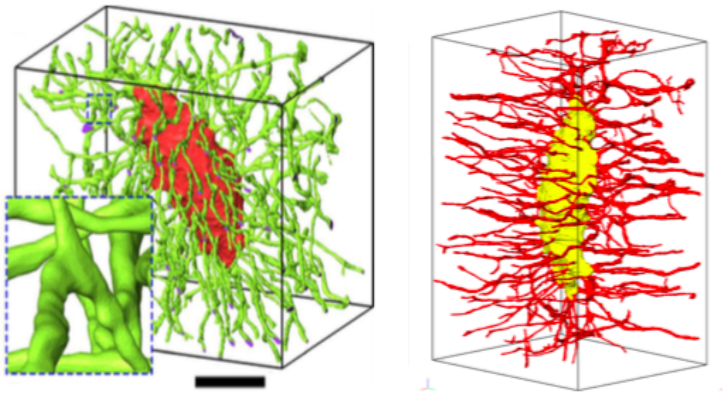
Fig. 2.4. Three-dimensional reconstructions of cortical bone samples from (a) 6-month-old and (b) 24-month-old male mice [43].

from the mid-diaphysis of a murine femur. Post-processing and reconstruction were performed using both ImageJ and IPL software (Figure 2.5(b)). However, as this was just a proof of concept study, only one sample and a limited ROI were analyzed [35]. Sharma et al. provided detailed reconstruction of single lacunae and their connecting canaliculi. Tibial, transverse cortical sections were obtained from 26-week-old female rats, stained with FITC, and imaged using CLSM. For three-dimensional analysis, ImageJ was used to quantify osteocyte lacunar density, Volocity was implemented to determine the number of canaliculi per lacuna (Figure 2.5(c)), and Mimics 3D reconstruction software's segmentation tools were used to isolate lacunar and canalicular volumes [37]. Most recently, in 2015, both Hesse et al. and Varga et al. both utilized synchrotron X-ray phase nano-tomography (SR-PNT) to characterize the LCS. Hesse et al. obtained images from human mandible sections, manually selected ROI without any microdamage, and reconstructed them using Matlab (Figure 2.5(d)) [44]. Human femur cortical bone samples were used by Varga et al. and reconstructed in Matlab using a custom-developed connectivity enhancement algorithm (Figure 2.5(e)) [5].

Unfortunately, many of the current reconstruction techniques characterizing both the lacunar and canalicular morphology involve manual work through segmentation [35] [37] [38], multiple softwares [35] [37] [38], or techniques, such as FIB-SEM and SR-PNT, that are not as accessible as CLSM [5] [35] [44]. Hesse et al. and Varga et al. produce promising results with their custom-implemented Matlab codes [5] [44]; however, creation of a similar code may prove difficult and time-consuming for researchers that could benefit from it. Thus, an open-source code reconstructing the LCS, as Heveran et al. provided for the lacunae, would be beneficial for further research, removing the variable of post-processing from the differences that are seen in characterization of the LCS.



(a) (b) (c)



(d) (e)

Fig. 2.5. (a) Using IMARIS software, reconstructed osteocytes from a 16-day-old chick calvariae, imaged using CLSM; scale bar=10μm [38]. (b) Three-dimensional reconstruction of FIB-SEM images from the mid-diaphysis of a 12-week-old murine femur. Both ImageJ and IPL software were utilized for reconstruction and to obtain LCS morphological parameters. The lacuna is shown in yellow, and its canaliculi are in green [35]. (c) Volocity software was utilized to reconstruct lacunae and canaliculi from the tibial cortex of 26-week-old female rat; scale bar=5μm [37].(d) Using Matlab, human mandible sections were reconstructed from images obtained with SR-PNT. Lacuna is shown in orange, and its connecting canaliculi are in green; scale bar=10μm [44].(e) Human femoral cortical bone samples reconstructed using a custom-developed connectivity enhancement algorithm. Lacuna is shown in yellow, and the canaliculi are in red [5].

2.3 Mechanosensing and Predictive Models

Bone has been shown to sense and respond to mechanical loading through modeling and remodeling of both the trabecular and cortical surfaces. Therefore, it has the ability to alter its external shape and internal structure in order to sustain the macroscopic loading placed upon it. This concept that bone can adapt in response to mechanical stress was first suggested by Dr. Julius Wolff in the 1860s. Termed "Wolff's Law," it states that the process of bone remodeling follows mathematical laws in response to changes in shape or stress on bone [4] [45]. Since then, the processes by which bones can sense and adapt to external stimuli have been investigated.

In 1977, Piekarski and Munro proposed that fluid flow within the LCS of bone is critical for nutrient delivery, waste removal, and mechanotransduction, which has since become widely accepted [46] [47]. It has been shown that not only does mechanical loading induce fluid convection in bone [46], but also that oscillatory fluid flow due to intramedullary pressure in the absence of mechanical deformation is strongly correlated with bone formation [48]. Because of this, osteocytes are considered to be the mechanosensors in bone as they form an interconnected network throughout the tissue, able to sense and signal to other areas [4]. In addition, osteocytes are surrounded by interstitial fluid flow within the LCS; the irregular nano-porosity of the LCS allows for fluid stress concentrations around the cell processes, which could potentially be a means of mechanotransduction [5] [6] [7].

However, the exact mechanism(s) by which osteocytes sense the mechanical stimuli is still under debate. As shown in Figure 2.6, recent reviews in the field of bone mechanotransduction point out that a definite method has yet to be proven, given the difficulty of performing and measuring in vivo experimental studies within the mineralized bone matrix [1] [2] [3]. A number of theories have arisen over the years to explain the modeling and remodeling behavior of bone in response to external loading. Direct cell strain due to deformation of the bone matrix [8] [9], piezoelectricity due to the direct piezoelectric effect [2] [49] [50] [51], hydrostatic pressure [2] [8] [9], streaming

potentials [3] [8] [52], fluid shear stress [2] [3] [8] [9] [10], drag forces and hoop strains due to an interstitial glycocalyx [9] [11] [12] [13] [14], and chemotransport [2] have all been theorized to transmit external mechanical loadings, therefore regulating bone modeling and remodeling. The interconnectedness of the macroscale loading to each of these mechanisms is outlined in Figure 2.7. Macroscopic loading on the bone causes torsion, bending, and/or compression, resulting in localized areas of bone matrix deformation. Known as the direct piezoelectric effect, collagen and apatite molecules rearrange due to this deformation, which can lead to charged surfaces within the LCS and potentially influence cellular activity [2] [49] [50] [51]. In addition, the localized deformation can cause direct strain on the osteocytes themselves [8] [9]. Hydrostatic pressure can also develop within the LCS due to the impermeability of the bone matrix [2] [8] [9], which can result in interstitial fluid flow. Streaming potentials [3] [8] [52] and chemotransport [2] can occur due to the movement of charged particles within the flow. Fluid shear stress can develop on the osteocyte cell process as a result of this fluid flow [2] [3] [8] [9] [10]. Last, amplified hoop strains and drag forces on the osteocyte's internal cytockeleton have been theorized due to the presence of a glycocalyx (composed of proteoglycans, glycoproteins, and hyaluronic acid) and transmembrane integrin connections [9] [11] [12] [13] [14].

The piezoelectricity of bone as a possible mechanotransducive means has diminished over the years in favor of direct strain and fluid-related mechanisms, such as shear stress and streaming potentials [51]. By recording whole-bone strain histories from a variety of species, it has been shown that peak bone strains due to activities of daily living result in strains around 2000 $\mu\epsilon$ [53] [54] [55]. However, Smalt et al. measured the release of nitric oxide (NO) and prostaglandin (PG) of mechanically stimulated in vitro cells, which are early in vivo responses to mechanical stimulation. A stimulus of 5000 $\mu\epsilon$, though, did not elicit an increased response of either NO or PGE(2), even though Fritton et al. reported much lower in vivo strains. Thus, this led to a paradox that the tissue and cellular levels of deformation are not equivalent. Recently, a poromicromechanic model was developed, revealing that tissue level phys-

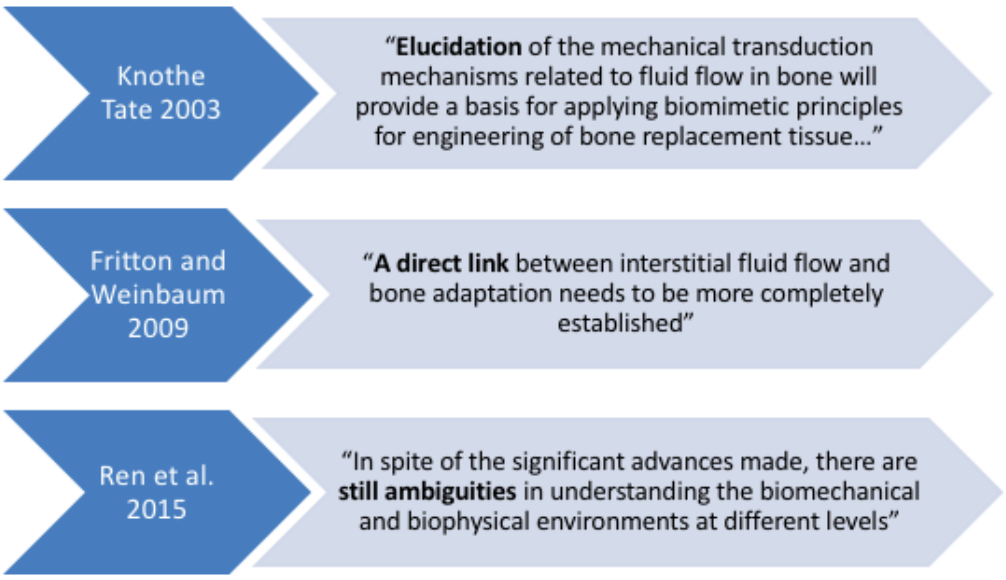


Fig. 2.6. Recent reviews on osteocyte cell mechanobiology and bone mechanotransduction. Each publication states that the exact mechanisms behind bone's remodeling capabilites have yet to be fully understood [1] [2] [3].

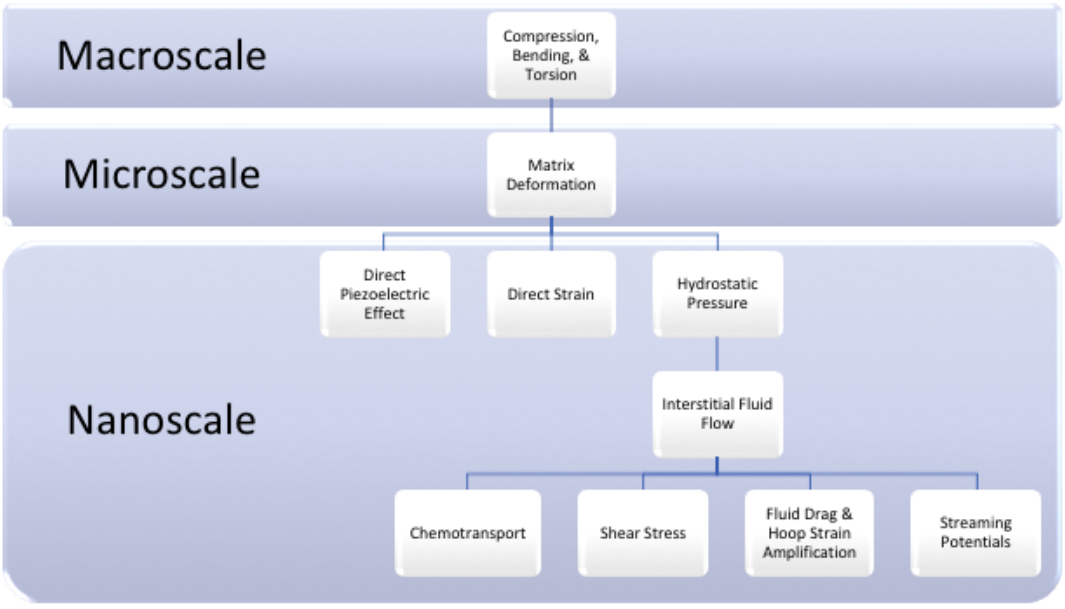


Fig. 2.7. Connection of macroscopic loading to the theories of bone mechanotransduction. Compression, bending, and torsion of the whole bone at the macroscopic level leads to bone matrix deformation in localized areas. Due to this deformation, collagen and apatite molecular rearrangement can lead to charged surfaces known as the direct piezoelectric effect [2] [49] [50] [51] in addition to direct strain placed on the osteocytes themselves [8] [9]. In addition, hydrostatic pressure develops within the LCS due to the impermeable bone matrix, resulting in interstitial fluid flow [2] [8] [9]. Such fluid flow can induce shear stress on the osteocyte cell process [2] [3] [8] [9] [10], amplified hoop strains and drag forces due to transmembrane integrin connections with the internal cytoskeleton and the external glycocalyx [9] [11] [12] [13] [14], and streaming potentials [3] [8] [52] and chemotransport due to the movement of charged nutrients [2].

iological bone strains can induce osteocyte-stimulating lacunar pressure [56]. Thus, the interstitial fluid flow due to this hydrostatic pressure and its potential effects have been more thoroughly studied, with the main focus on fluid-induced shear stress and fluid drag and hoop strain amplification due to integrin-mediated cytoskeletal and glycocalyx interactions.

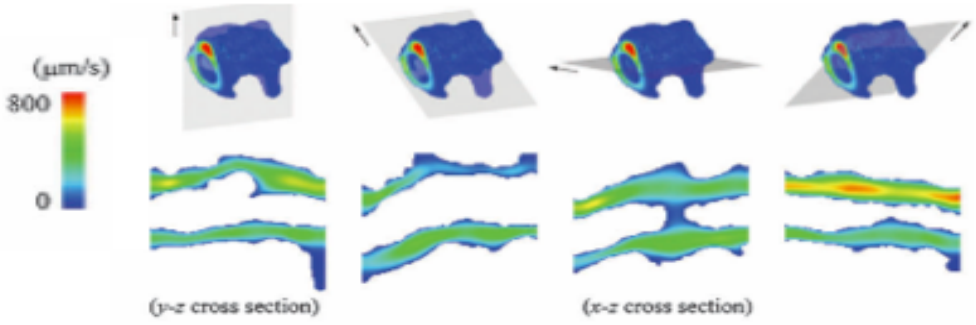
As direct in vivo measurements of shear stress and drag forces on the osteocyte cell processes are currently not possible, numerous mathematical and computational models have been developed to predict these measures. In order to model bone at the microscopic and nanoscopic scales, the theory of poroelasticity is commonly implemented, where the interaction of fluid-flow within a fluid-saturated porous medium due to deformation of the solid is studied [25]. Three main approaches have been developed in order to implement poroelastic models: 1) effective medium approach, 2) mixture theory approach, and 3) homogenization. While at the continuum point all of the approaches result in the same set of equations, the three approaches differ in the way that they average the system, taking into consideration all of the different constituents. In general, the effective medium approach is useful in determining solid component parameters, mixture theory allows for the averaging over different fluid phases, and homogenization provides wave propagation parameters. While effective medium approach has been more commonly utilized, mixture theory is beginning to be implemented as it can effectively model multiphase tissues [57] [58].

To implement the fundamental equations for poroelastic theory, a representative volume element (RVE) is developed, which is a continuum element containing all of the constituents in the model (i.e. fluid(s) and solid). A variety of RVEs for bone are used within the literature - either a full osteon, multiple lacunae and their canaliculi, a single lacunae and its canaliculi, or just a single canaliculus - which are then analytically or computationally modeled. In addition, the geometry is either idealized with the lacunae being approximated as an ellipsoid and the canaliculi as long tubes or three-dimensional reconstructions of the actual geometry are utilized with computational software. Last, depending on the assumptions made, the outputs

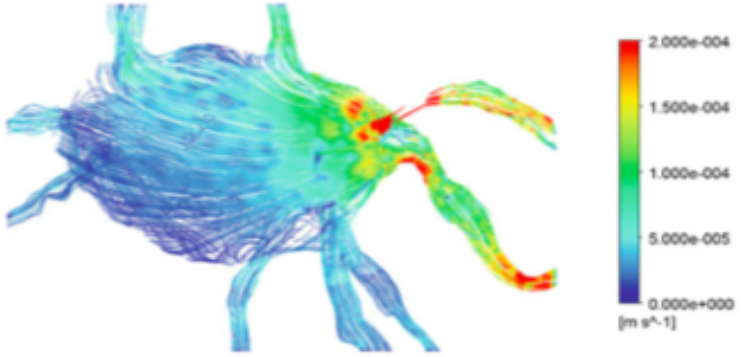
from each model differ, being either fluid shear stress, fluid strain, fluid velocity, pore pressure, or a combination of these measures.

In regards to mathematical models, You et al. proposed a poroelastic model to model strain within an idealized canaliculus with the assumption of tethering units in the interstitial space, which was updated in 2008 by Wang et al. [11] [15]. While popular, other poroelastic models have also been developed of idealized osteons and lacunae to model fluid shear stress, fluid velocity, and pore pressure [16] [59]. In addition, computational models have become widely popular. While idealized models have been utilized using these softwares [7] [17] [60], the ability to import anatomically-correct RVEs of the LCS has become extremely useful. In 2012, Kamioka et al. utilized ultra high voltage electron microscopy (UHVEM) and computational fluid dynamics (CFD) to model fluid velocities within human canaliculi [6]. Verbruggen et al. also implemented CFD to determine fluid velocity, shear stress, and strain within reconstructed CLSM lacunae and canaliculi [19]. Figure 2.8 shows results from both of these CFD simulations.

While these predictive models provide promising results, the majority of them are only valid under the assumption that a glycocalyx, and potentially tethering elements, within the interstitial space are present [5] [8] [9] [10] [11] [15] [16] [17] [18]. Studies utilizing tracers, such as microperoxidase, horseradish peroxidase, reactive red, and ferritin, have been used to indirectly determine the pore size or porosity of the glycocalyx [61]. In addition, protrusions from the canalicular wall have been observed, suggesting that the osteocyte cell process may connect to those points; however, direct attachments to these "hillocks" have yet to be shown [6] [15] [62]. Transmission electron microscopy (TEM) has also been used in order to identify tethering elements between the osteocyte process and the canalicular wall [36] [63]. However, the three-dimensional reconstruction of canalicular UHVEM images led Kamioka et al. to conclude that TEM produces images that provide a false connection between the canaliculus and the osteocyte cell process [6]. Using immunohistochemistry, McNamara et al. identified the presence of β_3 integrins along the length of the osteocyte cell



(a)



(b)

Fig. 2.8. Computational models of fluid flow within the LCS. (a) Utilizing three-dimensional canalicular UHVEM reconstructions, Kamioka et al. created CFD models to determine fluid velocities within human canaliculi [6]. (b) Verbruggen et al. used reconstructed CLSM models and CFD to quantify fluid velocity within lacunae and their surrounding canaliculi [19].

processes, possible attachment sites to a glycocalyx [62]. These integrins have shown to aid in mechanotransduction in in vitro osteocytes [64], but have yet to show any significant contribution to transmitting mechanical signals in vivo [9]. In addition, CD44, a transmembrane protein whose extracellular component has been shown to connect to hyaluronic acid (HA), has been identified on osteocyte cell processes [65].

While the presence of a glycocalyx provides a rationale for the strain paradox [36] [11] [13] [63], the actual structure of it has yet to be fully and directly identified or characterized. The parameters used for the glycocalyx and tethering elements within each of these models could greatly affect the predicted fluid shear stresses and drag forces, and thus confound which, if either, is the main mechanism of osteocyte mechanotransduction.

3. MATHEMATICAL MODEL

3.1 Motivation

The majority of poroelastic mathematical and computational models use the assumption that a glycocalyx exists within the interstitial space between osteocytes and the LCS [5] [8] [9] [10] [11] [15] [16] [17] [18]. However, the glycocalyx has yet to be fully identified or characterized due to limitations in imaging and the embedded nature of the LCS. Thus, the purpose of this chapter was to generate a poroelastic model of the LCS using mixture theory, as it is allows for expansion into multiphase fluids, and determine the relative importance of the glycocalyx and its structural parameters on velocity and shear stress outputs.

3.2 Theory

In order to model the behavior of interstitial fluid within cortical bone, a continuum mixture theory model was developed. Figure 3.1 illustrates the deformation that a mixture, represented as B, undergoes from the reference configuration to the current configuration. The location of a constituent in one configuration can be directly mapped to the other configuration, shown as $\chi^{\alpha}(\underline{x}^{\alpha}, t)$, where α represents any constituent found within the mixture.

Each constituent is considered an individual continuum. The volume fraction for a mixture of n constituents is defined as follows,

$$\sum_{\alpha=1}^{n} \Phi^{\alpha} = 1 \tag{3.1}$$

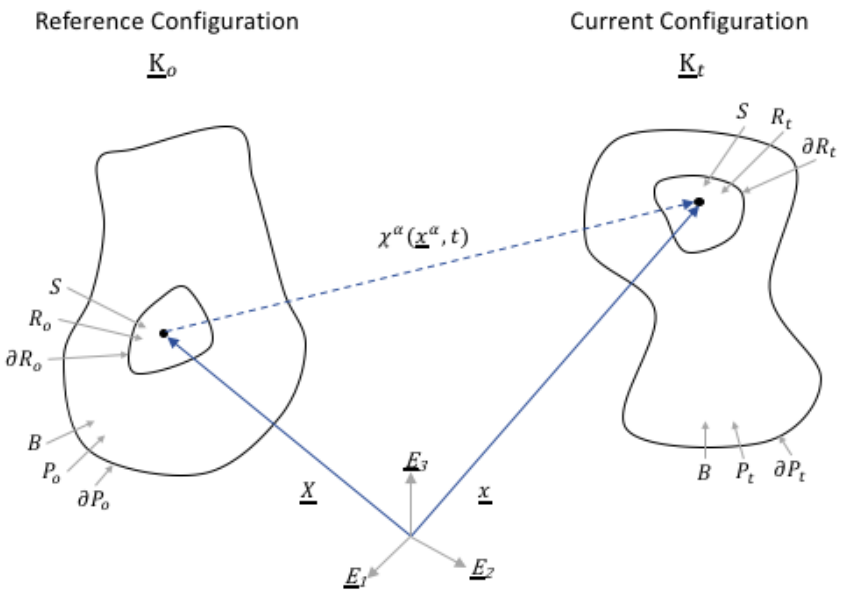


Fig. 3.1. Deformation of body B from the reference configuration to the current configuration. A point within the reference configuration can be directly mapped to the current configuration using χ^α.

3.2.1 Balance of Mass

In the current configuration and within body B that defines the whole mixture, region P_t is defined such that this mixture. The total mass within P_t for constituent α is

$$M^\alpha = \int_{P_t} \rho^\alpha dv + \int_{P_t} c^\alpha dv \tag{3.2}$$

where ρ^α is the apparent density and c^α is the interconversion of mass of constituent α. The apparent density of constituent α is defined as

$$\rho^\alpha = \rho_T^\alpha \phi^\alpha \tag{3.3}$$

where ρ_T^α is the true density of constituent α and ϕ^α is its respective volume fraction. The interconversion of mass within region P_t must follow

$$\sum_{\alpha=1}^{n} c^\alpha = 0 \tag{3.4}$$

While the location of P_t can change with time, the mass of the region remains constant,

$$\frac{D^\alpha M^\alpha}{Dt} = \frac{D^\alpha}{Dt}\int_{P_t} \rho^\alpha dv + \frac{D^\alpha}{Dt}\int_{P_t} c^\alpha dv = 0 \tag{3.5}$$

$$\frac{D^\alpha}{Dt}\int_{P_t} \rho^\alpha dv = \int_{P_t} c^\alpha dv \tag{3.6}$$

To map the left side of Equation (3.6) back to the reference configuration, the relationship $dv = JdV$ is needed, where J is the determinant of the deformation matrix. Thus, mapping back to the reference configuration results in,

$$\frac{D^\alpha}{Dt}\int_{P_o} \rho^\alpha J^\alpha dV = \int_{P_t} c^\alpha dv \tag{3.7}$$

Taking the derivative of the left side of Equation (3.7),

$$\int_{P_o} \Big(\frac{D^\alpha \rho^\alpha}{Dt} J^\alpha + \rho^\alpha \frac{D^\alpha J^\alpha}{Dt}\Big) dV = \int_{P_t} c^\alpha dv \tag{3.8}$$

$$\int_{P_o} \Big(\frac{D^\alpha \rho^\alpha}{Dt} + \rho^\alpha div(\underline{v}^\alpha)\Big) J^\alpha dV = \int_{P_t} c^\alpha dv \tag{3.9}$$

Mapping region P_o to the current configuration leads to,

$$\int_{P_t} \Big(\frac{D^\alpha \rho^\alpha}{Dt} + \rho^\alpha div(\underline{v}^\alpha) - c^\alpha \Big) dv = 0 \tag{3.10}$$

Since the integral in Equation (3.10) is equal to zero, this implies that the integrand is also zero. This results in the final form of the balance of mass for constituent α:

$$\frac{D^\alpha \rho^\alpha}{Dt} + \rho^\alpha div(\underline{v}^\alpha) = c^\alpha \tag{3.11}$$

3.2.2 Balance of Linear Momentum

As in the balance of mass in Section 3.2.1, region P_t is defined such that it contains all constituents of the mixture in the current configuration. The balance of linear momentum can be expressed as

$$\frac{D^\alpha}{Dt} \int_{P_t} \rho^\alpha \underline{v}^\alpha dv = \int_{\partial P_t} \underline{t}^\alpha da + \int_{P_t} (\rho^\alpha \underline{b}^\alpha + c^\alpha \underline{v}^\alpha + \underline{\pi}^\alpha) dv \tag{3.12}$$

where $\underline{v}^\alpha$ is the velocity of constituent α, $\underline{t}^\alpha$ is the traction vector acting on the surface ∂P_t of region P_t, $\rho^\alpha \underline{b}^\alpha$ represents the body force(s) acting on P_t, and $\underline{\pi}^\alpha$ is the interaction force of constituent α with surrounding constituents.

Mapping region P_t back to the reference configuration on the left side of Equation (3.12) produces the following:

$$\frac{D^\alpha}{Dt} \int_{P_t} \rho^\alpha \underline{v}^\alpha dv = \frac{D^\alpha}{Dt} \int_{P_o} \rho^\alpha \underline{v}^\alpha J^\alpha dV \tag{3.13}$$

$$= \int_{P_o} \Big(\dot{\rho}^\alpha \underline{v}^\alpha J^\alpha + \rho^\alpha \dot{\underline{v}}^\alpha J^\alpha + \rho^\alpha \underline{v}^\alpha J^\alpha div(\underline{v}^\alpha) \Big) dV \tag{3.14}$$

$$= \int_{P_t} \Big(\rho^\alpha \dot{\underline{v}}^\alpha + \underline{v}^\alpha \Big(\dot{\rho}^\alpha + \rho^\alpha div(\underline{v}^\alpha) \Big) \Big) dv \tag{3.15}$$

$$= \int_{P_t} \rho^\alpha \dot{\underline{v}}^\alpha dv \tag{3.16}$$

where $\dot{\rho}^\alpha + \rho^\alpha div(\underline{v}^\alpha) = 0$ due to the balance of mass.

Using the divergence theorem, the traction vector in Equation (3.12) can be expressed as the stress tensor of constituent α,

$$\int_{\partial P_t} \underline{t}^{\alpha} da = \int_{\partial P_t} (\underline{\underline{T}}^{\alpha} \cdot \underline{n}) da \tag{3.17}$$

$$= \int_{P_t} div(\underline{\underline{T}}^{\alpha}) dv \tag{3.18}$$

The balance equation can then be written as

$$\int_{P_t} \rho^{\alpha} \underline{\dot{v}}^{\alpha} dv = \int_{P_t} \Big(div(\underline{\underline{T}}^{\alpha}) + \rho^{\alpha} \underline{b}^{\alpha} + c^{\alpha} \underline{v}^{\alpha} + \underline{\pi}^{\alpha} \Big) dv \tag{3.19}$$

As with the balance of mass, the balance of linear momentum must also equate to zero. This, again, implies that the integrands within the integrals must also be equal to zero. Also, the assumption will be made that the interconversion term is also zero, as none of the constituents are being converted to one another. Thus, the final form of the balance of linear momentum is as follows,

$$\rho^{\alpha} \frac{D^{\alpha} \underline{v}^{\alpha}}{Dt} = div(\underline{\underline{T}}^{\alpha}) + \rho^{\alpha} \underline{b}^{\alpha} + \underline{\pi}^{\alpha} \tag{3.20}$$

3.2.3 Balance of Angular Momentum

The initial form of the balance of angular momentum for constituent α can be expressed as

$$\frac{D^{\alpha}}{Dt} \int_{P_t} (\underline{x} \times \rho^{\alpha} \underline{v}^{\alpha}) dv = \int_{\partial P_t} (\underline{x} \times \underline{t}^{\alpha}) da + \int_{P_t} \Big((\underline{x} \times \rho^{\alpha} \underline{b}^{\alpha}) + (\underline{x} \times \underline{\pi}^{\alpha}) + (\underline{x} \times c^{\alpha} \underline{v}^{\alpha}) + \underline{M}^{\alpha} \Big) dv \tag{3.21}$$

Mapping the left side of Equation (3.21) back to the reference configuration results in

$$\begin{aligned} \frac{D^{\alpha}}{Dt} \int_{P_t} (\underline{x} \times \rho^{\alpha} \underline{v}^{\alpha}) dv &= \frac{D^{\alpha}}{Dt} \int_{P_o} (\underline{x} \times \rho^{\alpha} \underline{v}^{\alpha}) J^{\alpha} dV \\ &= \int_{P_o} \Bigg((\underline{v}^{\alpha} \times \rho^{\alpha} \underline{v}^{\alpha} J^{\alpha}) + (\underline{x} \times \frac{D^{\alpha} \rho^{\alpha}}{Dt} \underline{v}^{\alpha} J^{\alpha}) + (\underline{x} \times \rho^{\alpha} \frac{D^{\alpha} \underline{v}^{\alpha}}{Dt} J^{\alpha}) \\ &\quad + \Big(\underline{x} \times \rho^{\alpha} \underline{v}^{\alpha} J^{\alpha} div(\underline{v}^{\alpha}) \Big) \Bigg) dV \end{aligned} \tag{3.22}$$

The term $\underline{v}^\alpha \times \rho^\alpha \underline{v}^\alpha J^\alpha$ is equal to zero as it is the velocity vector, $\underline{v}^\alpha$, crossed with itself. Without this term and rearranging Equation (3.22), the left side of Equation (3.21) becomes

$$\frac{D^\alpha}{Dt}\int_{P_t}(\underline{x}\times\rho^\alpha\underline{v}^\alpha)dv = \int_{P_o}\left[\left(\underline{x}\times\rho^\alpha\frac{D^\alpha\underline{v}^\alpha}{Dt}J^\alpha\right)+\left(\underline{x}\times\left(\frac{D^\alpha\rho^\alpha}{Dt}+\rho^\alpha div(\underline{v}^\alpha)\right)\underline{v}^\alpha J^\alpha\right)\right]dV \tag{3.23}$$

Because the term $\frac{D^\alpha\rho^\alpha}{Dt}+\rho^\alpha div(\underline{v}^\alpha)$ is the balance of mass and thus equal to zero, the result can be further simplified to

$$\frac{D^\alpha}{Dt}\int_{P_t}(\underline{x}\times\rho^\alpha\underline{v}^\alpha)dv = \int_{P_o}\left(\underline{x}\times\rho^\alpha\frac{D^\alpha\underline{v}^\alpha}{Dt}J^\alpha\right)dV \tag{3.24}$$

$$= \int_{P_t}\left(\underline{x}\times\rho^\alpha\frac{D^\alpha\underline{v}^\alpha}{Dt}\right)dv \tag{3.25}$$

Next, the traction vector on the surface ∂P_t in Equation (3.21) can be rewritten in component form as follows,

$$\int_{\partial P_t}(\underline{x}\times\underline{t}^\alpha)da = \int_{\partial P_t}(\underline{x}\times\underline{T}^\alpha\underline{n})da \tag{3.26}$$

$$= \int_{\partial P_t}\varepsilon_{ijk}x_i(T^\alpha_{jb})\underline{E}_k \tag{3.27}$$

where ε_{ijk} is the alternating tensor.

The divergence theorem is now used to express the traction vector on the surface ∂P_t as the stress tensor of constituent α within region P_t

$$\int_{\partial P_t}(\underline{x}\times\underline{t}^\alpha)da = \int_{P_t}\frac{\partial}{\partial x_b}\left(\varepsilon_{ijk}x_iT^\alpha_{jb}\underline{E}_k\right)dv \tag{3.28}$$

$$= \int_{P_t}\varepsilon_{ijk}\left(\frac{\partial x_i}{\partial x_b}T^\alpha_{jb}+x_i\frac{\partial T^\alpha_{jb}}{\partial x_b}\right)\underline{E}_k dv \tag{3.29}$$

$$= \int_{P_t}\varepsilon_{ijk}\left(T^\alpha_{ji}+x_i\frac{\partial T^\alpha_{jb}}{\partial x_b}\right)\underline{E}_k dv \tag{3.30}$$

$$= \int_{P_t}\left(\varepsilon_{ijk}T^\alpha_{ji}\underline{E}_k+\varepsilon_{ijk}x_i\frac{\partial T^\alpha_{jb}}{\partial x_b}\underline{E}_k\right)dv \tag{3.31}$$

Equation (3.21) can now be rewritten as

$$\int_{P_t}\left(\underline{x}\times\rho^\alpha\frac{D^\alpha\underline{v}^\alpha}{Dt}\right)dv = \int_{P_t}\Big(\left((\underline{T}^\alpha)^T-\underline{T}^\alpha\right)+\left(\underline{x}\times div(\underline{T}^\alpha)\right) + (\underline{x}\times\rho^\alpha\underline{b}^\alpha)+(\underline{x}\times\underline{\pi}^\alpha)+(\underline{x}\times c^\alpha\underline{v}^\alpha)+\underline{M}^\alpha\Big)dv \tag{3.32}$$

Rearranging Equation (3.32) produces

$$\int_{P_t} \left(\underline{x} \times \left(\rho^\alpha \frac{D^\alpha \underline{v}^\alpha}{Dt} - div(\underline{T}^\alpha) - \rho^\alpha \underline{b}^\alpha - \underline{\pi}^\alpha - c^\alpha \underline{v}^\alpha \right) \right) dv = \int_{P_t} \left((\underline{T}^\alpha)^T - \underline{T}^\alpha + \underline{M}^\alpha \right) dv \quad (3.33)$$

From the balance of linear momentum in Section 3.2.2, it has been shown that

$$\rho^\alpha \frac{D^\alpha \underline{v}^\alpha}{Dt} - div(\underline{T}^\alpha) - \rho^\alpha \underline{b}^\alpha - \underline{\pi}^\alpha - c^\alpha \underline{v}^\alpha = \underline{0} \quad (3.34)$$

Therefore, Equation (3.33) can be reduced to

$$\underline{0} = \int_{P_t} \left((\underline{T}^\alpha)^T - \underline{T}^\alpha + \underline{M}^\alpha \right) dv \quad (3.35)$$

Since the integrand in Equation (3.35) is equal to zero, the final form of the balance of angular momentum is, thus,

$$\underline{T}^\alpha - (\underline{T}^\alpha)^T = \underline{M}^\alpha \quad (3.36)$$

3.3 Methods

3.3.1 Model Scales

In order to employ the balance laws presented in Section 3.2, different length scales were defined, as shown in Figure 3.2. These scales - nanoscale, mesoscale, and macroscale - were used to create representative volume elements (RVE) in which interstitial fluid flow could be analyzed.

3.3.2 Pressure-Driven Flow

Representative Volume Element

In which to model interstitial fluid flow due solely to pressure, Figure 3.3 shows the representative volume, defined as B, of the system. This RVE consists of a single, idealized canaliculus. The balance laws derived in Section 3.2 were then applied to this RVE.

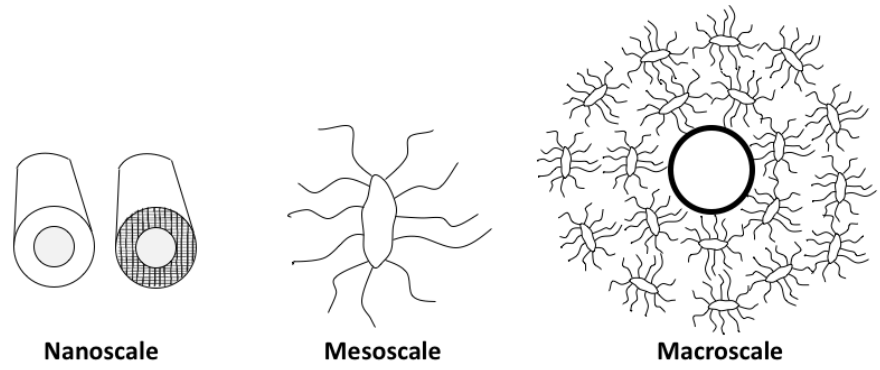


Fig. 3.2. The length scales defined for mathematical modeling. The nanoscale consists of a single, idealized canaliculus with (left) and without (right) a glycocalyx within the interstitial space between the cell process and canalicular wall. At the mesoscale, a single lacuna is present with its branching canaliculi. The macroscale consists of a single Haversian canal with its surrounding LCS.

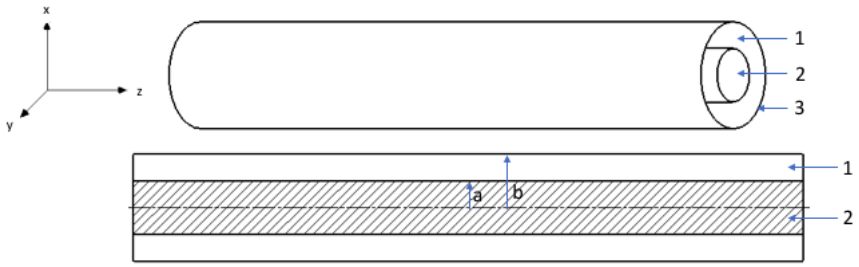


Fig. 3.3. Example of the representative body B containing the mixture: Idealized 3D-representation (top) of a canaliculus, which houses the osteocyte process, surrounded by cortical bone. A cross-section of the volume (bottom) is shown. The numbers indicate the following: (1) interstitial space, (2) osteocyte process, and (3) canalicular wall. The letters indicate: (a) osteocyte cell process radius and (b) canaliculus radius.

Nondimensionalization of Balance of Linear Momentum

The balance of linear momentum, Equation (3.20), was nondimensionalized for each constituent within the mixture in order to make it independent of the physical system and allow for influential terms to be identified.

To apply Equation (3.20) to each constituent, a constitutive equation is utilized to describe the interaction term $\underline{\pi}^{\alpha}$. Since this term describes internal forces, then the following condition must be met,

$$\sum_{\alpha=1}^{n} \underline{\pi}^{\alpha} = \underline{0} \tag{3.37}$$

The balance of linear momentum can then be rewritten for each constituent as,

$$\rho^s \frac{D^s \underline{v}^s}{Dt} = div(\underline{T}^s) + \rho^s \underline{b}^s + \mu \underline{k}^{-1}(\underline{v}^f - \underline{v}^s) \tag{3.38}$$

$$\rho^f \frac{D^f \underline{v}^f}{Dt} = div(\underline{T}^f) + \rho^f \underline{b}^f + \mu \underline{k}^{-1}(\underline{v}^s - \underline{v}^f) \tag{3.39}$$

where s and f indicate the solid and fluid constituents, respectively, μ is the viscosity of the fluid, and $\underline{k}$ is the permeability tensor of the RVE. $\underline{T}^s$ and $\underline{T}^f$ are represented by constitutive equations to describe the stress field for each constituent,

$$\underline{T}^s = -\phi^s p \underline{I} + \phi^s [\beta_o \underline{I} + \beta_1 \underline{B}^s + \beta_2 (\underline{B}^s)^2] \tag{3.40}$$

$$\underline{T}^f = -\phi^f p \underline{I} + 2\phi^f \mu \underline{D}^f \tag{3.41}$$

where p is the pressure shared by both constituents, $\underline{I}$ is the identity tensor, β_o is a coefficient representing the compressibility of the solid, $\underline{B}^s$ is the left Cauchy-Green deformation tensor, $\beta_1 \underline{B}^s + \beta_2 (\underline{B}^s)^2$ represents the nonlinearity of the stress-strain relationship of the solid, and $\underline{D}^f$ is the time rate of change of the fluid deformation tensor. Equation 3.40 assumes the solid is a hyperelastic material while equation 3.41 assumes that the interstitial fluid is a Newtonian incompressible fluid.

Thus, Equations 3.38 and 3.39 can be rewritten as follows,

$$\rho^s \frac{D^s \underline{v}^s}{Dt} = div\Big(-\phi^s p \underline{I} + \phi^s [\beta_o \underline{I} + \beta_1 \underline{B}^s + \beta_2 (\underline{B}^s)^2]\Big) + \rho^s \underline{b}^s + \mu \underline{k}^{-1}(\underline{v}^f - \underline{v}^s) \tag{3.42}$$

$$\rho^f \frac{D^f \underline{v}^f}{Dt} = div(-\phi^f p\underline{I} + 2\phi^f \mu \underline{D}^f) + \rho^f \underline{b}^f + \mu \underline{k}^{-1}(\underline{v}^s - \underline{v}^f) \tag{3.43}$$

In order to nondimensionalize Equations 3.42 and 3.43, three fundamental terms were chosen: mass, length, and time. Table 3.1 shows the RVE variables chosen to represent each fundamental term, where ρ_T^f is the true density of the fluid, v_o is the magnitude of the initial velocity of the fluid, and $\sqrt{k_o}$ is the square root of the magnitude of the permeability tensor $\underline{k}$.

Table 3.1.
Variables, for each fundamental term, utilized to nondimensionalize the fluid and solid balances of linear momentum.

Fundamental Term	**Variable**
Mass	$\rho_T^f k_o^{\frac{3}{2}}$
Length	$\sqrt{k_o}$
Time	$\frac{\sqrt{k_o}}{v_o}$

Each variable in Equations 3.42 and 3.43 can be expressed as a function of its dimensionless form (presented as the respective variable with an overbar) and the variables needed to nondimensionalize it. Table 3.2 shows the results of this nondimensionalization.

The solid equation (Equation 3.42) can be nondimensionalized using the forms in Table 3.2,

$$\frac{v_o^2 \rho_T^f}{\sqrt{k_o}}\left(\overline{\rho}^s \overline{\frac{D^s \underline{v}^s}{Dt}}\right) = -\frac{v_o^2 \rho_T^f}{\sqrt{k_o}}\left(\overline{\underline{\nabla}}(\phi^s \overline{p})\right) + \frac{1}{\sqrt{k_o}}\overline{div}\left(\phi^s[\beta_o \underline{I} + \beta_1 \underline{B}^s + \beta_2 (\underline{B}^s)^2]\right)$$
$$+\frac{v_o^2 \rho_T^f}{\sqrt{k_o}}\left(\overline{\rho}^s \overline{\underline{b}}^s\right) + \frac{\mu v_o \overline{\underline{k}}^{-1}}{k_o}\left(\overline{\underline{v}}^f - \overline{\underline{v}}^s\right) \tag{3.44}$$

where $div(-\phi^s p\underline{I}) = -\overline{\underline{\nabla}}(\phi^s \overline{p})$.

Table 3.2.
Balance of linear momentum terms and corresponding nondimensionalized forms using the fundamental variables.

Balance of Linear Momentum Term	**Nondimensionalized Form**
ρ^{α}	$\rho_T^f \overline{\rho}^{\alpha}$
$\underline{v}^{\alpha}$	$v_o \underline{\overline{v}}^{\alpha}$
p	$\rho_T^f v_o^2 \overline{p}$
$\underline{D}^f$	$\frac{v_o}{\sqrt{k_o}} \underline{\overline{D}}^f$
$\frac{\partial}{\partial \underline{x}}$	$\left(\frac{1}{\sqrt{k_o}}\right) \frac{\partial}{\partial \underline{\overline{x}}}$
$\frac{D^{\alpha} \underline{v}^{\alpha}}{Dt}$	$\left(\frac{v_o^2}{\sqrt{k_o}}\right) \overline{\frac{D^{\alpha} \underline{v}^{\alpha}}{Dt}}$
$\underline{k}$	$k_o \underline{\overline{k}}$

Multiplying Equation 3.44 through by $\frac{k_o}{\mu v_o}$ results in,

$$\frac{v_o \rho_T^f \sqrt{k_o}}{\mu} \left(\overline{\rho}^s \overline{\frac{D^s \underline{v}^s}{Dt}} \right) = -\frac{v_o \rho_T^f \sqrt{k_o}}{\mu} \left(\overline{\underline{\nabla}} (\phi^s \overline{p}) \right) + \overline{div} \left(\phi^s \left[\frac{\beta_0 \sqrt{k_o}}{\mu v_o} \underline{I} + \frac{\beta_1 \sqrt{k_o}}{\mu v_o} \underline{B}^s + \frac{\beta_2 \sqrt{k_o}}{\mu v_o} (\underline{B}^s)^2 \right] \right) + \frac{v_o \rho_T^f \sqrt{k_o}}{\mu} \left(\overline{\rho}^s \underline{\overline{b}}^s \right) + \underline{\overline{k}}^{-1} (\underline{\overline{v}}^f - \underline{\overline{v}}^s) \tag{3.45}$$

Substituting in $N = \frac{v_o \rho_T^f \sqrt{k_o}}{\mu}$ results in the following nondimensionalized balance of linear momentum for the solid constituent,

$$N \left(\overline{\rho}^s \overline{\frac{D^s \underline{v}^s}{Dt}} \right) = -N \left(\overline{\underline{\nabla}} (\phi^s \overline{p}) \right) + \overline{div} \left(\phi^s \left[\frac{\beta_0 \sqrt{k_o}}{\mu v_o} \underline{I} + \frac{\beta_1 \sqrt{k_o}}{\mu v_o} \underline{B}^s + \frac{\beta_2 \sqrt{k_o}}{\mu v_o} (\underline{B}^s)^2 \right] \right) + N \left(\overline{\rho}^s \underline{\overline{b}}^s \right) + \underline{\overline{k}}^{-1} (\underline{\overline{v}}^f - \underline{\overline{v}}^s) \tag{3.46}$$

Likewise, the fluid equation (Equation 3.43) can be nondimensionalized using the nondimensionalized form of the terms in Table 3.2,

$$\frac{v_o^2 \rho_T^f}{\sqrt{k_o}} \left(\overline{\rho}^f \overline{\frac{D^f \underline{v}^f}{Dt}} \right) = -\frac{v_o^2 \rho_T^f}{\sqrt{k_o}} \left(\overline{\underline{\nabla}} (\phi^f \overline{p}) \right) + \frac{2 \mu v_o}{k_o} \overline{div} \left(\phi^f \underline{\overline{D}}^f \right) + \frac{v_o^2 \rho_T^f}{\sqrt{k_o}} \left(\overline{\rho}^f \underline{\overline{b}}^f \right) + \frac{\mu v_o \underline{\overline{k}}^{-1}}{k_o} \left(\underline{\overline{v}}^s - \underline{\overline{v}}^f \right) \tag{3.47}$$

where, again, $\overline{div}(-\phi^f p\underline{I}) = -\overline{\underline{\nabla}}(\phi^f \overline{p})$.

Multiplying Equation 3.47 through by $\frac{k_o}{\mu v_o}$ results in,

$$\frac{v_o \rho_T^f \sqrt{k_o}}{\mu}\left(\overline{\rho}^f \overline{\frac{D^f \underline{v}^f}{Dt}}\right) = -\frac{v_o \rho_T^f \sqrt{k_o}}{\mu}\left(\underline{\overline{\nabla}}(\phi^f \overline{p})\right) + 2\overline{div}\left(\phi^f \underline{\overline{D}}^f\right) + \frac{v_o \rho_T^f \sqrt{k_o}}{\mu}\left(\overline{\rho}^f \underline{\overline{b}}^f\right) + \underline{\overline{k}}^{-1}\left(\underline{\overline{v}}^s - \underline{\overline{v}}^f\right) \quad (3.48)$$

Substituting in $N = \frac{v_o \rho_T^f \sqrt{k_o}}{\mu}$ results in the following nondimensionalized balance of linear momentum for the fluid constituent,

$$N\left(\overline{\rho}^f \overline{\frac{D^f \underline{v}^f}{Dt}}\right) = -N\left(\underline{\overline{\nabla}}(\phi^f \overline{p})\right) + 2\overline{div}\left(\phi^f \underline{\overline{D}}^f\right) + N\left(\overline{\rho}^f \underline{\overline{b}}^f\right) + \underline{\overline{k}}^{-1}\left(\underline{\overline{v}}^s - \underline{\overline{v}}^f\right) \quad (3.49)$$

Velocity Profile within the RVE: Glycocaylx Assumed

Pressure-driven interstitial fluid flow with no mechanical stimulus, such as in a bioreactor, flows down the length of the canaliculus in the RVE. The solid constituent, which consists of the osteocyte cell process, is not physically moving. The term v^s and its derivatives in Equation 3.46 are therefore equal to zero. In addition, physical deformation of the solid is not occurring; thus, the portion of the stress tensor that represents solid deformation, $\phi^s[\beta_o \underline{I} + \beta_1 \underline{B}^s + \beta_2 (\underline{B}^s)^2]$, is also zero. The body force terms in Equations 3.46 and 3.49, $\overline{\rho}^s \underline{\overline{b}}^f$ and $\overline{\rho}^f \underline{\overline{b}}^f$, respectively, were assumed to be negligible. An assumption was also made that the flow is steady, i.e. no fluid acceleration. Thus, $\overline{\frac{D^f \underline{v}^f}{Dt}}$ in Equation 3.49 is neglected. The nondimensionalized forms of the fluid and solid constituents are then,

$$\underline{0} = -N\left(\underline{\overline{\nabla}}(\phi^s \overline{p})\right) + \underline{\overline{k}}^{-1} \underline{\overline{v}}^f \quad (3.50)$$

$$\underline{0} = -N\left(\underline{\overline{\nabla}}(\phi^f \overline{p})\right) + 2\overline{div}\left(\phi^f \underline{\overline{D}}^f\right) - \underline{\overline{k}}^{-1} \underline{\overline{v}}^f \quad (3.51)$$

To obtain the velocity profile within the RVE, the fluid balance of linear momentum (Equation 3.51) was utilized. Using the definition of the rate of deformation

tensor $\overline{\underline{D}}^f = \frac{1}{2}\Big(\overline{\frac{\partial v_i}{\partial x_j}} + \overline{\frac{\partial v_j}{\partial x_i}}\Big)$, the balance of linear momentum was rewritten in index notation as follows,

$$0 = -N\frac{\partial \phi^f}{\partial \overline{x}_i}\overline{p} - N\phi^f\overline{\frac{\partial p}{\partial x_i}} + 2\frac{\partial \phi^f}{\partial \overline{x}_j}\overline{D}^f_{ij} + \phi^f\frac{\partial}{\partial \overline{x}_j}\Big(\overline{\frac{\partial v_i^f}{\partial x_j}} + \overline{\frac{\partial v_j^f}{\partial x_i}}\Big) - \overline{k}^{-1}_{ij}\overline{v}^f_i \tag{3.52}$$

The RVE is considered homogeneous; thus, $\frac{\partial \phi^f}{\partial \overline{x}_i}$ and $\frac{\partial \phi^f}{\partial \overline{x}_j}$ are equal to zero, resulting in

$$0 = -N\phi^f\overline{\frac{\partial p}{\partial x_i}} + \phi^f\frac{\partial}{\partial \overline{x}_j}\Big(\overline{\frac{\partial v_i^f}{\partial x_j}} + \overline{\frac{\partial v_j^f}{\partial x_i}}\Big) - \overline{k}^{-1}_{ij}\overline{v}^f_i \tag{3.53}$$

The balance of mass in Section 3.2.1 resulted in Equation 3.11, which can be used to further simplify Equation 3.53. For this system of pressure-driven flow, no interconversion of mass is occurring, and the fluid is assumed to be incompressible. The balance of mass (Equation 3.11) is then reduced to,

$$\rho^\alpha div(\underline{v}^\alpha) = 0 \tag{3.54}$$

Thus, $\overline{\frac{\partial v_i^f}{\partial x_i}}$ is equal to zero. Using this, Equation 3.53 is further simplified,

$$0 = -N\phi^f\frac{\partial \overline{p}}{\partial \overline{x}_i} + \phi^f\Big(\frac{\partial^2 \overline{v}_i^f}{\partial \overline{x}_j^2}\Big) - \overline{k}^{-1}_{ij}\overline{v}^f_i \tag{3.55}$$

Equation 3.55, written in tensor notation,

$$\phi^f\overline{\underline{\nabla}}^2\underline{\overline{v}}^f - \underline{\overline{k}}^{-1}\overline{v}^f = N\phi^f\overline{\underline{\nabla}}\overline{p} \tag{3.56}$$

is similar in form to the result that Weinbaum et al. developed [10]

This simplified balance of linear momentum was converted to a cylindrical coordinate system to correspond to the geometry of the RVE. In this case, the z-axis is along the longitudinal axis of the canaliculus. Using the result in Equation 3.55, the second term can be expanded out in cylindrical coordinates as follows,

$$\phi^f\Big(\overline{\frac{\partial^2 v_i^f}{\partial x_j^2}}\Big) = \phi^f\Big[\frac{1}{\overline{r}}\frac{\partial}{\partial \overline{r}}\Big(\overline{r}\overline{\frac{\partial v_z^f}{\partial r}}\Big) + \frac{1}{\overline{r}^2}\Big(\overline{\frac{\partial^2 v_z^f}{\partial \theta^2}}\Big) + \overline{\frac{\partial^2 v_z^f}{\partial z^2}}\Big] \tag{3.57}$$

$$= \frac{\phi^f}{\overline{r}}\overline{\frac{\partial v_z^f}{\partial r}} + \phi^f\overline{\frac{\partial^2 v_z^f}{\partial r^2}} \tag{3.58}$$

when assuming that the flow is steady and uniform.

Equation 3.56 can then be rewritten,

$$\phi^f \overline{\frac{\partial^2 v_z^f}{\partial r^2}} + \frac{\phi^f}{\overline{r}} \overline{\frac{\partial v_z^f}{\partial r}} - \overline{k}_{zr}^{-1} \overline{v}_z^f = N\phi^f \overline{\frac{\partial p}{\partial z}} \tag{3.59}$$

The differential equation in Equation 3.59 has the form of a modified Bessel equation. In order to solve for the velocity profile, a few manipulations were first made to obtain a solvable form. Equation 3.59 was multiplied by $\frac{\overline{r}^2}{\phi^f}$,

$$\overline{r}^2 \overline{\frac{\partial^2 v_z^f}{\partial r^2}} + \overline{r} \overline{\frac{\partial v_z^f}{\partial r}} - \overline{r}^2 \overline{k_v}^{-1} \overline{v}_z^f = N\overline{r}^2 \overline{\frac{\partial p}{\partial z}} \tag{3.60}$$

where $\overline{k}_v = \phi^f \overline{k}_{zr}$.

In addition, Equation 3.60 was put in terms of $\hat{\overline{r}}$, which is set equal to $\frac{\gamma \overline{r}}{\overline{b}}$. γ is $\frac{b}{\sqrt{k_v}}$, where b is the radius of the canaliculus.

Rewriting Equation 3.60 with this substitution leads to,

$$\frac{\overline{b}^2 \hat{\overline{r}}^2}{\gamma^2} \Big(\overline{\frac{\partial^2 v_z^f}{\partial \hat{r}}} \Big) \frac{\gamma^2}{\overline{b}^2} + \frac{\overline{b}\hat{\overline{r}}}{\gamma} \Big(\overline{\frac{\partial v_z^f}{\partial \hat{r}}} \Big) \frac{\gamma}{\overline{b}} - \frac{\overline{b}^2 \overline{k_v}^{-1} \hat{\overline{r}}^2}{\gamma^2} \overline{v}_z^f = \frac{N\overline{b}^2 \hat{\overline{r}}^2}{\gamma^2} \Big(\overline{\frac{\partial p}{\partial z}} \Big) \tag{3.61}$$

Simplifying Equation 3.61 results in the needed form of a modified Bessel function,

$$\hat{\overline{r}}^2 \overline{\frac{\partial^2 v_z^f}{\partial \hat{r}^2}} + \hat{\overline{r}} \overline{\frac{\partial\ v_z^f}{\partial \hat{r}}} - \hat{\overline{r}}^2 \overline{v}_z^f = \frac{N\overline{b}^2 \hat{\overline{r}}^2}{\gamma^2} \Big(\overline{\frac{\partial p}{\partial z}} \Big) \tag{3.62}$$

The homogeneous solution to the differential equation in Equation 3.62 has the following form,

$$\overline{v}_z^f = A\overline{I}_o(\hat{\overline{r}}) + C\overline{K}_o(\hat{\overline{r}}) \tag{3.63}$$

$$= A\overline{I}_o\Big(\frac{\gamma\overline{r}}{\overline{b}}\Big) + C\overline{K}_o\Big(\frac{\gamma\overline{r}}{\overline{b}}\Big) \tag{3.64}$$

where A and C are constants of integration and $\overline{I}_o$ and $\overline{K}_o$ are modified Bessel functions of the zeroth order.

To find the solution to the nonhomogeneous equation, the particular solution was inserted into Equation 3.62. This particular solution is the following,

$$\overline{v}_p^f = D\hat{\overline{r}}^2 + E\hat{\overline{r}} + F \tag{3.65}$$

where D, E, and F are constants.

Differentiating and plugging in Equation 3.62 results in,

$$(2D)\hat{\bar{r}}^2 + (2D\hat{\bar{r}} + E)\hat{\bar{r}} - (D\hat{\bar{r}}^2 + E\hat{\bar{r}} + F)\hat{\bar{r}}^2 = \frac{N\bar{b}^2\hat{\bar{r}}^2}{\gamma^2}\Big(\overline{\frac{\partial p}{\partial z}}\Big) \tag{3.66}$$

From Equation 3.66, constants D and E are equal to 0, and constant F is $-N\bar{k}_v\overline{\frac{\partial p}{\partial z}}$. Thus, the particular solution to Equation 3.62 is,

$$\bar{v}_p^f = -N\bar{k}_v\overline{\frac{\partial p}{\partial z}} \tag{3.67}$$

The generalized solution to Equation 3.62 is the linear combination of the homogeneous and nonhomogeneous solutions,

$$\bar{v}_z^f = A\bar{I}_o\Big(\frac{\gamma\bar{r}}{\bar{b}}\Big) + C\overline{K}_o\Big(\frac{\gamma\bar{r}}{\bar{b}}\Big) - N\bar{k}_v\overline{\frac{\partial p}{\partial z}} \tag{3.68}$$

To solve for constants A and B, the boundary conditions must be applied. A no-slip boundary condition is assumed at both the osteocyte cell process (radius = a) and the wall of the canaliculus (radius = b). Thus, $\bar{v}_z^f = 0$ at $\bar{r} = \bar{a}$ and $\bar{r} = \bar{b}$. The finalized form for the velocity profile of pressure-driven interstitial fluid within a canaliculus is,

$$\bar{v}_z^f = N\bar{k}_v\overline{\frac{\partial p}{\partial z}}\Big[\overline{A}_1\bar{I}_o\Big(\frac{\gamma\bar{r}}{\bar{b}}\Big) + \overline{C}_1\overline{K}_o\Big(\frac{\gamma\bar{r}}{\bar{b}}\Big) - 1\Big] \tag{3.69}$$

where

$$\overline{A}_1 = \frac{\overline{K}_o\Big(\gamma\Big) - \overline{K}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big)}{\overline{I}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big)\overline{K}_o\Big(\gamma\Big) - \overline{I}_o\Big(\gamma\Big)\overline{K}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big)} \tag{3.70}$$

$$\overline{C}_1 = \frac{\overline{I}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big) - \overline{I}_o\Big(\gamma\Big)}{\overline{I}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big)\overline{K}_o\Big(\gamma\Big) - \overline{I}_o\Big(\gamma\Big)\overline{K}_o\Big(\frac{\gamma\bar{a}}{\bar{b}}\Big)} \tag{3.71}$$

Velocity Profile within the RVE: No Glycocaylx Assumed

When a glycocaylx is not assumed to exist, the flow through an annulus (with an internal cylinder) can be modeled using the Hagen-Poiseuille equation, assuming

incompressible, steady-state flow. The differential equation representing this flow is as follows,

$$-\frac{1}{\mu}\frac{\partial p}{\partial z} = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial v_z^f}{\partial r}\right) \tag{3.72}$$

As done previously for the balance of linear momentum, the three fundamental terms of mass, length, and time were used to nondimensionalize Equation 3.72. Table 3.3 shows the RVE variables chosen to represent each fundamental term, where ρ_T^f is the true density of the fluid, v_o is the magnitude of the initial velocity of the fluid, and a is the radius of the osteocyte cell process. Each variable in Equation 3.72 can be expressed as a function of its dimensionless form (presented as the respective variable with an overbar) and the variables used to nondimensionalize it. Table 3.4 shows the results of this nondimensionalization.

Table 3.3.
Variables, for each fundamental term, utilized to nondimensionalize the Hagen-Poiseuille type flow within the RVE when a glycocalyx is not assumed.

Fundamental Term	**Variable**
Mass	$\rho_T^f a^3$
Length	a
Time	$\frac{a}{v_o}$

Table 3.4.
Hagen-Poiseuille flow terms and their corresponding nondimensionalized forms using the fundamental quantity variables.

Hagen-Poiseuille Flow Term	**Nondimensionalized Form**
$\frac{\partial p}{\partial z}$	$\left(\frac{\rho_T^f v_o^2}{a}\right)\overline{\frac{\partial p}{\partial z}}$
$\frac{\partial}{\partial r}$	$\left(\frac{1}{a}\right)\overline{\frac{\partial}{\partial r}}$
r	$a\overline{r}$
$\frac{\partial v_z^f}{\partial r}$	$\left(\frac{v_o}{a}\right)\overline{\frac{\partial v_z}{\partial r}}$

The nondimensionalized form of Equation 3.72 is thus the following,

$$-\frac{\rho_T^f v_o^2}{a\mu}\left(\overline{\frac{\partial p}{\partial z}}\right) = \frac{1}{a^2\overline{r}}\overline{\frac{\partial}{\partial r}}\left(v_o\overline{r}\overline{\frac{\partial v_z^f}{\partial r}}\right) \tag{3.73}$$

Integrating Equation 3.73 results in the following velocity profile,

$$\overline{v_z}^f = -\frac{a\rho_T^f v_o \overline{r}^2}{4\mu}\overline{\frac{\partial p}{\partial z}} + A\ln(\overline{r}) + B \tag{3.74}$$

where A and B are constants of integration.

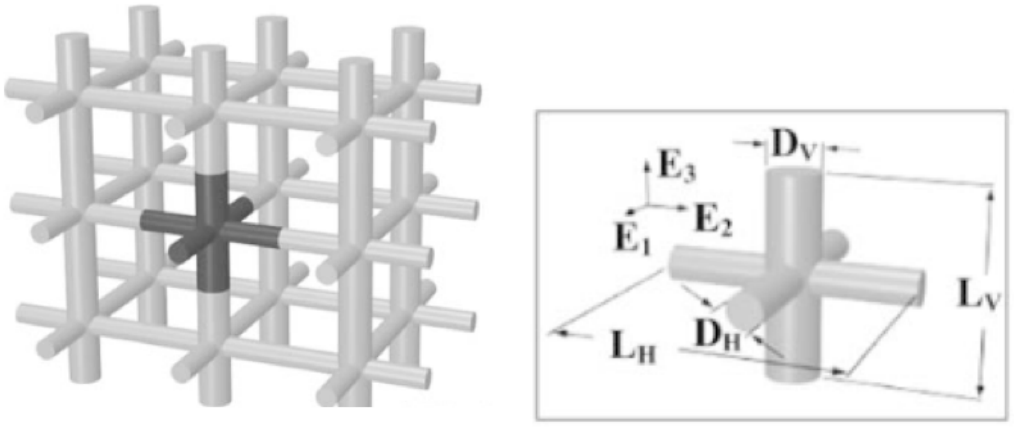
Using the boundary conditions that $\overline{v_z}^f = 0$ when $\overline{r} = \overline{a}$ and $\overline{r} = \overline{b}$, the solution of the velocity profile within the RVE in the absence of a glycocalyx is

$$\overline{v_z}^f = -\frac{a\rho_T^f v_o \overline{b^2}}{4\mu}\left(\overline{\frac{\partial p}{\partial z}}\right)\left[\left(\frac{\overline{r}}{\overline{b}}\right)^2 - \frac{\left((\frac{\overline{a}}{\overline{b}})^2 - 1\right)}{\ln(\frac{\overline{a}}{\overline{b}})}\ln\left(\frac{\overline{r}}{\overline{b}}\right) - 1\right] \tag{3.75}$$

Table 3.8 shows the values used to determine the sensitivity of these terms on the velocity profile of a Hagen-Poiseuille-type flow. Using the relationship for a Newtonian fluid $\tau_z^f = \mu\frac{\partial v_z^f}{\partial r}$, the shear stress on the osteocyte cell process ($\overline{r} = \overline{a}$) was determined using Equation 3.75,

$$\overline{\tau_z}^f = -\frac{\overline{b^2}}{4}\left(\overline{\frac{\partial p}{\partial z}}\right)\left[\left(\frac{2\overline{r}}{\overline{b^2}}\right) - \frac{\left((\frac{\overline{a}}{\overline{b}})^2 - 1\right)}{\ln(\frac{\overline{a}}{\overline{b}})}\left(\frac{1}{\overline{r}}\right)\right] \tag{3.76}$$

where $\tau_z^f = v_o^2\rho_T^f\overline{\tau_z}^f$.



(a) (b)

Fig. 3.4. (a) Assumed glycocalyx structure. (b) Dimensions of a single unit cell within the structure [68].

3.3.3 Pressure-Driven Flow: Sensitivity Analyses

Using a custom Matlab (MathWorks, Natick, MA) code, a sensitivity analysis was performed using Cotter's Method to rank the input terms based on their influence on the output (velocity) in Equation 3.69 [66] [67]. Cotter's Method involves assigning a minimum and maximum value to each input term using their respective experimental ranges. Simulations are then performed by systematically assigning one term to one of its extreme values, with all the other terms set to the opposite extreme. An additional two simulations are performed with all terms set to their minimum and all terms assigned to their maximum value, respectively. Inputs that the output is sensitive to are identified as $1/n_c$, where n_c is the total number of input terms.

As the glycocalyx within the LCS has yet to be fully characterized, approximations regarding its structure were made, in order for the permeability to be determined. An idealized open-cell model, constructed (Figure 3.4(a)) using rectangular unit cells (Figure 3.4(b)), was assumed to be within the interstitial space of the RVE. Values for Dh, Lh, Dv, and Lv were obtained from approximated dimensions used in previous

literature [17] [69] [70] [61]. These structural dimensions were used to determine the minimum and maximum permeability of the glycocalyx, using equations derived by Sander et al. [68]. In addition, the ratio of the osteocyte process radius to the canaliculus radius was used as limited osteoctye process measurements are reported. The pressure gradient was determined using the difference between the nutrient artery and nutrient vein pressures within the human tibia and an average tibial, diaphyseal cortex thickness of $6mm$. Last, approximating the interstitial fluid to be saline, the fluid viscosity at body temperature (37°C) was determined to be $0.0007Pa \cdot s$, and the true density used was $1000kg/m^3$. Minimum and maximum values for these terms are shown in Table 3.5.

From preliminary calculations, the modified Bessel function constants A and B (Equations 3.70 and 3.71), used to describe the velocity profile within the RVE (Equation 3.69), were unsolvable as the permeability of the glycocalyx decreased due to values set as the minimum and maximum glycocalyx structure terms. When solving the differential equation in Equation 3.60, the modified Bessel function form assumes the velocity is changing as a function of the radius. Thus, the solution containing constants A and B describes a Brinkman-type fluid flow. However, as the permeability of the glycocalyx decreases, the fluid flow within the RVE transitions from a Brinkman-type flow to a Darcy-type flow, where the velocity profile is not dependent upon the radius (except at the no-slip boundaries). This, then, explains the unsolvable behavior of these modified Bessel function constants.

To determine when the transition from Brinkman-type to Darcy-type flow occurs, a sensitivity analysis was performed on the γ term that the velocity profile and modified Bessel constants are functions of (Equations 3.69, 3.70, and 3.71). The γ term is a function of the canaliculus radius, the volume fraction of the fluid constituent, and the permeability of the glycocalyx. Thus, due to the changing permeability, the velocity profile and modified Bessel constants are solvable for only certain ranges of the γ term. From Figure 3.5, the γ term is sensitive to both Dh and Lh, dimensions of the glycocalyx structure.

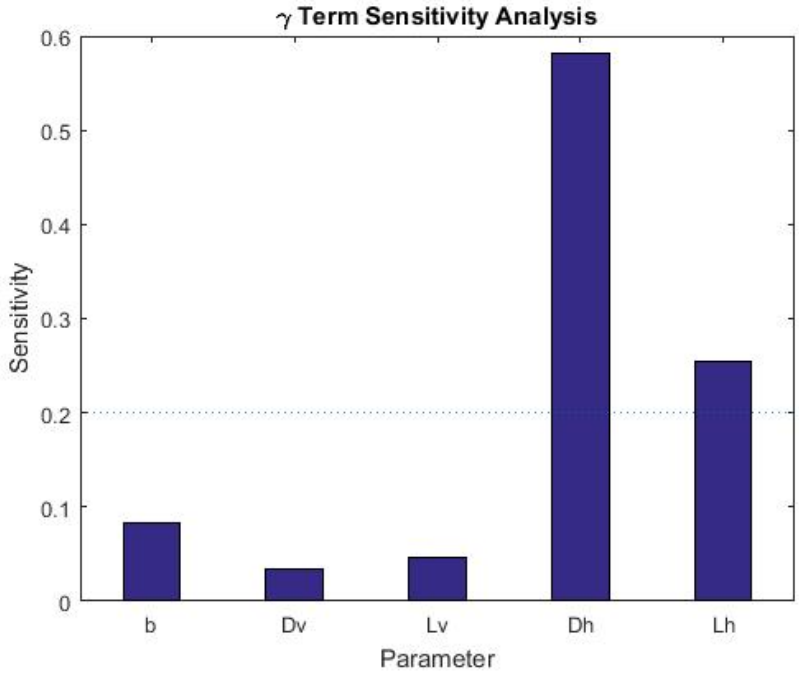


Fig. 3.5. Sensitivity analysis of the γ term, which is a function of the canaliculus radius, the volume fraction of the fluid constituent, and the permeability of the glycocalyx. The horizontal line indicates the threshold $(1/n_c)$ that a term must be greater than in order to be considered sensitive to the output. The inputs, that the γ term is sensitive to, are Dh and Lh. (b: canaliculus radius, Dv: diameter of glycocalyx vertical fibers, Lv: length of glycocalyx vertical fibers, Dh: diameter of glycocalyx horizontal fibers, Lh: length of glycocalyx horizontal fibers).

From the previous, preliminary calculations, the velocity profile and modified Bessel constants were unsolvable when γ was greater than 700. Thus, the parameters of the glycocalyx structure which are most sensitive to the γ term (Dh and Lh) were systematically modified. The maximum value for Dh and the minimum value for Lh were decreased and increased, respectively, until it resulted in the γ term less than 700. The maximum value for Dh was decreased from $6nm$ to $4.75nm$, and the minimum value for Lh was increased from $6nm$ to $7.25nm$. The modified values for each term, Table 3.6, were then used to determine their sensitivity on the velocity profile of a Brinkman-type flow. In addition, the shear stress on the osteocyte cell process ($\overline{r} = \overline{a}$) was computationally computed within Matlab, using the relationship for a Newtonian fluid $\tau_z^f = \mu \frac{\partial v_z^f}{\partial r}$; Cotter's Method was again performed to determine each term's sensitivity of this shear stress.

When the flow within the RVE is not a Brinkman-type flow due to such a small glycocalyx permeability, Darcy-type flow dominates. The first two terms on the left-hand side in Equation 3.60 are zero. Thus, the resulting Darcy-type flow profile within the RVE is as follows,

$$\overline{v}_z^f = -N\overline{k_v}\overline{\frac{\partial p}{\partial z}} \tag{3.77}$$

Before Cotter's Method could be performed to determine the sensitivity of the Darcy-type flow to its inputs, the ranges of the inputs needed to be adjusted. Overlap of the estimated values from literature exists for the glycocalyx parameters (such as Lh and Dv being equivalent), which would result in an impermeable interstitial space as the volume fraction of the solid constituent could be either greater than or equal to one. Thus, a sensitivity analysis was performed on the volume fraction (ϕ^s) of the glycocalyx within the RVE using Cotter's Method, as shown in Figure 3.6. From this sensitivity analysis, the volume fraction of the solid constituent is sensitive to both Lv and Dh. Thus, the minimum value of Lv was increased from $6nm$ to $6.75nm$, and the maximum value of Dh was decreased from $6mm$ to $5.25nm$. The value of Dv was also slightly decreased from $6nm$ to $5.9nm$; even though it was not identified as a sensitive parameter, it needed to be reduced in order to not overlap with the

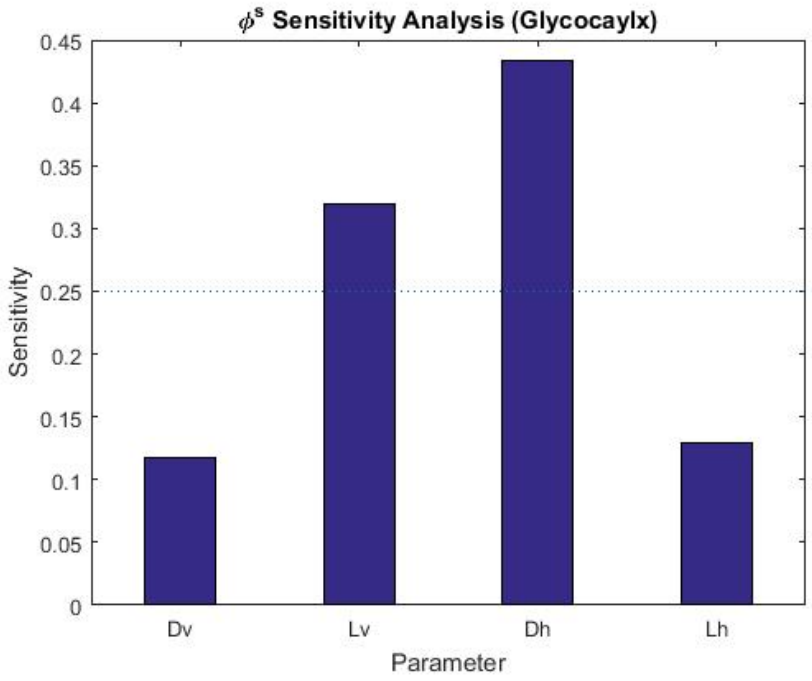


Fig. 3.6. Sensitivity analysis of the ϕ^s term, which is a function of the four geometric parameters of the glycocalyx. The horizontal line indicates the threshold $(1/n_c)$ that a term must be greater than in order to be considered sensitive to the output. The inputs, that ϕ^s is sensitive to, are Lv and Dh. (Dv: diameter of glycocalyx vertical fibers, Lv: length of glycocalyx vertical fibers, Dh: diameter of glycocalyx horizontal fibers, Lh: length of glycocalyx horizontal fibers)

minimum value of Lh, which is set at $6nm$. With these changes, ϕ^s was below 1 for all simulations. Table 3.7 shows the modified minimum and maximum values, with the blue values indicating those adjusted due to the ϕ^s sensitivity analysis and the red values indicating those previously modified due to the sensitivity analysis on the γ term. Cotter's Method was then performed on Equation 3.77, the Darcy-type flow velocity profile.

Last, Cotter's Method was performed under the assumption that the glycocalyx within the interstitial space of the RVE is not present. Sensitivity analyses were

performed for both the velocity profile and the shear stress acting on the osteocyte cell process, Equations 3.75 and 3.76, respectively.

3.3.4 Compression-Driven Flow: Preliminary Work

Representative Volume Element

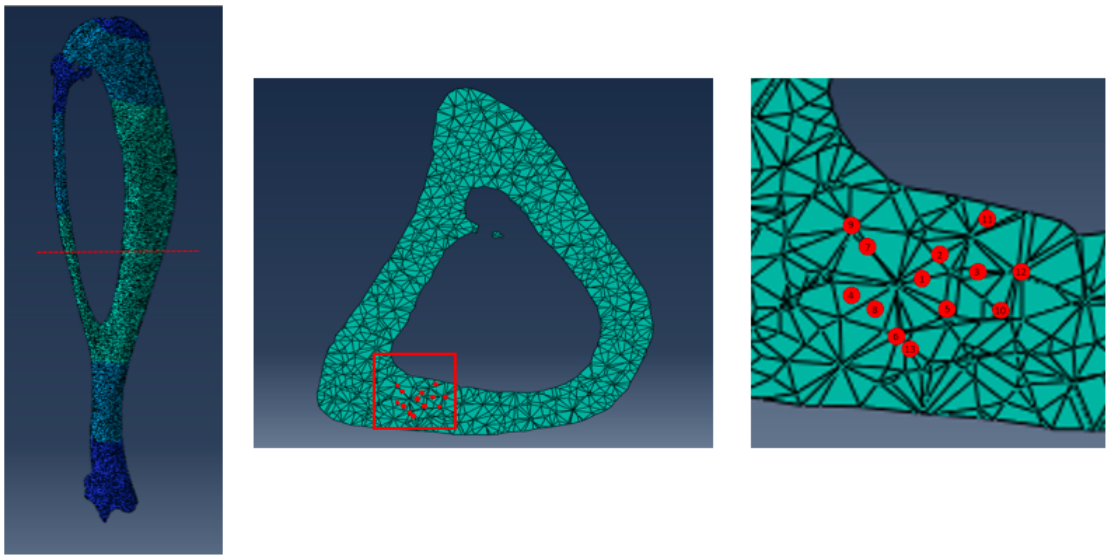
In which to model interstitial fluid flow due to compression-driven flow, the RVE consists of multiple, idealized canaliculi in parallel within a block of mineralized bone. This RVE is at length scale of a single osteocyte, or the mesoscale, as shown in Figure 3.2.

Velocity Profile within the RVE

Deformation-induced flow due to external compressive loading is next investigated as this is a common, anatomic loading that bone continually experiences, e.g. ambulation. To obtain the deformation of the solid constituent, a previously generated finite element analysis (FEA) of a mouse tibia, which underwent 12N of compressive loading in Abaqus (Johnston, RI), was utilized [74]. The model was sectioned as indicated by the dashed, red line in Figure 3.7(a). From this section, a reference node was selected, which is indicated as Node 1 in Figure 3.7(b), (c). Twelve surrounding nodes (Nodes 2-13) were next selected. The original coordinates of each node as well as the displacement coordinates were output from Abaqus. The Left Cauchy-Green deformation tensor, $\underline{B}^s$, can be determined from the original and displaced coordinates as,

$$\underline{B}^s = \underline{F}\underline{F}^T = \left(\frac{\partial \underline{x}}{\partial \underline{X}}\right)\left(\frac{\partial \underline{x}}{\partial \underline{X}}\right)^T \tag{3.78}$$

Four left Cauchy-Green deformation tensors were calculated for Node 1 using the reference node with Nodes 2-4, Nodes 5-7, Nodes 8-10, and Nodes 11-13, respectively.



(a) (b) (c)

[H]

Fig. 3.7. (a) Mouse tibia model in Abaqus, mapping the displacement due to a compressive 12N load at the proximal end [74]. The dashed, red line indicates the location where the bone was computationally sectioned. (b) Cross-section of the cut made in the model. Nodes used to calculate the deformation gradient are indicated by red dots. (c) A close-up image of the red box in (b), showing the locations of the nodes. Node 1 is the reference node.

They were then averaged to obtain an overall, average deformation tensor for reference Node 1,

$$\underline{B}^s = \begin{bmatrix} 1.0032 & 1.8818e-4 & -0.0014 \\ 1.8818e-4 & 1.0026 & 0.0028 \\ -0.0014 & 0.0028 & 0.9885 \end{bmatrix} \tag{3.79}$$

In order to solve for the stress tensor of the solid constituent $\underline{T}^s$, the material constants in Equation 3.40, β_o, β_1, and β_2, need to be determined. The Mooney-Rivlin strain energy function for a porous material needs to be utilized,

$$W = c_1(I_1 - 3) + c_2(I_2 - 3) \tag{3.80}$$

where c_1 and c_2 are constants and I_1 and I_2 are the first and second invariants of the Right Cauchy-Green deformation tensor $\underline{C}$, respectively [75]. $\underline{C}$, I_1, and I_2 are defined as follows,

$$\underline{C} = \underline{F}^T \underline{F} \tag{3.81}$$

$$I_1 = tr\underline{C} \tag{3.82}$$

$$I_2 = \frac{1}{2}\Big((tr\underline{C})^2 - tr\underline{C}^2\Big) \tag{3.83}$$

The Mooney-Rivlin strain energy function is related to the the Second Piola-Kirchhoff stress, $\underline{S}$, by the following

$$\underline{S} = 2\frac{\partial W}{\partial \underline{C}} = 2\frac{\partial W}{\partial \underline{I}}\frac{\partial \underline{I}}{\partial \underline{C}} \tag{3.84}$$

where $\frac{\partial I_1}{\partial \underline{C}}$ and $\frac{\partial I_2}{\partial \underline{C}}$ are the following,

$$\frac{\partial I_1}{\partial \underline{C}} = \underline{I} \tag{3.85}$$

$$\frac{\partial I_2}{\partial \underline{C}} = I_1\underline{I} - \underline{C} \tag{3.86}$$

Thus, the Second Piola-Kirchhoff stress tensor is,

$$\underline{S} = 2c_1\underline{I} + 2c_2 tr(\underline{C})\underline{I} - 2c_2\underline{C} \tag{3.87}$$

$\underline{S}$ can then be related to the First Piola-Kirchhoff stress tensor,

$$\underline{P} = \underline{FS} \tag{3.88}$$

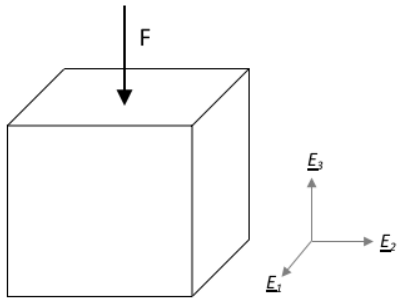


Fig. 3.8. Block of mineralized bone. Tissue is assumed to undergo compression in the E_3 direction as indicated by force F.

As the First Piola-Kirchhoff stress quantifies engineering stress, the following relationship can be made,

$$E\epsilon = \underline{FS} \tag{3.89}$$

where E is the elastic modulus of bone and ϵ is strain.

In order to solve for constants c_1 and c_2 in Equation 3.80, the compression of a block of mineralized bone is assumed, as shown in Figure 3.8. The kinematics of this motion can be derived as follows,

$$x_1 = X_1 - \eta\epsilon X_1 \tag{3.90}$$

$$x_2 = X_2 - \eta\epsilon X_2 \tag{3.91}$$

$$x_3 = X_3 + \epsilon X_3 \tag{3.92}$$

where X_i refers to the reference configuration, x_i refers to the current configuration, and η is Poisson's ratio of cortical bone.

Once the material constants are determined, the balance of linear momentum for each constituent, as shown in Section 3.3.2, can be utilized to determine the flow profile within the interstitial space and predict the shear stresses and/or drag forces that the osteocyte may be experiencing.

3.4 Results

3.4.1 Pressure-Driven Flow: Sensitivity Analyses

When utilizing Cotter's Method, a term is deemed sensitive if its sensitivity is greater than $1/n_c$, where n_c is the total number of input terms. The Brinkman-type velocity profile within the RVE (glycocalyx present) is most sensitive to Dv and Dh, as shown in Figure 3.9. The nondimensionalized maximum velocity of the Brinkman-type flow (the midpoint between the canalicular wall and the osteocyte cell process) within the RVE containing a glycocalyx is shown in Figure 3.10(a), using the averages of $\frac{\partial p}{\partial z}$, b, and v_o. It is plotted as a function of increasing osteocyte cell process radius and γ. These same results are plotted in Figure 3.10(b) but have been re-dimensionalized. The shear stress on the osteocyte cell process due to this Brinkman-type velocity profile is most sensitive to terms Dv, Dh, and v_o (Figure 3.11). Considering all of the simulations performed, the shear stress on the osteocytic cell process ranges from $0.3306\mu Pa$ to $0.01186Pa$. When the permeability is small enough that the γ term is larger than 700, the Darcy-type velocity profile occurs within the RVE, which is most sensitive to Dv and Lh, as shown in Figure 3.12. The re-dimensionalized velocity ranges from is $0.0119nm/s$ to $0.0078\mu m/s$ across all simulations performed.

When the glycocalyx is assumed to not exist, the velocity profile can be modeled as a Hagen-Poiseuille equation. This velocity within the RVE is most sensitive to v_o and b, as shown in Figure 3.13. The nondimensionalized maximum velocity of the Poiseuille-type flow (the midpoint between the canalicular wall and the osteocyte cell process) within the RVE is shown in Figure 3.14(b), using the averages of $\frac{\partial p}{\partial z}$, b, and v_o. It is plotted as a function of increasing osteocyte cell process radius and γ. These results re-dimensionalized are plotted in Figure 3.14(a).The shear stress on the osteocyte cell process due to this Poiseuille-type velocity profile is also most sensitive to v_o (Figure 3.15). Out of the simulations performed, the shear stress experienced on the osteocyte cell process ranges from $0.0182Pa$ to $0.4571Pa$.

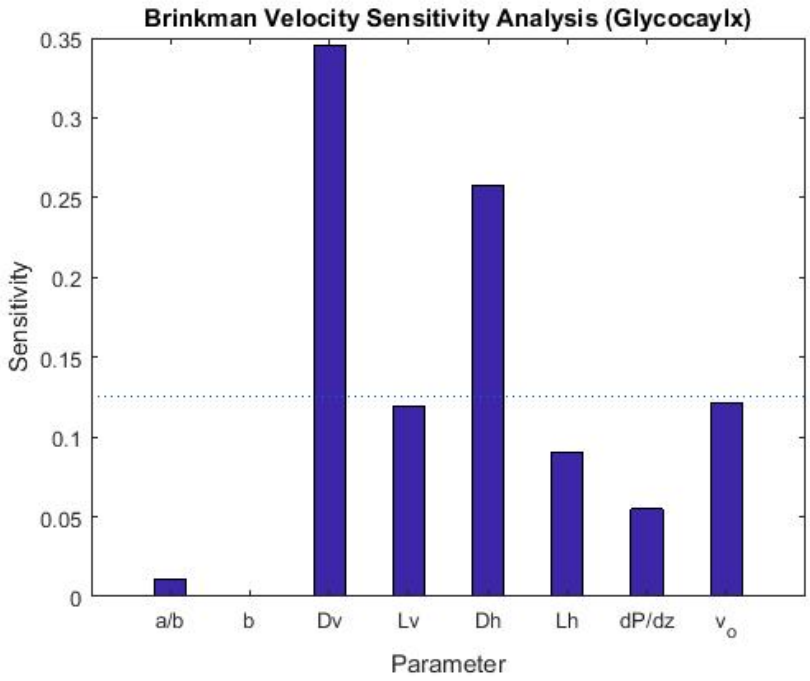
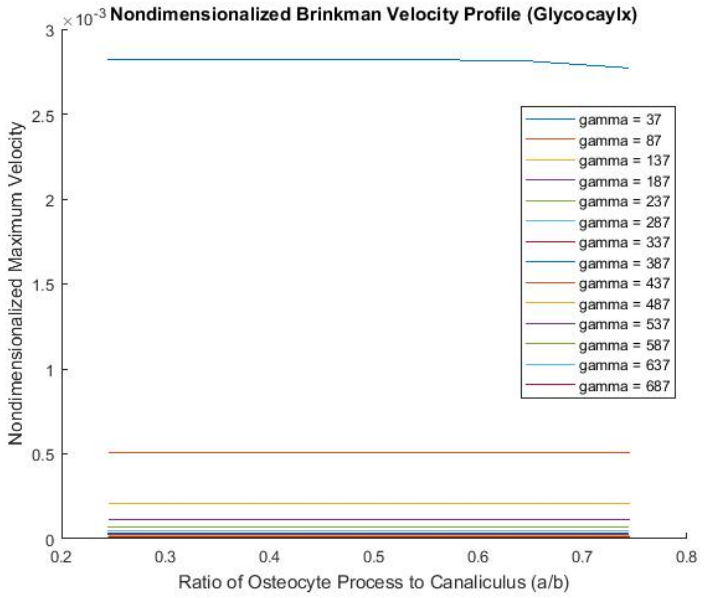
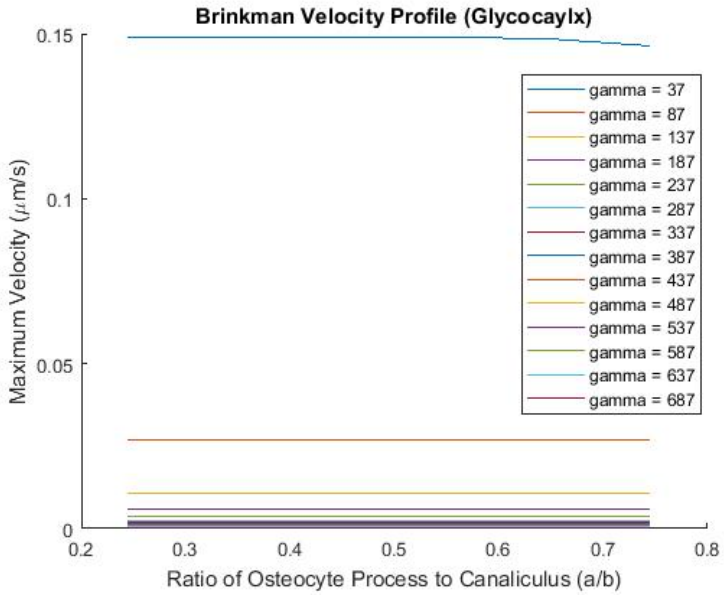


Fig. 3.9. Cotter's Method sensitivity analysis of Equation 3.69, using modified maximum and minimum values from Table 3.6. The horizontal line indicates the threshold $(1/n_c)$ that a term must be greater than in order to be considered sensitive to the output. The Brinkman-type velocity profile is most sensitive to terms Dv and Dh. (a/b: ratio of osteocyte cell process radius to canaliculus radius, b: radius of canaliculus, Dv: diameter of glycocalyx vertical fibers, Lv: length of glycocalyx vertical fibers, Dh: diameter of glycocalyx horizontal fibers, Lh: length of glycocalyx horizontal fibers, dP/dz: pressure gradient, v_o: initial velocity).



(a)



(b)

Fig. 3.10. (a) Nondimensionalized maximum velocity of the Brinkman-type flow profile, when the glycocalyx in present, as a function of increasing cell process radius and γ. Averages of $\frac{\partial p}{\partial z}$, b, and v_o were used. (b) Re-dimensionalizion of the maximum velocities shown in (a).

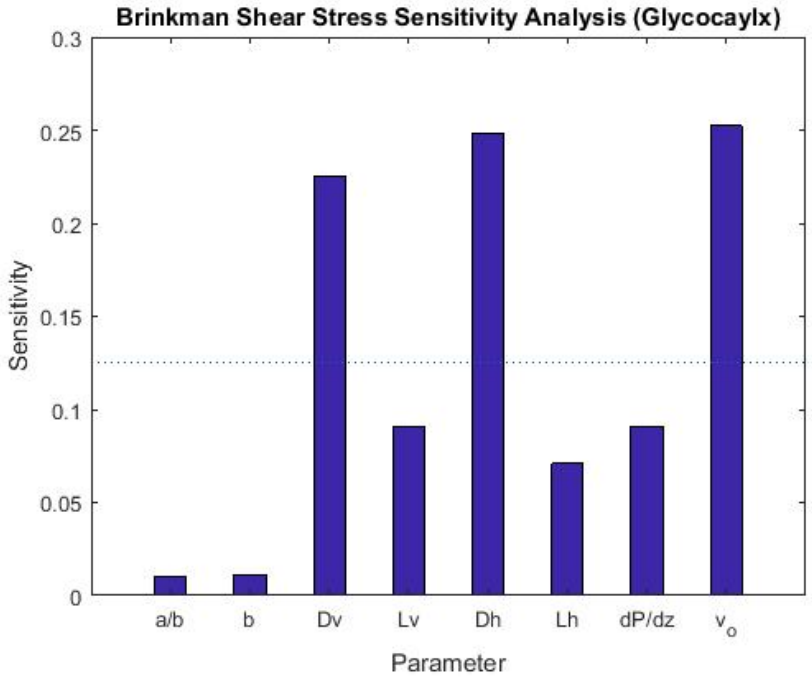


Fig. 3.11. Cotter's Method sensitivity analysis of the computationally derived shear stress on the osteocyte cell process (a) within the RVE, using modified maximum and minimum values from Table 3.6. The horizontal line indicates the threshold ($1/n_c$) that a term must be greater than in order to be considered sensitive to the output. The shear stress due to this Brinkman-type velocity profile is most sensitive to terms Dv, Dh, and v_o. (a/b: ratio of osteocyte cell process radius to canaliculus radius, b: radius of canaliculus, Dv: diameter of glycocalyx vertical fibers, Lv: length of glycocalyx vertical fibers, Dh: diameter of glycocalyx horizontal fibers, Lh: length of glycocalyx horizontal fibers, dP/dz: pressure gradient, v_o: initial velocity)

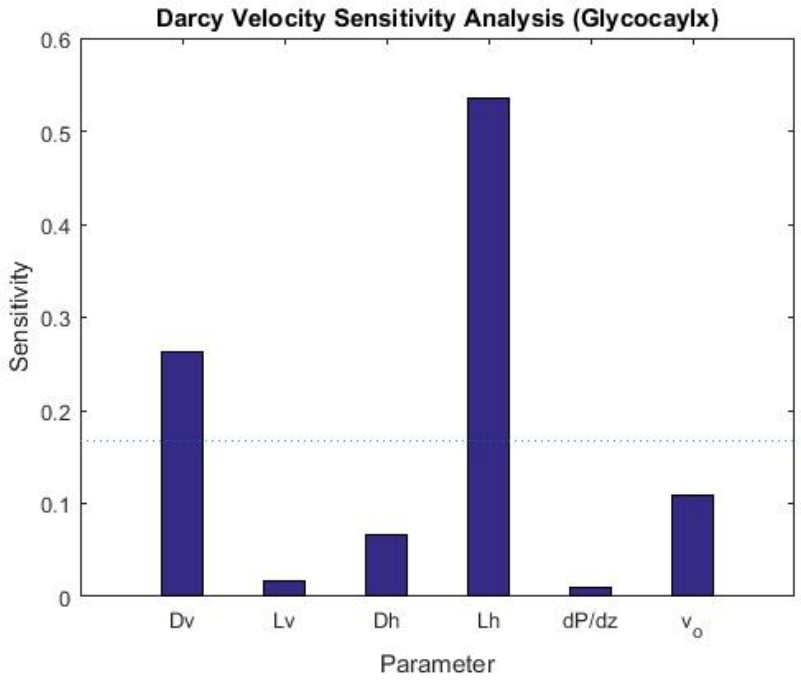


Fig. 3.12. Cotter's Method sensitivity analysis of Equation 3.77, using modified maximum and minimum values from Table 3.7. The horizontal line indicates the threshold $(1/n_c)$ that a term must be greater than in order to be considered sensitive to the output. The Darcy-type velocity profile is most sensitive to terms Dv and Lh. (Dv: diameter of glycocalyx vertical fibers, Lv: length of glycocalyx vertical fibers, Dh: diameter of glycocalyx horizontal fibers, Lh: length of glycocalyx horizontal fibers, dP/dz: pressure gradient, v_o: initial velocity).

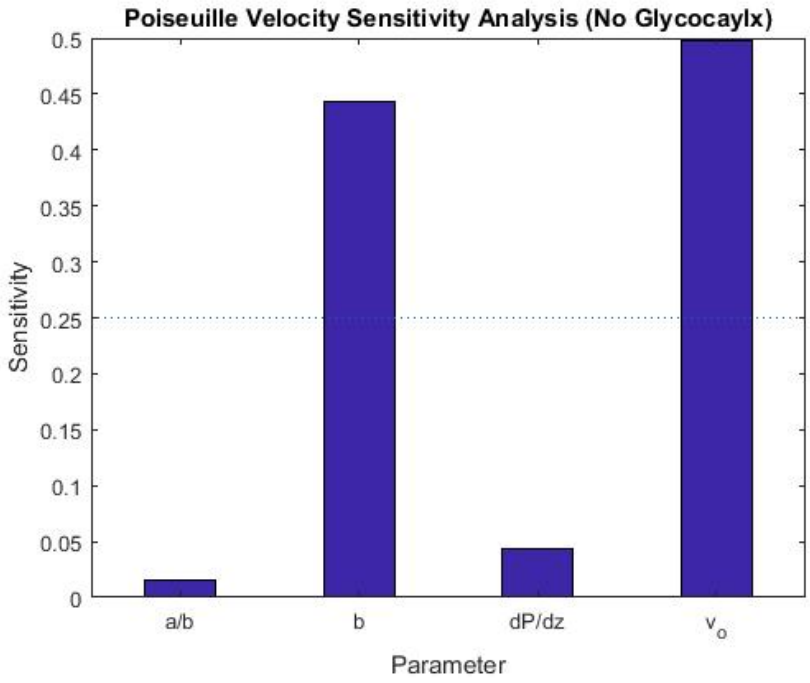
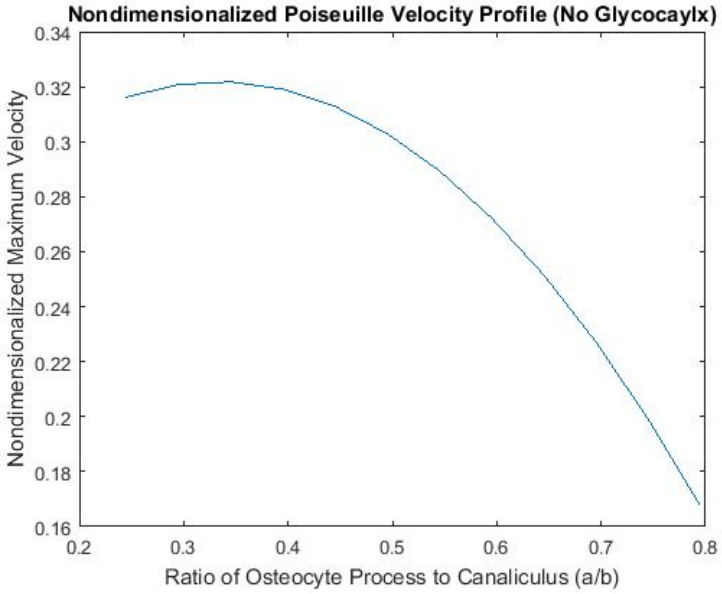
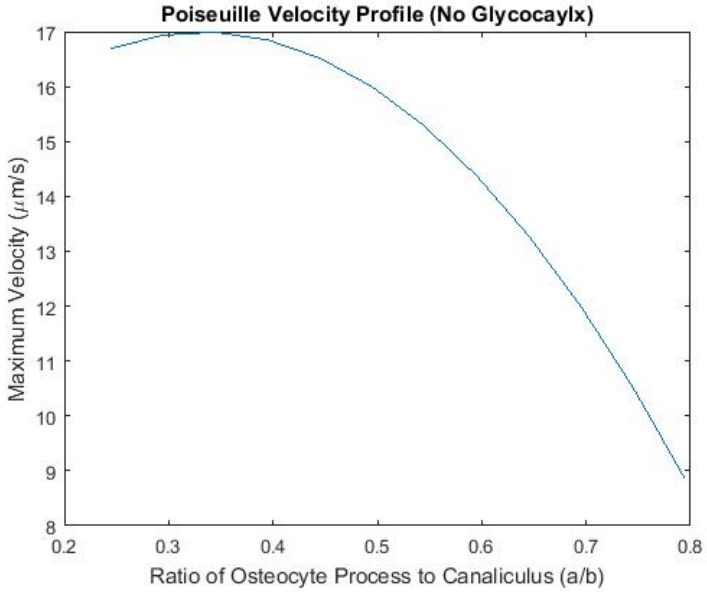


Fig. 3.13. Cotter's Method sensitivity analysis of Equation 3.75, using modified maximum and minimum values from Table 3.8. The horizontal line indicates the threshold ($1/n_c$) that a term must be greater than in order to be considered sensitive to the output. The Poiseuille-type velocity profile (assuming no glycocalyx) is most sensitive to v_o and b. (a/b: ratio of osteocyte cell process radius to canaliculus radius, b: radius of canaliculus, dP/dz: pressure gradient, v_o: initial velocity).



(a)



(b)

Fig. 3.14. (a) Nondimensionalized maximum velocity of the Poiseuille-type flow profile as a function of increasing cell process radius and γ. Averages of $\frac{\partial p}{\partial z}$, b, and v_o were used. (b) Re-dimensionalization of the maximum velocities shown in (a).

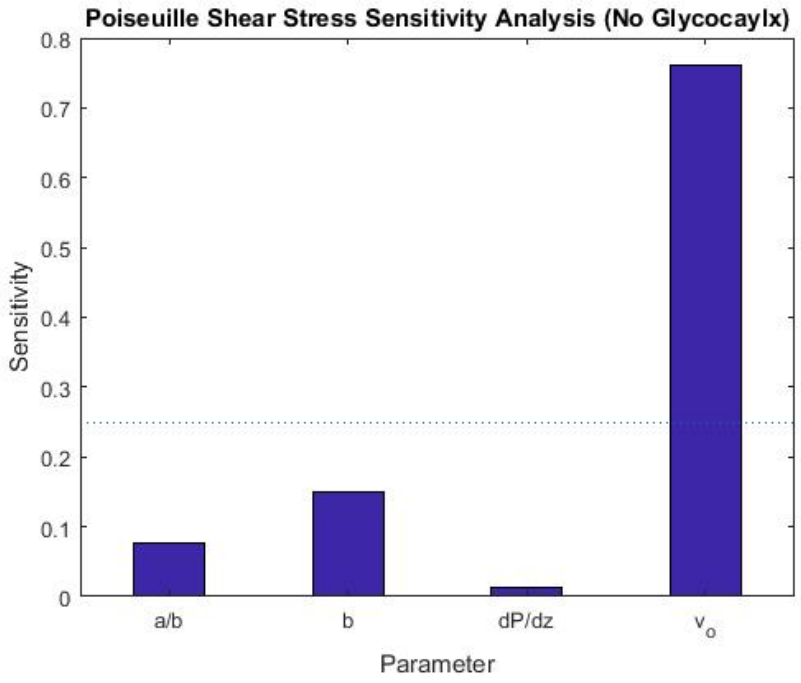


Fig. 3.15. Cotter's Method sensitivity analysis of the shear stress acting on the osteocyte cell process (a) within the RVE (Equation 3.76), using modified maximum and minimum values from Table 3.8. The horizontal line indicates the threshold ($1/n_c$) that a term must be greater than in order to be considered sensitive to the output. The shear stress due to this Poiseuille-type velocity profile (assuming no glycocalyx) is most sensitive to v_o. (a/b: ratio of osteocyte cell process radius to canaliculus radius, b: radius of canaliculus, dP/dz: pressure gradient, v_o: initial velocity).

3.5 Discussion

By utilizing the theory of poroelasticity and the mixture theory approach, velocity and shear stresses within an idealized canaliculus were able to be determined, along with the impact of the glycocalyx on these results. Based upon the presence and structure of the glycocalyx, three different types of flow were determined to exist within the interstitial space: 1) Brinkman flow, 2) Darcy flow, and 3) Poiseuille flow. Brinkman flow occurs when a glycocalyx was assumed to exist and its permeability results in a γ term greater than 700. As the permeability decreases, causing an increase in the γ term above 700, the flow is modeled as a Darcy-type flow. Last, when no glycocalyx is assumed to exist, Poiseuille flow exists between the canalicular wall and the osteocyte cell process.

Due to arterial pressure-driven flow, the Brinkman-type maximum velocity is approximately 0.15 $\mu m/s$ when γ is equal to 37. As the γ term increases to just 87, the maximum velocity within the interstitial space is greatly reduced to just 0.025 $\mu m/s$. The shear stress on the ostocytic process due to the Brinkman flow ranges from 0.3306 μPa to 0.01186 Pa. Both the velocity and shear stress are sensitive to the glycocalyx diametric parameters Dv and Dh with the shear stress also being sensitive to the initial velocity v_o. When the γ term is greater than 700, the maximum Darcy velocity that occurs ranges from 0.0119 nm/s to 0.0078 $\mu m/s$; this velocity is also sensitive to structural parameters of the glycocalyx, Dv and Lh. When a glycocalyx is not assumed to be within the interstitial space, Poisueille flow results in a maximum velocity of 17 $\mu m/s$ when the ratio of the osteocyte process to the canalicular wall is 0.35 and decreases parabolically to approximately 9 $\mu m/s$ when the the ratio is increased to 0.80. The shear stress on the cell process ranges from 0.0182 Pa to 0.4571 kPa. The Poiseuille velocity and shear stress are both sensitive to the initial velocity v_o, with the velocity also being sensitive to the canalicular radius b.

From these results, it is shown that the presence of glycocalyx greatly impacts not just the values of the velocity and shear stress but also the type of flow that dominates

within the interstitial space. While this model does assume a rectangular unit cell of the glycocalyx, it still clearly shows the impact of the ranges of values that have been assumed of the glycocalyx [17] [69] [70]. For example, Kamioka et al. did not assume a glycocalyx in their UHVEM canalicular reconstruction (Figure 2.8(a)), which lead to maximum velocity values of 800 $\mu m/s$. In comparison to the idealized model Poiseuille velocities modeled within this thesis, the irregularities from the reconstructed images as well as a much larger pressure gradient of 1 Pa/nm [10] in Kamioka et al.'s model explain the large discrepancies observed between the two models [6]. On the other hand, Verbruggen et al. assumed a glycocalyx within their computational models of CLSM lacunar canalicular structures, resulting in maximum velocities of only 200 $\mu m/s$ (Figure 2.8(b)). They reported an average velocity of approximately 60.5 $\mu m/s$ and an average shear stress of approximately 11 Pa. Again, differences in this model and the one discussed here are due to the non-idealized structure of the LCS and the modeling of vigorous activity by Vergruggen et al. (Input pressure: 300 Pa) [19]. Thus, mathematical and computational models need to take these uncertainties of the glycocalyx into account, especially since the type of dominating flow will enable determination of the type of stimulus the cell process is experiencing, i.e. shear stress or drag forces/amplified hoop strains.

Finally, expanding this mixture theory analysis to model compression-driven flow, as in Section 3.3.2, will enable more relevant, physiologic constituent relationships to be determined. It will be able to take into account the deformation of the solid and determine its impact on the fluid profile within the interstitial space with or without the presence of a glycocalyx. Overall, these results will be a more accurate prediction to the forces that the osteocyte may be experiencing during mechanical loading.

Table 3.5.
High and low values for each input term in Equation 3.69 for implementing Cotter's Method.

Term	**Variable**	**Low**	**Method**	**High**	**Method**
Canaliculus radius (nm)	b	75	SEM, human tibiae [34]	422	AFM, bovine tibiae [27]
Ratio of osteocyte process radius to canaliculus radius	a/b	0.245	TEM, rat tibiae [71]	0.819	TEM, rat tibiae [71]
Diameter of vertical fibers (nm)	Dv	0.5	Estimated value [17] [69]	6	Estimated value [17] [69]
Length of vertical fibers (nm)	Lv	6	Tracer study, rat tibiae [70]	10	Tracer study, rat tibiae [61]
Diameter of horizontal fibers (nm)	Dh	0.5	Estimated value [17] [69]	6	Estimated value [17] [69]
Length of horizontal fibers (nm)	Lh	6	Tracer study, rat tibiae [70]	10	Tracer study, rat tibiae [61]
Pressure gradient (kPa/m)	$\partial p/\partial z$	-1978.33	Change in nutrient vascular pressure, human tibiae [72] [73]	-2645	Change in nutrient vascular pressure, human tibiae [72] [73]
Initial velocity (μm/s)	v_o	21.6	FRAP and mechanical loading, murine tibiae [17]	84	FRAP and mechanical loading [46]

Table 3.6.
Modified high and low values for each input term in Equation 3.69 (Brinkman-type flow) for implementing Cotter's Method. Red numbers indicate modified values.

Term	**Variable**	**Low**	**Method**	**High**	**Method**
Canaliculus radius (nm)	b	75	SEM, human tibiae [34]	422	AFM, bovine tibiae [27]
Ratio of osteocyte process radius to canaliculus radius	a/b	0.245	TEM, rat tibiae [71]	0.819	TEM, rat tibiae [71]
Diameter of vertical fibers (nm)	Dv	0.5	Estimated value [17] [69]	6	Estimated value [17] [69]
Length of vertical fibers (nm)	Lv	6	Tracer study, rat tibiae [70]	10	Tracer study, rat tibiae [61]
Diameter of horizontal fibers (nm)	Dh	0.5	Estimated value [17] [69]	4.75	Estimated value [17] [69]
Length of horizontal fibers (nm)	Lh	7.25	Tracer study, rat tibiae [70]	10	Tracer study, rat tibiae [61]
Pressure gradient (kPa/m)	$\partial p/\partial z$	-1978.33	Change in nutrient vascular pressure, human tibiae [72] [73]	-2645	Change in nutrient vascular pressure, human tibiae [72] [73]
Initial velocity (μm/s)	v_o	21.6	FRAP and mechanical loading, murine tibiae [17]	84	FRAP and mechanical loading [46]

Table 3.7.
Modified high and low values for each input term in Equation 3.77 (Darcy-type flow) for implementing Cotter's Method. Red values indicate modified values to satisfy the γ sensitivity analysis, and blue values indicate those modified due to the ϕ^s sensitivity analysis.

Term	**Variable**	**Low**	**Method**	**High**	**Method**
Diameter of vertical fibers (nm)	Dv	0.5	Estimated value [17] [69]	5.9	Estimated value [17] [69]
Length of vertical fibers (nm)	Lv	6.75	Tracer study, rat tibiae [70]	10	Tracer study, rat tibiae [61]
Diameter of horizontal fibers (nm)	Dh	4.75	Estimated value [17] [69]	5.25	Estimated value [17] [69]
Length of horizontal fibers (nm)	Lh	6	Tracer study, rat tibiae [70]	7.25	Tracer study, rat tibiae [61]
Pressure gradient (kPa/m)	$\partial p/\partial z$	-1978.33	Change in nutrient vascular pressure, human tibiae [72] [73]	-2645	Change in nutrient vascular pressure, human tibiae [72] [73]
Initial velocity (μm/s)	v_o	21.6	FRAP and mechanical loading, murine tibiae [17]	84	FRAP and mechanical loading [46]

Table 3.8.
High and low values for each input term in Equation 3.75 (Poiseuille-type flow) for implementing Cotter's Method, using the assumption that the glycocalyx is not present.

Term	**Variable**	**Low**	**Method**	**High**	**Method**
Canaliculus radius (nm)	b	75	SEM, human tibiae [34]	422	AFM, bovine tibiae [27]
Ratio of osteocyte process radius to canaliculus radius	a/b	0.245	TEM, rat tibiae [71]	0.819	TEM, rat tibiae [71]
Pressure gradient (kPa/m)	$\partial p/\partial z$	-1978.33	Change in nutrient vascular pressure, human tibiae [72] [73]	-2645	Change in nutrient vascular pressure, human tibiae [72] [73]
Initial velocity (μm/s)	v_o	21.6	FRAP and mechanical loading, murine tibiae [17]	84	FRAP and mechanical loading [46]

4. CHARACTERIZATION OF THE LCS

4.1 Motivation

Advancements in nanoscale imaging and reconstruction softwares have enabled three-dimensional characterization of the LCS. However, the current techniques utilized to reconstruct both the lacunar and canalicular morphology require either manual segmentation which is subject to a user bias [35] [37] [38], multiple softwares [35] [37] [38], or techniques, such as FIB-SEM and SR-PNT, that are not as accessible as CLSM [5] [35] [44]. Hesse et al. and Varga et al. have shown promising results with their custom-implemented Matlab codes [5] [44]; however, creation of a similar code may prove difficult and time-consuming for researchers without the computational background. Thus, an open-source code reconstructing the LCS from CLSM images, as Heveran et al. provided for the lacunae, would enable further advancements in the field, removing the variable of post-processing from the differences that are seen in characterization of the LCS.

4.2 Methods

4.2.1 Sample Preparation

For this study, the femora of 20-week-old and 52-week-old female wild type C57BL/6 mice were utilized to visualize the lacunar-canalicular structure within cortical bone. These mice, as genetic controls for previous experiments, had been previously sacrificed. The left femora from the 20-week-old mice and right femora from the 52-week-old mice had been harvested, cleaned of soft tissue, fixed in 10% neutral buffered formalin (NBF), and stored in 70% ethanol (EtOH) (Fisher Scientific, Hampton, NH) following euthanasia.

Three femora from each age group were obtained for sectioning. In order to handle and support the bones, Eppendorf PCR tubes were filled approximately three-fourths full with Quickset Epoxy (Loctite, Düsseldorf, GER). Once the epoxy was set, each femur was inserted into a PCR tube so that the distal end of the bone was in contact with the epoxy. Holding the femora upright, the remainder of the PCR tubes were filled with epoxy. A thin coat of epoxy was then applied around the portions of the bones not embedded within the PCR tube; this was done in order to prevent splintering of the bone during sectioning.

The PCR tubes were horizontally inserted into a single saddle chuck (Buehler, Lake Bluff, IL) which was attached to the swivel arm of an Isomet 1000 Precision Cutter (Buehler, Lake Bluff, IL). A diamond blade was used to cut multiple cross sections from the metaphysis and diaphysis of each bone. Samples were then stored in individual PCR tubes filled with 70% EtOH for at least 24 hours.

After soaking in EtOH, the epoxy coating the outside of each section was peeled off using forceps. Using deionized water and 600 grit waterproof sandpaper (Norton, Worchester, MA), one metaphyseal cross-section from each mouse bone was ground down to a final thickness of 150 μm. Each section was washed in ascending concentrations of 70%, 85%, and 100% EtOH for 5 minutes each. Samples were placed in a 24-well plate (one sample per well) and stained with 0.1 mg/mL Alexa Fluor 488 carboxylic acid, succinimidyl ester (Invitrogen, Carlsbad, CA) for 24 hours. During this incubation time, the covered 24 well-plate was placed on a standard analog shaker (Setting 1; VWR, Radnor, PA) to ensure penetration into the LCS.

After the Alexa 488 was removed from the wells, each sample was submerged in 10% NBF and again placed on the shaker for at least 48 hours. Once the samples were fixed, ascending concentrations EtOH were again used to wash the samples. Last, samples were placed on microscope slides; no. 1.5 cover glass (VWR, Radnor, PA) and ProLong Diamond Antifade Mountant (Invitrogen, Carlsbad, CA) were used to secure the samples on the slides. Slides were covered and left to dry for at least 36 hours before imaging.

4.2.2 Confocal Imaging

All slides were imaged on a Zeiss LSM 880 Upright Confocal (ZEISS, Oberkochen, GER), using the Plan-Apochromat 63x/1.40 Oil DIC objective. In order to keep image settings consistent between samples, all slides were first briefly imaged to determine overall optimal imaging parameters. Three z-stacks were then obtained from the posterior surface of each sample. The goal of each z-stack was to focus and obtain images on a single full lacunae and its connecting canaliculi. The x-y resolution of the images (1024 x 1024) approximately ranged from 0.030 μm - 0.040 μm, while the z-resolution was 0.394 μm. The depth, or number of slices for each z-stack, was variable as it depended on the size and orientation of the lacunae. The settings used for each z-stack for all samples were as follows: master gain: 586, pixel dwell: 0.38 μs, pinhole: 1 AU, averaging: line 8, and pixel depth: 16-bit. As laser penetration diminishes greatly with depth due to the mineralized structure of the samples, the laser power was increased from 5% to 7.5% at 50% thickness of the scan and also from 7.5% to 10% once 75% of the sample had been scanned. All z-stacks were saved with and without a scale-bar as the scale-bar interferes with automated volume reconstruction.

In addition, a z-stack was performed on the medial surface of each sample in order to make quantitative measurements regarding volumetric lacunar density. Using the Plan-Apochromat 20x/.8 M27 objective, each image (1024 x 1024) had an x-y resolution of 0.142 μm and a z-resolution of 0.832 μm, resulting in an overall image size of approximately 146 μm by 146 μm and an average depth of 33 μm. The settings for these scans were the following: laser power: 10%, master gain: 586, pixel dwell: 0.38 μs, pinhole: 1 AU, averaging: line 8, and pixel depth: 16-bit. The z-stacks were then imported into Mimics (Materialise, Leuven, BEL) where the lacunae were masked, with a lower threshold of -994 and an upper threshold of -842. From this segmentation, a three-dimensional reconstruction of the lacunae was generated, as shown in Figure 4.1. Using the annotation tool, all internal, full lacunae were counted. In addition, partial lacunae were counted if they were on the proximal (top), lateral,

or anterior surface of the reconstructed volume. In order to determine if the two ages had significantly different volumetric lacunar densities, a Kruskal-Wallis H test was performed in Stata (StatCorp, College Station, TX), with significance defined as $p < 0.05$.

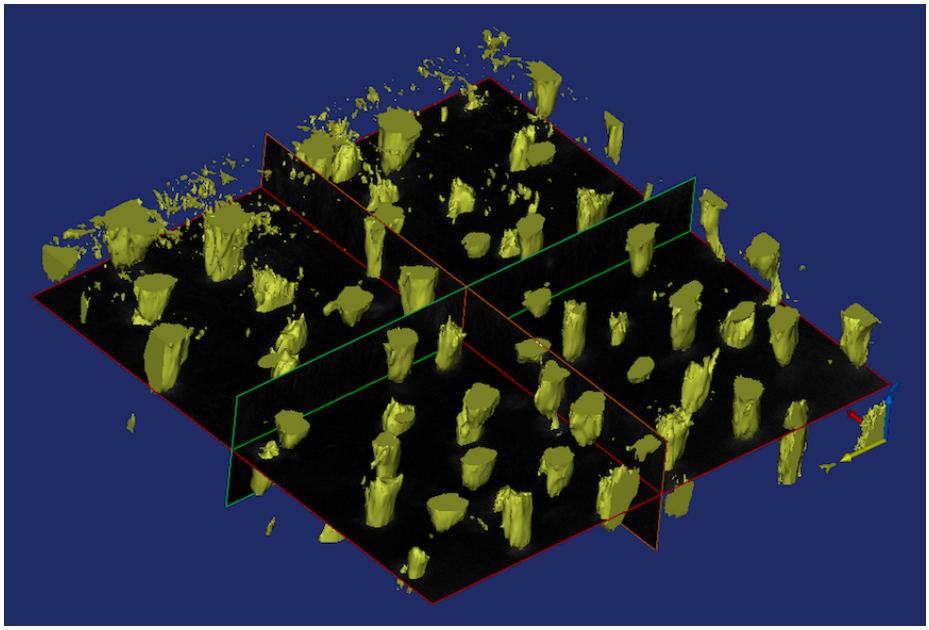


Fig. 4.1. Reconstruction in Mimics of a z-stack collected to determine volumetric lacunar density.

4.2.3 LCS 3D Reconstruction

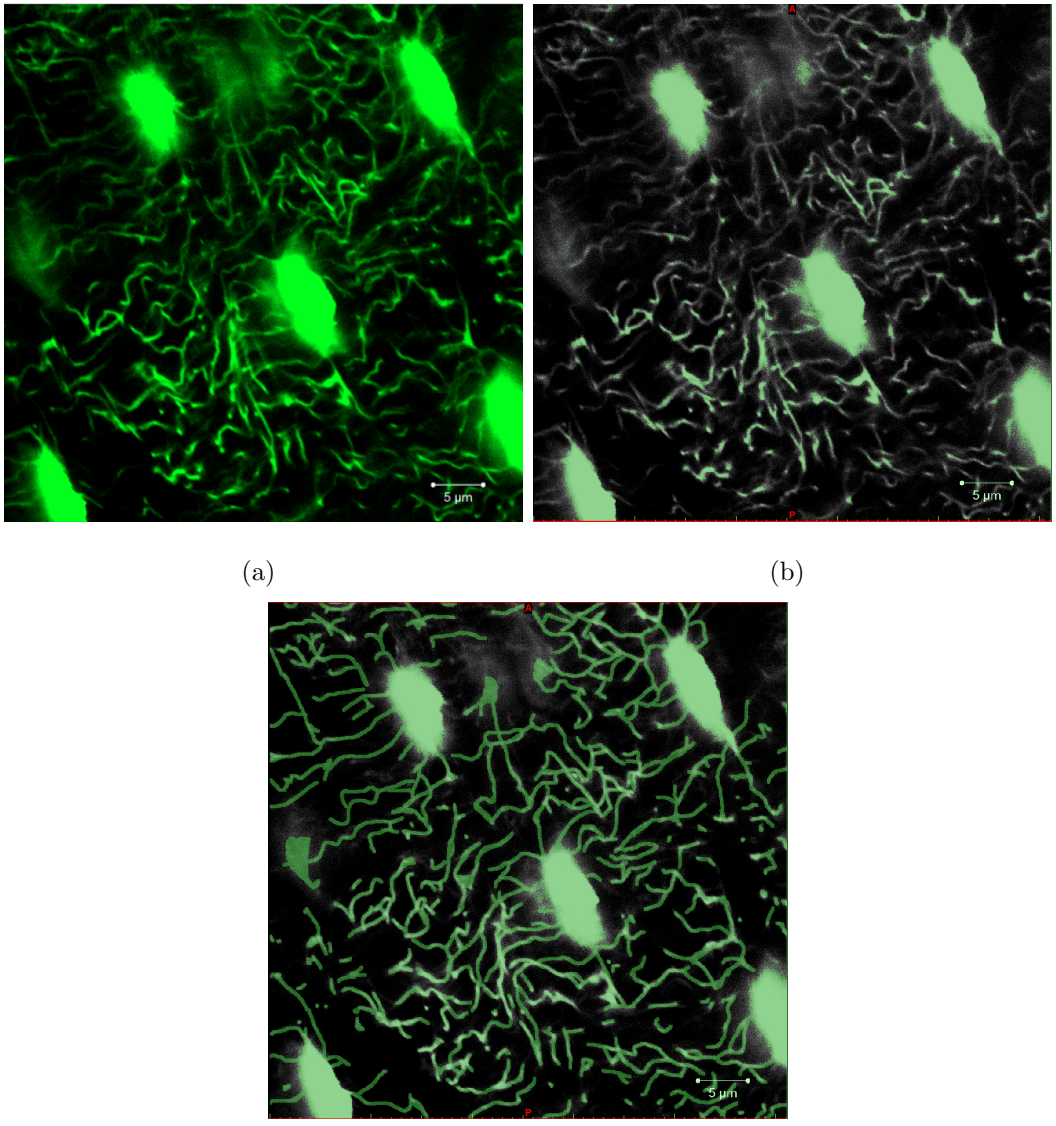
Current reconstruction softwares, such as Mimics, were created for clinical, larger-scale applications, mainly to reconstruct and model computed tomography (CT) scans. As the laser power needed to be increased during confocal scanning (due to depth into the mineralized bone sample), the obtained confocal z-stacks were too noisy for the software to accurately segment, mask, and reconstruct. Using Mimics, the thresholding of the Alexa 488 stained LCS was unreliable and would have involved manual editing on every slice for each z-stack. Thus, a custom Matlab code was developed in order to reconstruct the Zeiss confocal z-stacks for each sample.

Before running the code, the x-y and z resolutions were entered into the code for 'xyRes' and 'zRes' variables, respectively. These are the only parameters that *need* to be modified. Additionally, the following parameters can be either set to '0' for false or '1' for true: 'display3D', 'displayRaw', and 'display3DCan', 'runValidation'. 'display3D' allows for an output three-dimensional representation of the lacuna of interest and its connecting cananliculi. 'displayRaw' displays the raw data slices and 'display3DCan' displays the final reconstruction of the z-stack containing just the main lacuna of interest and all the canaliculi. 'runValidation' allows for validation of the segmentation processes (given a manually thresholded z-stack in addition to the raw dataset).

By running the custom code, the two-dimensional image (TIF) files of a z-stack (without the scale bars) were imported into Matlab (when prompted), where they were first converted into greyscale. A simple image adjustment was next automatically performed on each slice to maximize its contrast. Another image adjustment was automatically executed to conform all slices to the same intensity scale. Next, a non-localized means filter was utilized for noise filtering on each slice [76] [77] [78]. In order to segment the lacunae and canaliculi, a 2D Frangi filter, which is a Hessian-based multi-scale filter, was used [79] [80] [81], followed by smoothing of the canalculi [82] [83] [84]. Finally, both the lacunae and canaliculi were masked.

In order to determine the accuracy of this segmentation process, a validation dataset was created using Mimics. A previously prepared mouse femoral transverse cross-section, stained with 0.1 mg/mL Alexa 488, was scanned using a Zeiss LSM Confocal. A 51.58 μm x 51.58 μm x 14.31 μm z-stack was obtained (16-bit, 1024 x 1024 x 30). The following settings were utilized: laser power: 5.0%, master gain: 759-784, pixel dwell: 0.34 μs, averaging: line 8, and pinhole: 1.29 AU. Importing the 30 slices into Mimics, a mask was created, and the images were thresholded. On each z-stack slice, the mask was edited by manually drawing in regions of the LCS that had not been captured by the thresholding. Figure 4.2 shows an example of this process on slice 10 of the z-stack where (a) is the original confocal image, (b) is the thresholded

slice in Mimics, and (c) is the manually reconstructed image in Mimics. Following contouring, the corrected dataset was reconstructed (for visualization purposes) in Mimics, as shown in Figure 4.3.



(a) (b)

(c)

Fig. 4.2. (a) Slice 10 of the validation data z-stack. (b) Slice after being thresholded in Mimics software. (c) Manual reconstruction of the confocal image within Mimics.

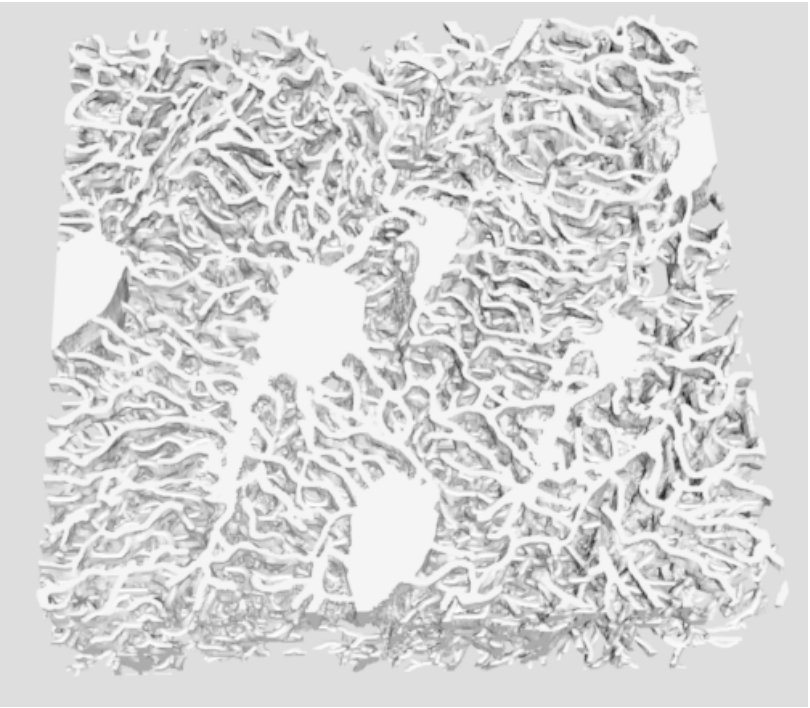


Fig. 4.3. Reconstruction in Mimics of the validation data set from the confocal z-stack.

In order to assess the code's accuracy, the variable 'runValidation' was set from '0' (false) to '1' (true) in order for the validation dataset to be analyzed. The raw 2D slices of the validation dataset were read into Matlab, segmented, masked, and then compared to the 2D slices of the Mimics-corrected dataset. The accuracy of both the lacunae and canaliculi were determined using the following equation,

$$accuracy = \frac{RegionOverlap}{RegionOverlap + ReconstructedData + RawData} \tag{4.1}$$

where *RegionOverlap* is the area of overlap of the raw dataset and the reconstructed dataset, *ReconstructedData* is the area that the reconstructed dataset demarcates as part of the LCS whereas the raw dataset does not, and *RawData* is the area that the raw dataset delineates as the part of the LCS whereas the reconstructed dataset does not.

Once the accuracy was determined, the 2D canaliculi were skeletonized on each slice [85], and an approximated average canalicular diameter was determined using the total 2D area of the canaliculi divided by the total length of the canaliculi. This diameter calculation assumes a circular cross-section for each canaliculi.

The data was next adjusted so that it was isometric. This was done by stacking each slice in multiples to equalize x-y and z resolutions. The 3D canaliculi and lacunae were then smoothed [82] [83] [84] before being segmented [86]. As it is the typical idealized structure of a lacuna, an ellipsoid was then automatically fitted to the largest volumetric imaged lacuna (lacuna of interest) by minimizing the error between the actual lacuna and the fitted ellipsoid [43] [87] [88]. In order to identify connecting, primary canaliculi, a shell of the lacuna was created by expanding its lacunar surface by a distance of twice the smallest ellipsoid axis length. The canaliculi were then skeletonized in 3D [89] [90], and the endpoints were used to identify which canaliculi directly connect to the lacuna (if an endpoint falls within the generated shell). Finally, the skeletonized canaliculi were dilated to the average canalicular diameter. The following outputs were generated for each z-stack, based upon the reconstructed z-stack: average canalicular diameter, average canalicular length, average canalicular porosity (volume fraction of the canaliculi within the volume of the z-stack), lacunar volume, canalicular density (with respect to lacunar volume). In addition, the following parameters were output to describe the fitted ellipsoid: ellipsoid diameters (1 major and 2 minors), ellipsoid orientation (major axis with respect to the z-axis), ellipsoid sphericity and oblateness, and ellipsoid surface area and volume [43].

Using Stata, Kruskal-Wallis H Tests were performed between the 20-week-old and the 52-week-old age groups for the measurements based upon the reconstructed z-stack (canalicular diameter, canalicular length, canalicular porosity, lacunar volume, and canalicular density). In order to account for the multiple comparisons, the Bonferroni correction was utilized so that the family-wise error rate remained at 5%; thus, for these 5 comparisons, a p-value less than 0.01 indicates significance. For the measurements based upon the fitting of the ellipsoid [ellipsoid diameters (1 major

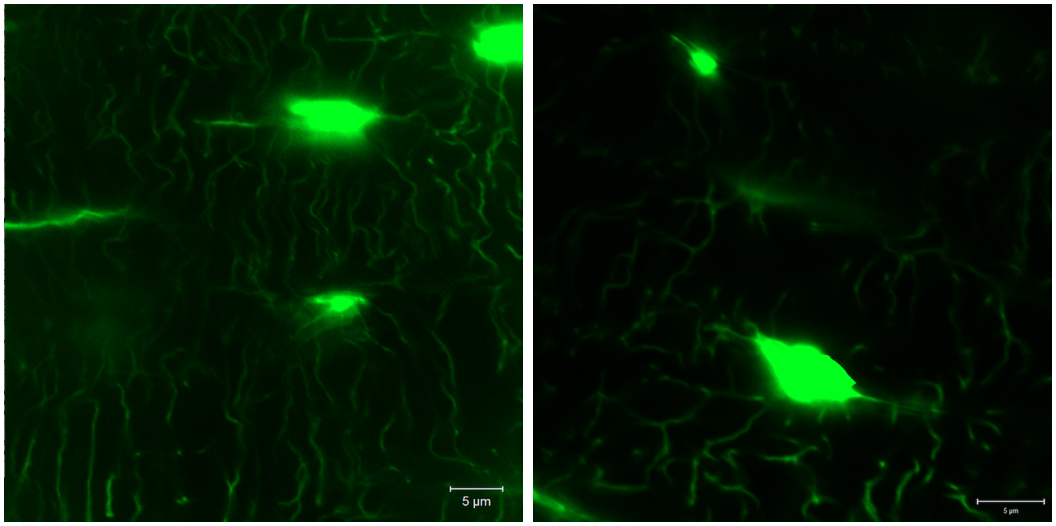
and 2 minors), ellipsoid orientation (major axis with respect to the z-axis), ellipsoid sphericity and oblateness, and ellipsoid surface area and volume], Kruskal-Wallis H Tests, using Bonferroni's correction, were again performed between age groups. A p-value less than 0.00625 indicates significance due to a total of 8 comparisons made.

4.3 Results

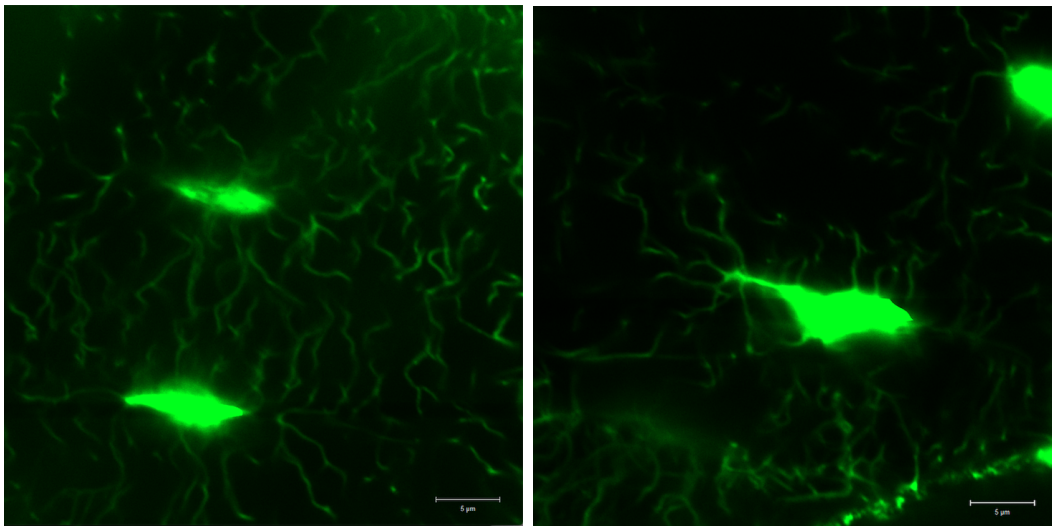
4.3.1 Confocal Imaging

Using CLSM, three z-stacks were obtained on the posterior surface of a proximal, femoral diaphyseal cross-section. A total of three mice were imaged for each age group: 20-week-old and 52-week-old. Figure 4.4 shows example z-stack slices from 20-week-old mice ((a) and (c)) and from 52-week-old mice ((b) and (d)).

Figure 4.5 illustrates slices from (a) a 20-week-old mouse z-stack and (b) a 52-week-old mouse z-stack on the medial surface of a proximal, femoral metaphyseal cross-section. These z-stacks were utilized to obtain volumetric lacunar density for each mouse age.

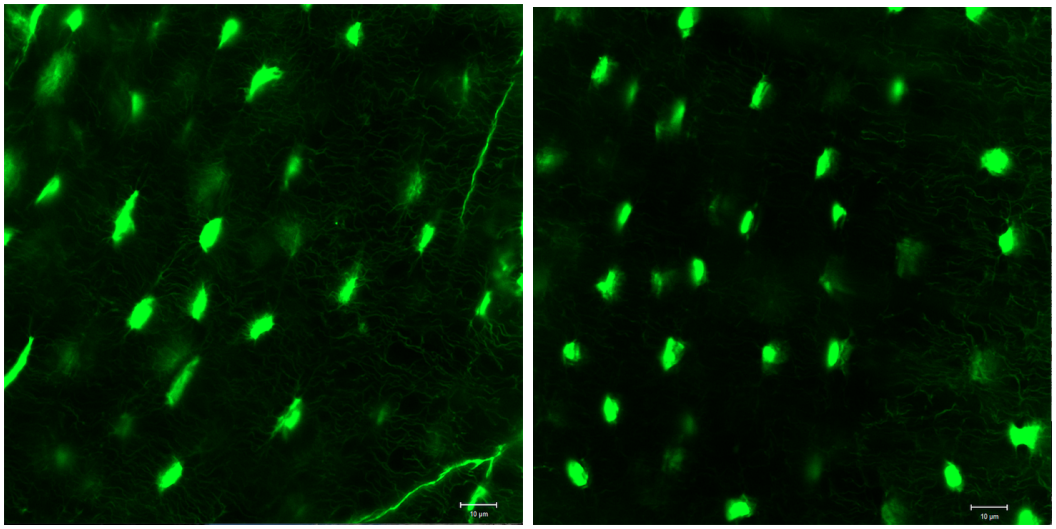


(a) (b)



(c) (d)

Fig. 4.4. Confocal z-stack slices to characterize the LCS in: (a) and (c) 20-week-old mice, and (b) and (d) 52-week-old mice; scale bar = 5 μm.



(a) (b)

Fig. 4.5. Confocal z-stack slices to determine volumetric lacunar density: (a) 20-week-old mouse (b) 52-week-old mouse; scale bar = 10 μm.

From reconstruction of the lacunae in Mimics, the average volumetric lacunar densities for the 20-week-old and 52-week-old mice are shown in Figure 4.6. Performing a Kruskal-Wallis H test, a statistical difference in volumetric density was determined between the two ages of mice (p=0.0463).

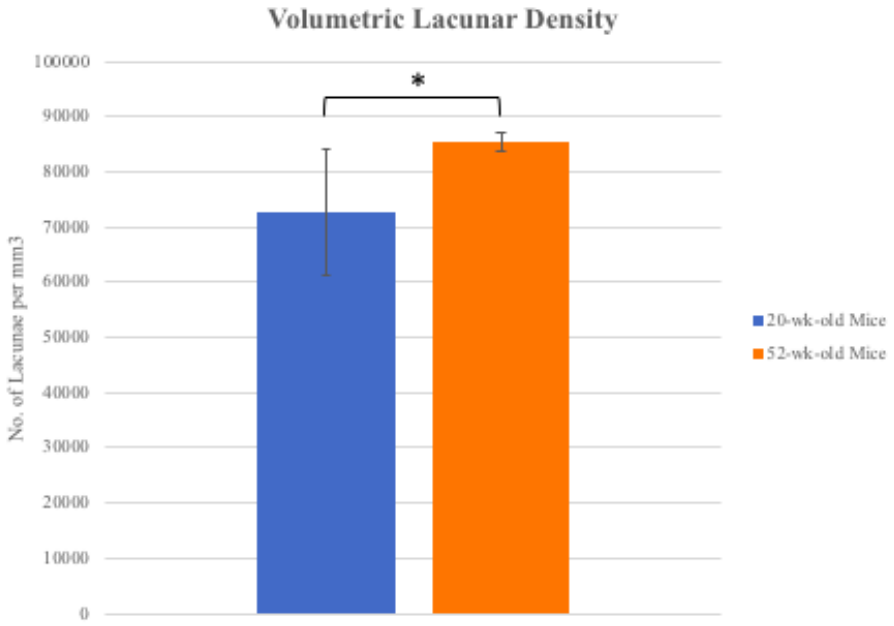


Fig. 4.6. Volumetric lacunar density for 20-week-old and 52-week-old mice (n=3). Statistical significance was found between the two groups of mice (p=0.0463); mean ± stdev.

4.3.2 LCS 3D Reconstruction

Using the validation dataset, the overall accuracy of the code in segmenting out the LCS is 57.6%. Specifically, for segmentation of the lacunae, its accuracy is 81.7% while for the canaliculi, it is 47.6%. Slice 10 of this validation dataset, as shown in Figure 4.2, is again displayed in Figure 4.7, following implementation of the validation procedure in Matlab. The Matlab segmented lacunae and canaliculi are shown in (a) and (c), respectively. Images (b) and (d) show the overlap of the lacunae and canaliculi, respectively, from the validation dataset (pink) and the Matlab segmentation (light blue). Image (e) shows the overlay of the lacunae and canaliculi (images (b) and (d)) Overlapping regions between the Mimics and Matlab datasets are shown in white.

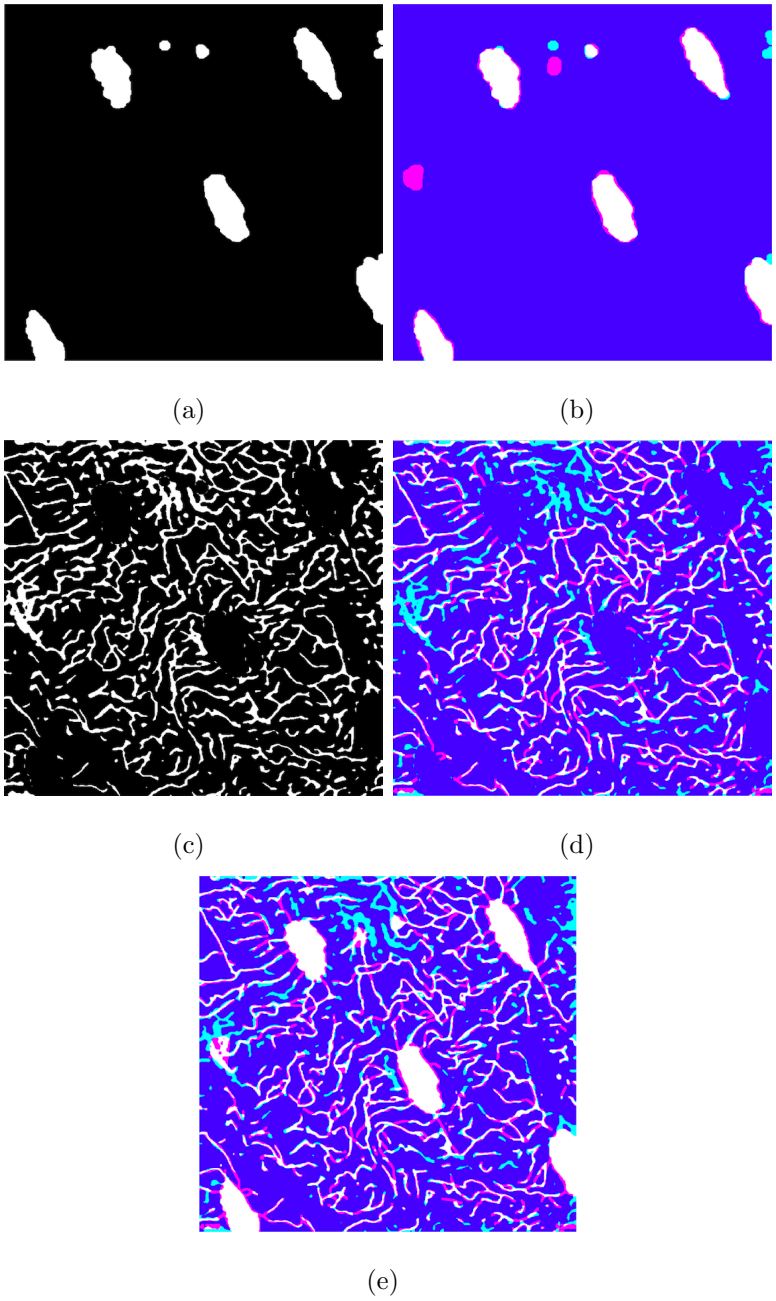
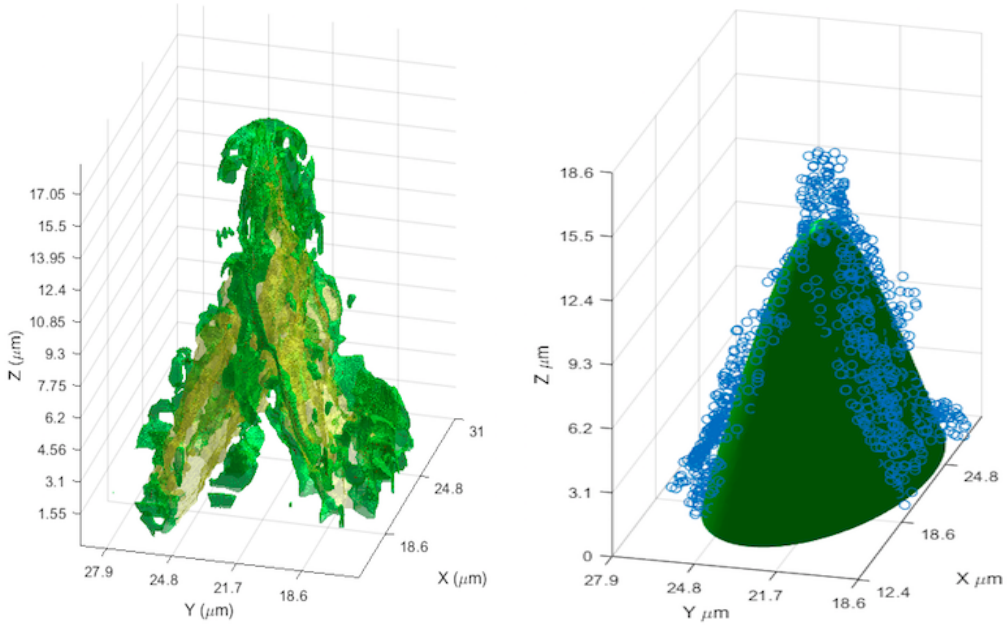


Fig. 4.7. Slice 10 of the validation dataset after processing in Matlab. (a) Processed lacunae in Matlab (b) Overlay of lacunae from the validation dataset (Mimics) and processed segmentation (Matlab). (c) Processed canaliculi in Matlab. (d) Overlay of canaliculi from the validation dataset (Mimics) and processed segmentation (Matlab). (e) Overlay of (b) and (d). Validation dataset (Mimics) = pink, processed dataset (Matlab) = light blue, overlapping region = white.

A total of 18 z-stacks were reconstructed using the custom Matlab code: 3 mice per age group (20-week-old and 52-week-old) and 3 lacunae per mouse imaged. During 3D reconstruction of individual lacuna, numerous lacunae were observed to not be ellipsoidal in shape but actually appeared to be separating or branching in the longitudinal direction, as shown in Figure 4.8(a). The yellow volume is the reconstruction of the single lacuna that appears to be splitting in the z-direction, where the top of the sample is the most proximal. The green surrounding tendrils are the reconstructed canaliculi that directly connect to the lacunae. Figure 4.8(b) shows the ellipsoid (green) that attempted to be fit to the shape of the lacuna, where the blue data points indicate the portion of the segmented lacuna that lies outside the fitted ellipsoid. From this, it becomes apparent that an ellipsoid is not an appropriate approximation when this morphology is observed. On the other hand, Figure 4.9(a) shows the reconstruction of a lacuna that is more accurately approximated as an ellipsoid (Figure 4.9(b)).

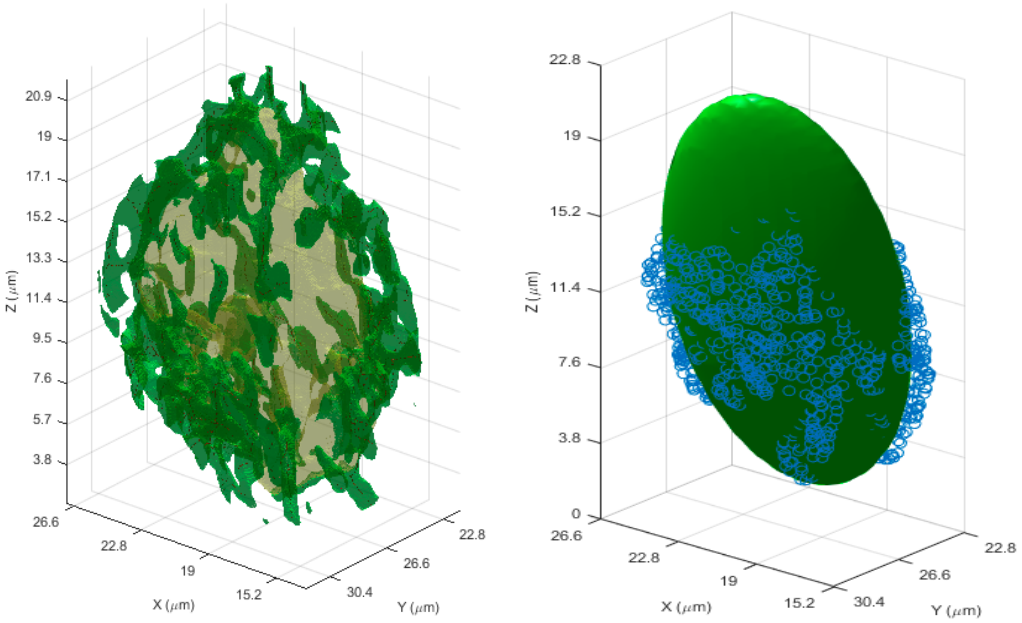
The lacunar splitting was observed in 6 out of the 9 z-stacks for the 20-week-old mice and 4 out of the 9 z-stacks for the 52-week-old mice. This splitting, or "butterfly-effect" could potentially be a distortion of the true lacuna during imaging due to the objective coming into contact with the coverslip. Thus, all of the split lacunae were removed from subsequent analyses. Figure 4.10 shows the reconstruction of the 3 ideal LCS z-stacks for the 20-week-old mice while Figure 4.11 shows the reconstruction of the 5 ideal LCS z-stacks for the 52-week-old mice. Additionally, the custom Matlab code produced full reconstructions of the lacuna of interest and all surrounding canaliculi, as shown in Figure 4.12 for the 20-week-old mice and Figure 4.13 for the 52-week-old mice.



(a) (b)

Fig. 4.8. Reconstruction of a branching ellipsoid. (a) Reconstruction of the confocal z-stack for a sample that contains a lacuna that splits in the z-direction. The yellow volume is the reconstructed lacuna volume and the green branches are the reconstructed canaliculi that directly connect to the lacuna. (b) Approximated ellipsoidal fit of the lacuna in (a). The green shape is the fitted ellipsoid and the blue data points indicate the portion of the segmented lacunae that lie outside the fitted shape.

Because of the splitting lacunae, sample sizes were decreased, 3 z-stacks for the 20-week-old mice and 5 z-stacks for the 52-week-old mice. If the z-stacks are treated as independent of the mouse, higher sample sizes can be utilized. Genetically, all of the mice are identical. In addition, each lacuna in the mice undergo their own individual modeling and remodeling processes to grow and adapt; thus, each lacuna (or z-stack sample) can be considered independent of the mouse. In order to quantitatively support this - whether all of the samples in each age group could be treated independently - Kruskal-Wallis H Tests (for the raw measurements and for the fitted ellipsoid measurements) were performed for each age group to determine if any depen-



(a) (b)

Fig. 4.9. Reconstruction of an ideal ellipsoid. (a) Reconstruction of a confocal z-stack that captures a full lacuna. The yellow volume is the reconstructed lacuna volume and the green branches are the reconstructed canaliculi that directly connect to the lacuna. (b) Ellipsoidal fit of the lacuna in (a). The green shape is the fitted ellipsoid and the blue data points indicate the portion of the segmented lacunae that lie outside the fitted shape.

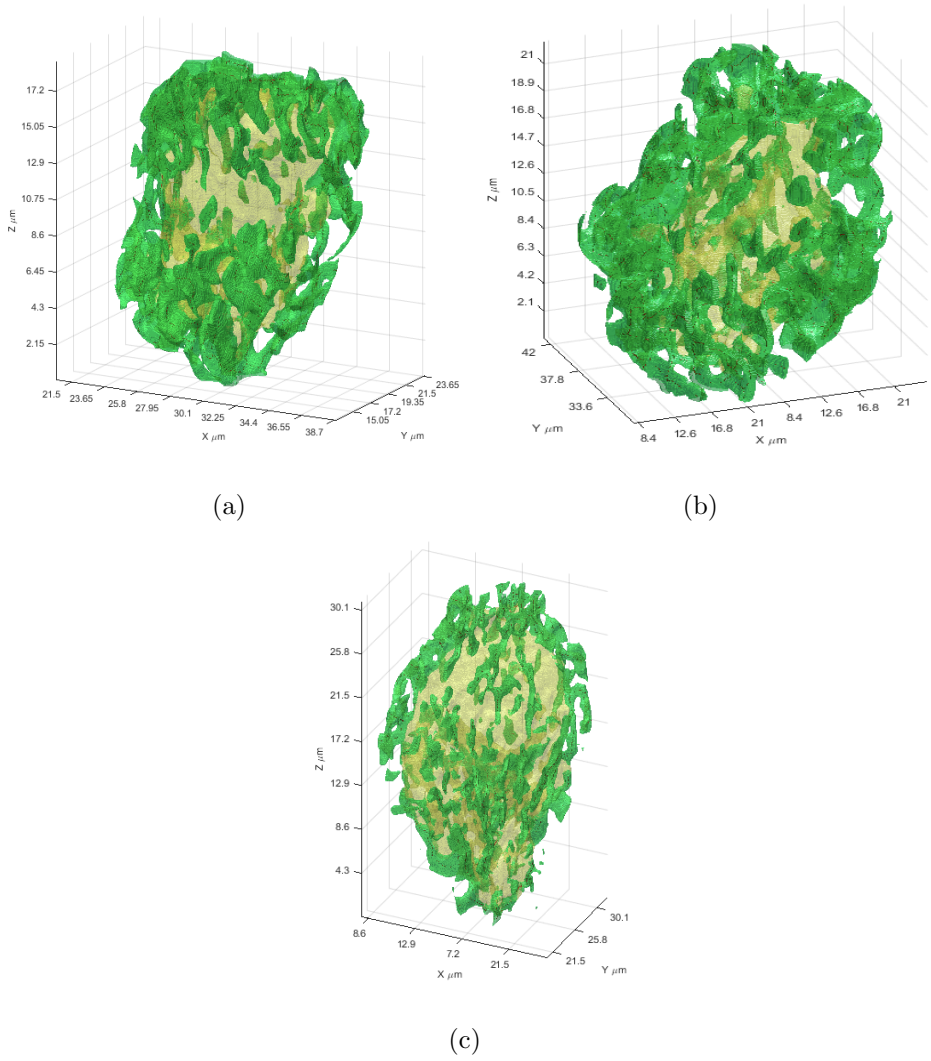


Fig. 4.10. Reconstructions of the 3 ideal lacunae (yellow) and connecting canaliculi (green) for the 20-week-old mice.

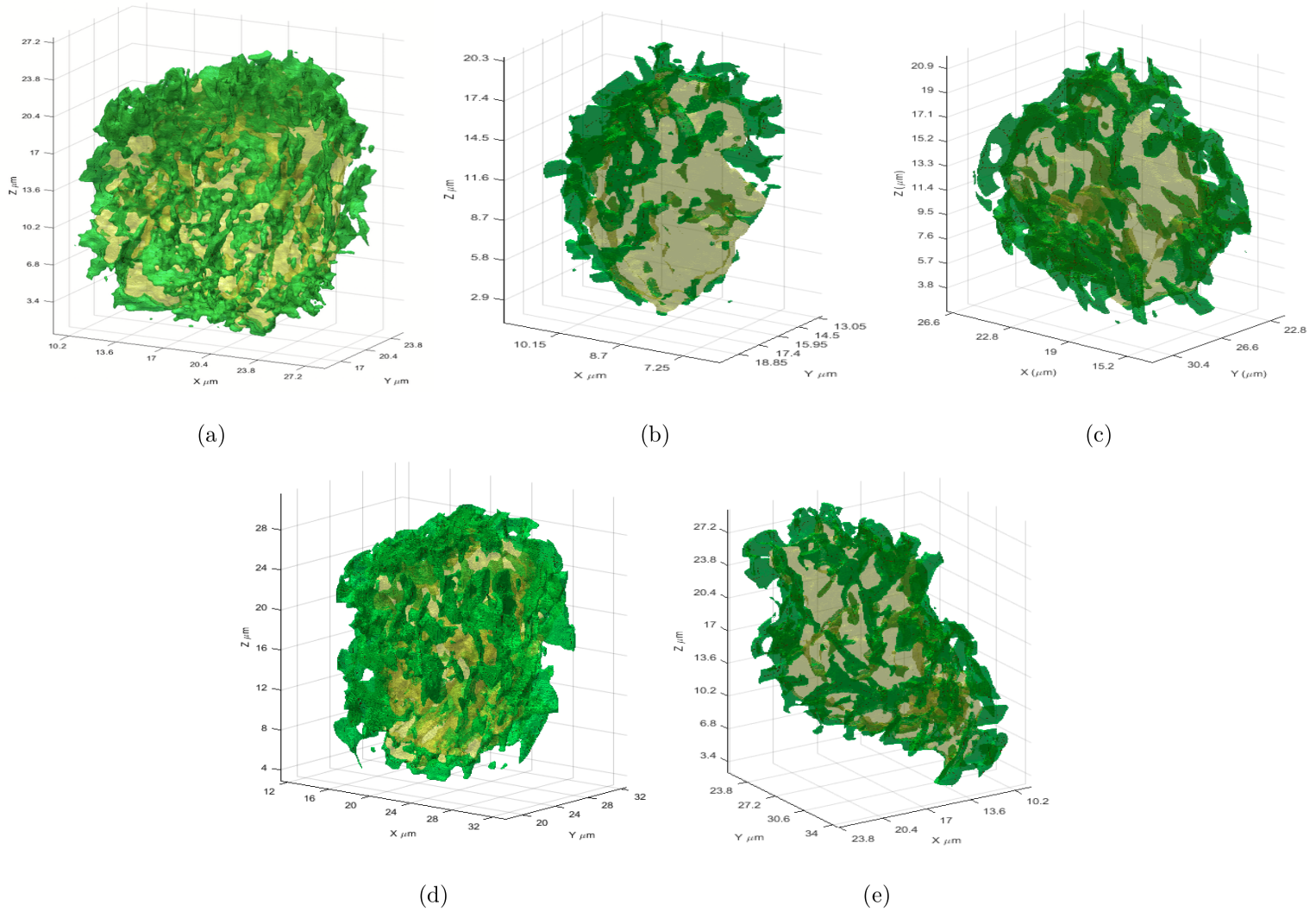
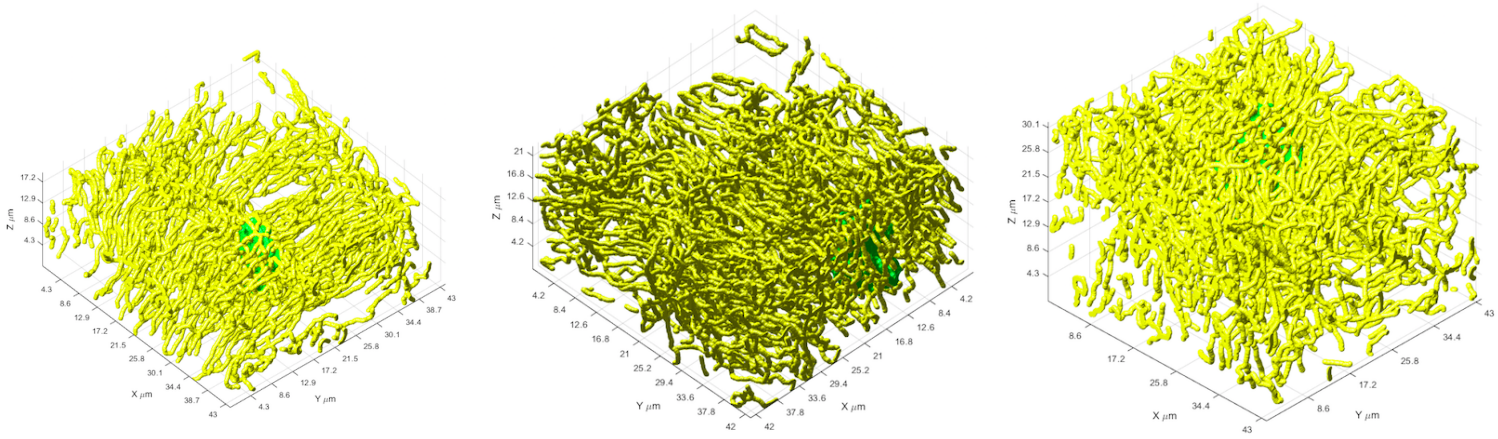


Fig. 4.11. Reconstructions of the 5 ideal lacunae (yellow) and connecting canaliculi (green) for the 52-week-old mice.



(a) (b) (c)

Fig. 4.12. Full reconstruction of a lacuna of interest (green) and the surrounding reconstructed canaliculi (yellow) from 20-week-old mice.

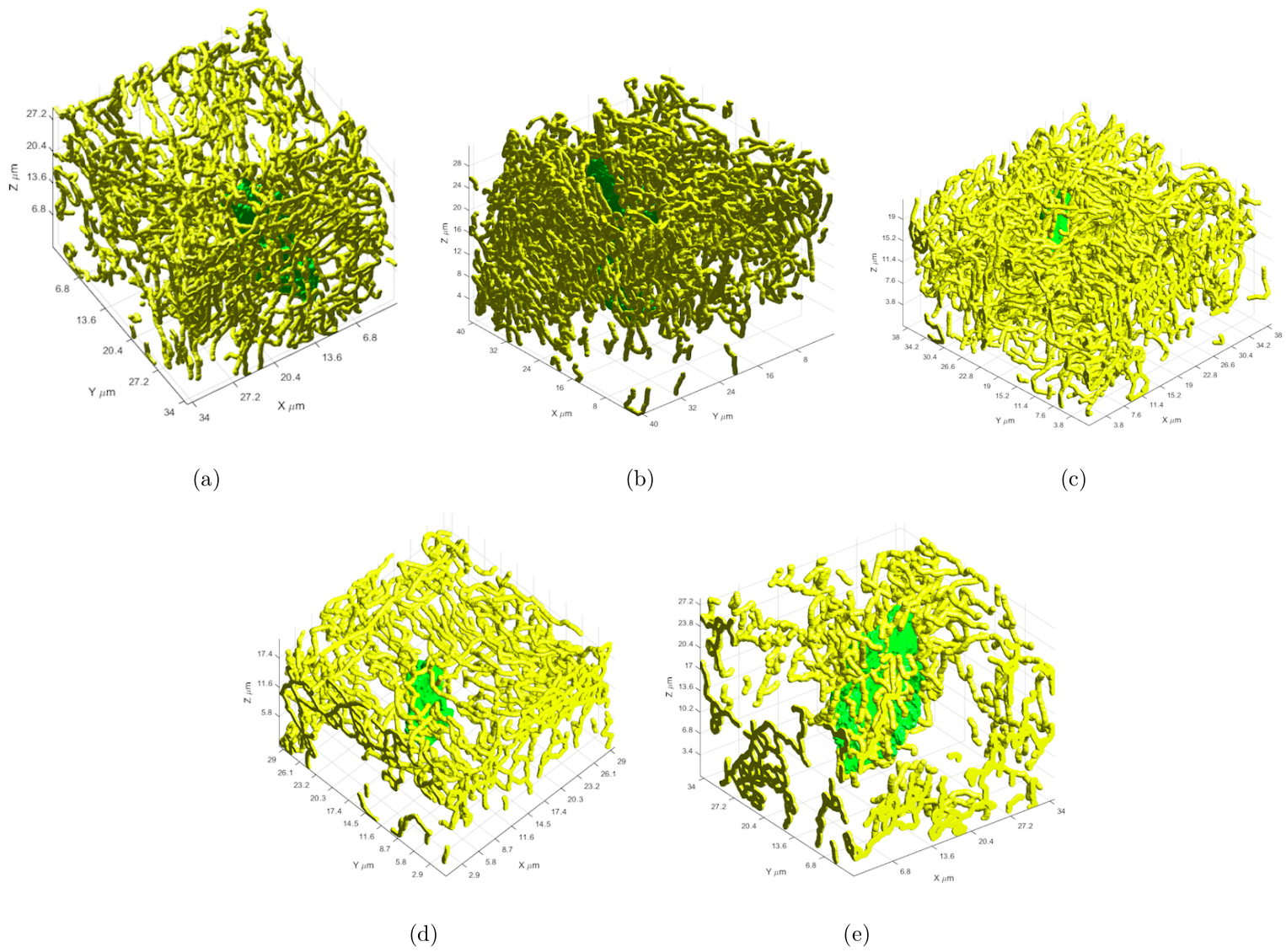


Fig. 4.13. Full reconstruction of a lacuna of interest (green) and the surrounding reconstructed canaliculi (yellow) from 52-week-old mice.

dence exists between the mouse and its measurements, using Bonferroni's correction to account for multiple comparisons. For both age groups, each raw measurement did not depend upon the mouse from which the sample came ($p>0.01$). Similarly, each ellipsoid fitted measurement also did not depend upon the mouse ($p>0.00625$). Thus, the samples in each age group were considered to be independent of mouse when performing the following analyses.

In regard to characterization of the raw reconstructed z-stacks, the average canalicular (a) diameter, (b) length, and (c) porosity were determined for each age group (n=3 for 20-week-old, n=5 for 52-week-old), as shown in Figure 4.14. Using Bonferroni-corrected Kruskal-Wallis H Tests, no statistical differences were detected between age group and any of the measures. Figure 4.15 shows the volumes for both the lacunae and for the fitted ellipsoids and also the connecting canalicular density (with respect to lacunar volume). No statistical differences were found for either the lacunar volume or the canalicular density ($p> 0.01$) or the ellipsoid volume ($p>0.00625$) with respect to age group (n=3 for 20-week-old, n=5 for 52-week-old).

The surface area of the ellipsoid and its three diameters of are shown in Figure 4.16 for each age group, where 'LacDia1' is the major axis diameter and 'LacDia2' and 'LacDia3' are the two minor axis diameters. Additionally, Figure 4.17 shows the lacunar ellipsoid (a) sphericity, (b) oblateness, and (c) orientation of the major diameter with respect to the z-axis. A sphericity measure of 1 indicates that the ellipsoid is perfectly spherical while an oblateness measures of -1 and +1 indicates that the ellipsoid is either perfectly prolate (rod-shaped) or perfectly oblate (plate-shaped), respectively [43]. Using Bonferroni-corrected Kruskal-Wallis H Tests Tests, no statistical differences were detected between age group and ellipsoid parameters ($p > 0.00625$).

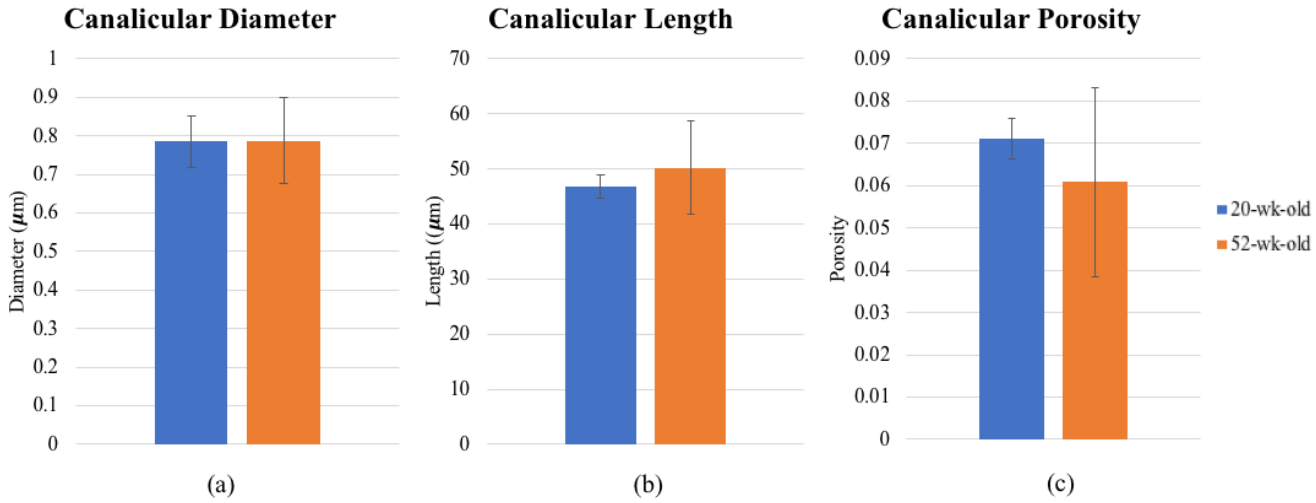


Fig. 4.14. Characterization of canalicular (a) diameter, (b) length, and (c) porosity in 20-week-old and 52-week-old female mouse femora (n=3 for 20-week-old, n=5 for 52-week-old). Using Bonferroni-corrected Kruskal-Wallis H Tests, no statistical differences were determined between age group and the measured parameters (p-value>0.01); mean ± stdev.

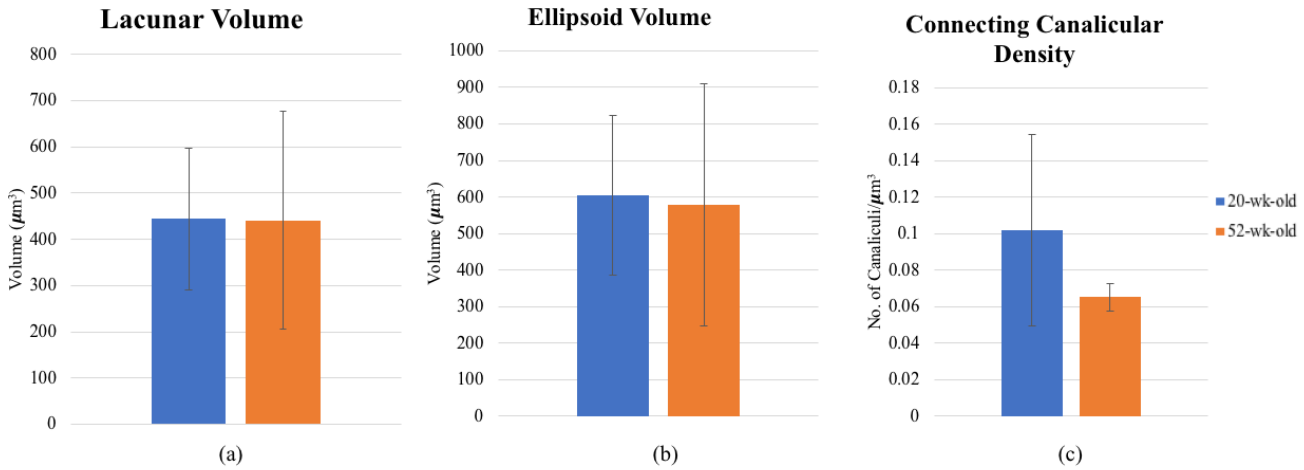


Fig. 4.15. Characterization of true and approximate lacunar volumes and the connecting canalicular density. (a) Volume of segmented lacunae and (b) volume of fitted ellipsoid in 20-week-old and 52-week-old female mouse femora. (c) Density of connecting canaliculi to the lacunar volume. Using Bonferroni corrected Kruskal-Wallis H Tests, no statistical differences were found for either the lacunar volume and its connecting canalicular density ($p > 0.01$) or the ellipsoid volume ($p > 0.00625$) (n=3 for 20-week-old, n=5 for 52-week-old); mean ± stdev.

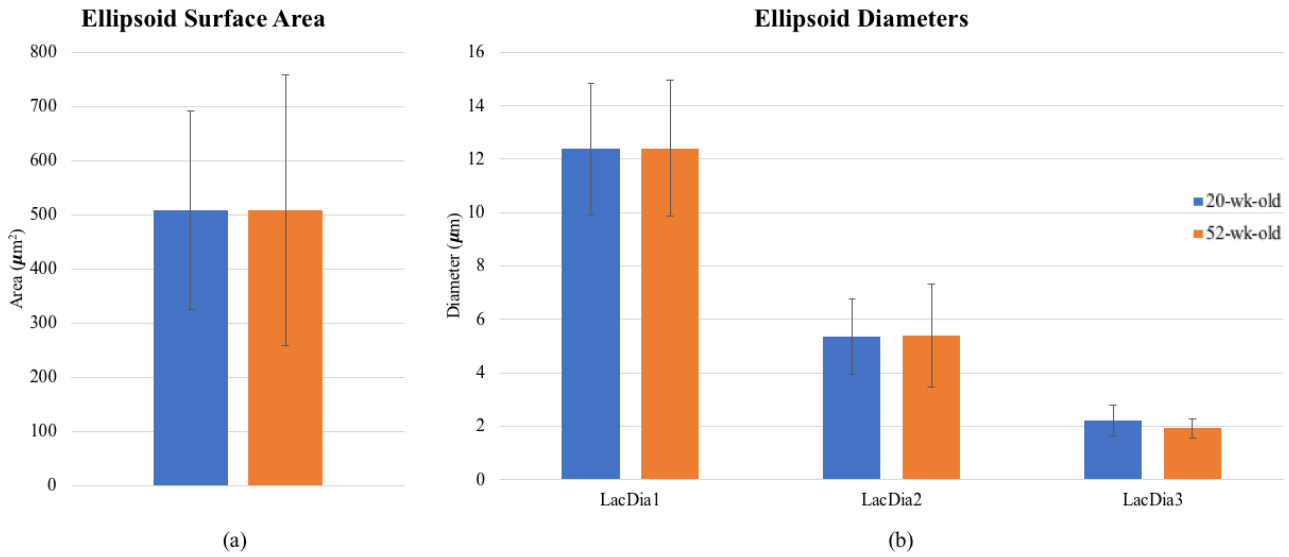


Fig. 4.16. (a) Surface area of fitted lacunar ellipsoids. (b) Lacunar ellipsoid diameters in 20-week-old and 52-week-old female mouse femora, where 'LacDia1' is the major diameter and 'LacDia2' and 'LacDia3' are the two minor diameters (n=3 for 20-week-old, n=5 for 52-week-old). Using Bonferroni-corrected Kruskal-Wallis H Tests, no statistical differences were determined between age and any of the lacunar diameters ($p > 0.0056$); mean $\pm$ stdev. Using Bonferroni-corrected Kruskal-Wallis H Tests, no statistical differences were determined ($p > 0.00625$); mean $\pm$ stdev.

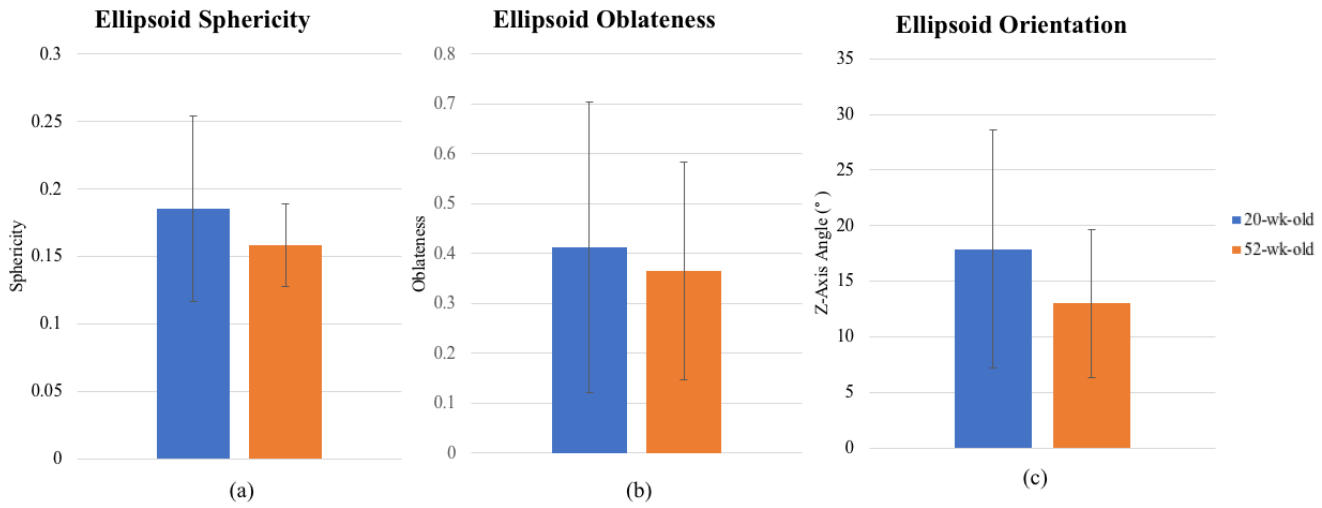


Fig. 4.17. Characterization of lacunar ellipsoid (a) sphericity, (b) oblateness, and (c) orientation in 20-week-old and 52-week-old female mouse femora (n=3 for 20-week-old, n=5 for 52-week-old). A sphericity measure of 1 indicates that the ellipsoid is perfectly spherical while an oblateness measures of -1 and +1 indicates that the ellipsoid is either perfectly prolate (rod-shaped) or perfectly oblate (plate-shaped), respectively [43]. Using Bonferroni-corrected Kruskal-Wallis H Tests, no statistical differences were determined between age and any of these measured characteristics ($p > 0.00625$); mean $\pm$ stdev.

4.4 Discussion

In order to remain consistent for each z-stack, the same CLSM settings were utilized for each z-stack for LCS characterization and also for volumetric lacunar density imaging, respectively. While optimal when comparing across the samples, the quality of staining, thus, greatly influenced the CLSM and reconstruction results. From the results shown in Figure 4.4, the staining and imaging quality appears to be suboptimal for the 52-week-old samples as compared to the 20-week-old mice. Presumably, this difference in staining and imaging resulted from the sample preparation. While all samples were ground down to an approximate thickness of $150\mu m$, there was no way to verify the overall levelness and flatness of the cross-sections. The levelness would influence the orientation at which the LCS is being viewed, impacting whether it is a true transverse cross-section. Sectioning of the samples would need to better take this into account. In addition, flatness of the sample greatly affects the surface staining of the sample. An uneven surface, or scratches produced by the grinding the sample, led to areas of fluorescent streaking, impeding the quality of the z-stacks. Utilization of a finer grit, such as 1200 or 1500 grit paper, would aid in a smoother surface and eliminate the "butterfly effect" that was observed.

While the quality of the z-stacks for LCS characterization varied due to staining, the intensity of staining for the z-stacks obtained for volumetric lacunar density analysis were more uniform. As the lacunae are several orders of magnitude larger than the canaliculi, the penetration of the Alexa 488 was much less of an issue or concern. Using Mimics, the average volumetric lacunar densities for the 20-week-old and 52-week-old mice were determined to be approximately 72,600 $lacunae/mm^3$ and 85,400 $lacunae/mm^3$, respectively. These averages fall within the typical range reported in the literature (40,0000 $lacunae/mm^3$ - 90,000 $lacunae/mm^3$) [31] [37] [41] [91] and were found to be statistically significant using a Kruskal-Wallis H test. However, previous studies have shown that areal lacunar density decreases as a function of age in humans [92] [93]. Additionally, a decrease in volumetric lacunar density in mice

with respect to age has been reported, when comparing 20-week-old and 88-week-old mice [94], as well as a decrease in areal lacunar density between 15-week-old and 32-week-old mice [95]. While these other studies include either young mice (15-week-old) or old mice (88-week-old), the comparison made here was with young adult (20-week-old mice) and adult (52-week-old) mice; moreover, the limited sample size of 3 for each age group needs to be increased in order to determine if this significant change is robust and anatomically correct.

With regards to the LCS 3D reconstruction, the initial validation on the automatic 2D segmentation resulted in a 81.7%, 47.6%, and 57.6% accuracies in segmenting out lacunae, canaliculi, and the LCS as a whole, respectively. While the canaliculi accuracy is lower than desired, the code currently is able to quickly segment the LCS in approximately 1 - 2 hours. More rigorous methods could be employed to reach a higher accuracy, but the computation time would dramatically increase as well as the need for more advanced computing resources to manage the data. Additionally, only one validation dataset was created and tested due to the time demands of manual reconstruction. In order to more accurately validate the code, more than one validation dataset needs to be created, by different users to avoid bias. Moreover, these additional datasets would include higher magnifications to more closely model those which were processed to characterize the LCS. Thus, given the single validation dataset and time and computing constraints, the custom reconstruction code's accuracies were deemed to be acceptable.

While the lacunae and canaliculi have been manually segmented and reconstructed [5] [38] [35] [37] [44], this is the first time, to the author's knowledge, that a custom, automated method has been utilized to reconstruct both the lacunae *and* the canaliculi. From these reconstructions, the dense network of the LCS can be visualized and utilized for predictive modeling, such as finite element analyses. In addition to the three-dimensional reconstruction, key characteristics of the structure of LCS were obtained. To illustrate the capabilities of the reconstruction code, 20-week-old and 52-week-old femora metaphyseal sections were obtained from female mice. Table 4.1

displays the measured values that were obtained during this study for the 20-week-old and 52-week-old mice as well as the range for each variable that has been previously reported. While the diameter of canaliculi have been reported to range from 0.095 - 0.55 μm [35] [37], the average canalicular diameter for the age groups in this study ranged from 0.78 - 0.79 μm. This apparent overestimation of the canalicular diameter could explain the 47.6% accuracy of the code in segmenting out the canaliculi. Using the previously mentioned techniques to improve sample preparation and staining would allow for an enhanced signal-to-noise ratio when imaging. The canalicular length and porosity and the lacunar volume for both age groups fall within the ranges found. For the canalicular density, the 52-week-old mice appear to have a lower number of connecting canaliculi than previously published values. This may tie back into the fact that these older sections did not stain as well as the younger bone.

Finally, an ellipsoid was fit to the reconstructed lacunae to approximate characteristics regarding its shape. Table 4.2 summarizes the data obtained here for the fitted ellipsoid as well as the ranges for these values that have been reported in the literature. The ellipsoids produced in this study were more prolate and less spherical, for both age groups, than those previously reported [43] [96] and more aligned with the z-axis [96]. The diameters of the major and minor axes of the ellipsoid are within or fall just outside the ranges for typical ellipsoid fits [31] [32] [33] [42] [96]. While the volume of the ellipsoid agrees with previous findings [43], it appears to overestimate the actual volume of the lacunae. For both age groups, the average ellipsoid volume overestimated the actual average volume of the lacunae by approximately 150 μm. Thus, while an ellipsoid may be a decent approximation of the lacunar morphology, these results indicate that it may not always be the most accurate.

Table 4.1.
Measured parameters of the LCS using the custom Matlab reconstruction code (n=3 for 20-week-old mice, n=5 for 52-week-old mice) compared to ranges previously reported within literature.

	Measured Values		**Reported Values**			
	Age	*Mean ± stdev*	*Low*	*Method*	*High*	*Method*
Canalicular Diameter (μm)	20	0.785 ± 0.066	0.095	FIB/SEM, murine femora [35]	0.553	CLSM, rat tibiae [37]
	52	0.788 ± 0.111				
Canalicular Length (μm)	20	46.875 ± 2.142	23	CLSM, murine tibiae [97]	50	CLSM, murine tibiae [97]
	52	50.196 ± 8.441				
Canalicular Porosity	20	0.071 ± 0.005	0.007	FIB/SEM, murine femora [35]	0.14	CLSM, rat tibiae [37]
	52	0.061 ± 0.022				
Lacunar Volume (μm^3)	20	443.968 ± 153.410	290	SR-μCT, human femur [31]	455	CLSM, human hip [40]
	52	441.117 ± 234.813				
Canalicular Density (Number/μm^3)	20	0.102 ± 0.053; (52.667 ± 25.007 canaliculi/lacuna)	49.7	CLSM, murine calvaria [98]	83.9	CLSM, rat tibiae [37]
	52	0.065 ± 0.008; (28.400 ± 15.060 canaliculi/lacuna)				

Table 4.2.
Measured parameters of the fitted ellipsoid to the a lacuna of interest using the custom Matlab reconstruction code (n=3 for 20-week-old mice, n=5 for 52-week-old mice) compared to ranges previously reported within literature.

	Measured Values		**Reported Values**			
	Age	*Mean stdev*	*Low*	*Method*	*High*	*Method*
Ellipsoid Diameter (major) (μm)	20	12.378 ± 2.451	8.96	SR-μCT, human femur [31]	28.7	CLSM, murine fibula [32]
	52	12.405 ± 2.546				
Ellipsoid Diameter (large minor) (μm)	20	5.358 ± 1.430	5.51	SR-μCT, rat femora [96]	8.86	SR-μCT, human femora [42]
	52	5.400 ± 1.930				
Ellipsoid Diameter (small minor) (μm)	20	2.216 ± 0.567	2.00	CLSM, chick calveria [33]	5.16	SR-μCT, human femora [42]
	52	1.922 ± 0.672				
Ellipsoid Sphericity	20	0.185 ± 0.068	0.283	CLSM, murine tibiae [43]	0.790	SR-μCT, rat femora [96]
	52	0.158 ± 0.031				
Ellipsoid Oblateness	20	0.411 ± 0.292	-0.316	CLSM, murine tibiae [43]	0.080	SR-μCT, rat femora [96]
	52	0.365 ± 0.218				
Ellipsoid Surface Area (μm^2)	20	509.166 ± 183.769	195	CLSM, murine tibiae [43]	460	CLSM, murine tibiae [43]
	52	509.085 ± 250.514				
Ellipsoid Volume (μm^3)	20	604.633 ± 218.021	200	CLSM, murine tibiae [43]	750	CLSM, murine tibiae [43]
	52	578.603 ± 330.213				
Ellipsoid Orientation (°)	20	17.879 ± 10.713	23	SR-μCT, rat femora [96]	37	SR-μCT, rat femora [96]
	52	13.017 ± 6.666				

5. CONCLUSIONS & FUTURE WORK

Mathematical and computational modeling are powerful tools to illustrate the flow profiles within the LCS and to also extrapolate conclusions regarding the mechanosensory capabilities of skeletal tissue. In order to implement such models, assumptions concerning the properties and characteristics are typically made to simplify the analysis. In particular, within the interstitial space of the LCS, a glycoclyx has been assumed to exist in various models. However, from this work, the flow profile within this interstitial space is extremely sensitive to the presence of such a glycocalyx. Its physical structure greatly impacts the permeability of fluid within this space, which can alter the flow. In fact, from the sensitivity analyses presented here, the flow profile can mimic Brinkman, Darcy, or Poiseuille flow depending on the structure of the glycocalyx. Elucidating the dominating flow profile will provide insight into the main mechanotransductive mechanisms that may be playing a role in bone modeling and remodeling. Thus, the nanostructure of the glycocalyx needs to be fully characterized. Predictive models need to modify the assumptions typically made regarding the presence and structure of the glycocalyx and enable it to be a variable. Moreover, this mixture theory model should be expanded to additional cases, such as compression-driven and bending-driven flow, which are more physiologically relevant scenarios when osteocytes are stimulated.

From this work, an automated segmentation and reconstruction code was developed to reconstruct the LCS of cortical bone. This enables quicker and less biased segmentation of the lacunae and canaliculi for more accurate characterization of the LCS, which can then be utilized with computational modeling for fluid analyses. While the code presents promising results, additional validation datasets need to be produced and run in order to improve its segmentation accuracy. Furthermore, improved sample preparation techniques and a smaller z-step, perhaps even imaging on

longitudinal sections, will produce more rigorous results due to noise reduction from staining and less uncertainty between slices. Enhanced staining techniques may also allow for an increased probability in obtaining full lacunae z-stacks. These refinements would further more accurate modeling in determining the main mechanotransductive signal(s) involved in bone modeling and remodeling.

REFERENCES

REFERENCES

[1] M. L. Knothe Tate, ""Whither flows the fluid in bone?" An osteocyte's perspective," *Journal of Biomechanics*, vol. 36, no. 10, pp. 1409–1424, 2003.

[2] L. Ren, P. Yang, Z. Wang, J. Zhang, C. Ding, and P. Shang, "Biomechanical and biophysical environment of bone from the macroscopic to the pericellular and molecular level," *Journal of the Mechanical Behavior of Biomedical Materials*, vol. 50, pp. 104–122, 2015.

[3] S. P. Fritton and S. Weinbaum, "Fluid and solute transport in bone: Flow-induced mechanotransduction." *Annual Review of Fluid Mechanics*, vol. 41, pp. 347–374, 2009.

[4] D. B. Burr and M. R. Allen, Eds., *Basic and Applied Bone Biology*. Waltham: Academic Press, 2009.

[5] P. Varga, B. Hesse, M. Langer, S. Schrof, N. Männicke, H. Suhonen, A. Pacureanu, D. Pahr, F. Peyrin, and K. Raum, "Synchrotron X-ray phase nano-tomography-based analysis of the lacunarcanalicular network morphology and its relation to the strains experienced by osteocytes in situ as predicted by case-specific finite element analysis," *Biomechanics and Modeling in Mechanobiology*, vol. 14, no. 2, pp. 267–282, 2015.

[6] H. Kamioka, Y. Kameo, Y. Imai, A. D. Bakker, R. G. Bacabac, N. Yamada, A. Takaoka, T. Yamashiro, T. Adachi, and J. Klein-Nulend, "Microscale fluid flow analysis in a human osteocyte canaliculus using a realistic high-resolution image-based three-dimensional model," *Integrative Biology*, vol. 4, no. 10, p. 1198, 2012.

[7] E. J. Anderson, S. M. Kreuzer, O. Small, and M. L. Knothe Tate, "Pairing computational and scaled physical models to determine permeability as a measure of cellular communication in micro- and nano-scale pericellular spaces," *Microfluidics and Nanofluidics*, vol. 4, no. 3, pp. 193–204, 2008.

[8] J. Klein-Nulend, A. D. Bakker, R. G. Bacabac, A. Vatsa, and S. Weinbaum, "Mechanosensation and transduction in osteocytes," *Bone*, vol. 54, no. 2, pp. 182–190, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2012.10.013

[9] M. B. Schaffler, W. Y. Cheung, R. Majeska, and O. Kennedy, "Osteocytes: Master orchestrators of bone," *Calcified Tissue International*, vol. 94, no. 1, pp. 5–24, 2014.

[10] S. Weinbaum, S. Cowin, and Y. Zeng, "A model for the exciation of osteocytes by mechanical loading-induced bone fluid shear stresses," *Journal of Biomechanics*, vol. 27, no. 3, pp. 339–360, 1994.

[11] L. You, S. C. Cowin, M. B. Schaffler, and S. Weinbaum, "A model for strain amplification in the actin cytoskeleton of osteocytes due to fluid drag on pericellular matrix," *Journal of Biomechanics*, vol. 34, no. 11, pp. 1375–1386, 2001.

[12] S. C. Cowin and S. Weinbaum, "Strain amplification in the bone mechanosensory system," *The American Journal of the Medical Sciences*, vol. 316, no. 3, pp. 184–188, 1998.

[13] Y. Han, S. C. Cowin, M. B. Schaffler, and S. Weinbaum, "Mechanotransduction and strain amplification in osteocyte cell processes," *Proceedings of the National Academy of Sciences*, vol. 101, no. 47, pp. 16 689–16 694, 2004.

[14] S. Weinbaum, P. Guo, and L. You, "A new view of mechanotransduction and strain amplification in cells with microvilli and cell processes," *Biorheology*, vol. 38, no. 2-3, pp. 119–142, 2001.

[15] Y. Wang, L. M. McNamara, M. B. Schaffler, and S. Weinbaum, "Strain amplification and integrin based signaling in osteocytes," *Journal of Musculoskeletal Neuronal Interactions*, vol. 8, no. 4, pp. 332–334, 2008.

[16] G. C. Goulet, D. Coombe, R. J. Martinuzzi, and R. F. Zernicke, "Poroelastic evaluation of fluid movement through the lacunocanalicular system," *Annals of Biomedical Engineering*, vol. 37, no. 7, pp. 1390–1402, 2009.

[17] B. Wang, X. Lai, C. Price, W. R. Thompson, W. Li, T. R. Quabili, W. J. Tseng, X. S. Liu, H. Zhang, J. Pan, C. B. Kirn-Safran, M. C. Farach-Carson, and L. Wang, "Perlecan-containing pericellular matrix regulates solute transport and mechanosensing within the osteocyte lacunar-canalicular system," *Journal of Bone and Mineral Research*, vol. 29, no. 4, pp. 878–891, 2014.

[18] S. Gururaja, H. J. Kim, C. C. Swan, R. A. Brand, and R. S. Lakes, "Modeling deformation-induced fluid flow in cortical bone's canalicular-lacunar system," *Annals of Biomedical Engineering*, vol. 33, no. 1, pp. 7–25, 2005.

[19] S. W. Verbruggen, T. J. Vaughan, and L. M. Mcnamara, "Fluid flow in the osteocyte mechanical environment : a fluid structure interaction approach," pp. 85–97, 2014.

[20] J. Y. Rho, L. Kuhn-Spearing, and P. Zioupos, "Mechanical properties and the hierarchical structure of bone," *Medical Engineering and Physics*, vol. 20, no. 2, pp. 92–102, 1998.

[21] Y. Liu, D. Luo, and T. Wang, "Hierarchical structures of bone and bioinspired bone tissue engineering," *Small*, vol. 12, no. 34, pp. 4611–4632, 2016.

[22] M. Benalla, P. E. Palacio-Mancheno, S. P. Fritton, L. Cardoso, and S. C. Cowin, "Dynamic permeability of the lacunar-canalicular system in human cortical bone," *Biomechanics and Modeling in Mechanobiology*, vol. 13, no. 4, pp. 801–812, 2014.

[23] M. Benalla, L. Cardoso, and S. C. Cowin, "Analytical basis for the determination of the lacunar-canalicular permeability of bone using cyclic loading," *Biomechanics and Modeling in Mechanobiology*, vol. 11, no. 6, pp. 767–780, 2012.

[24] S. Cowin and D. Hegedus, "Bone remodeling I: theory of adaptive elasticity," *Journal of Elasticity*, vol. 6, no. 3, pp. 313–326, 1976.

[25] S. C. Cowin and L. Cardoso, "Blood and interstitial flow in the hierarchical pore space architecture of bone tissue," *Journal of Biomechanics*, vol. 48, no. 5, pp. 842–854, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jbiomech.2014.12.013

[26] A. Bernhard, P. Milovanovic, E. A. Zimmermann, M. Hahn, D. Djonic, M. Krause, S. Breer, K. Püschel, M. Djuric, M. Amling, and B. Busse, "Micromorphological properties of osteons reveal changes in cortical bone stability during aging, osteoporosis, and bisphosphonate treatment in women," *Osteoporosis International*, vol. 24, no. 10, pp. 2671–2680, 2013.

[27] Y. Lin and S. Xu, "AFM analysis of the lacunar-canalicular network in demineralized compact bone," *Journal of Microscopy*, vol. 241, no. 3, pp. 291–302, 2011.

[28] C. Ciani, S. B. Doty, and S. P. Fritton, "An effective histological staining process to visualize bone interstitial fluid space using confocal microscopy," *Bone*, vol. 44, no. 5, pp. 1015–1017, 2009. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2009.01.376

[29] L. Cardoso, S. P. Fritton, G. Gailani, M. Benalla, and S. C. Cowin, "Advances in assessment of bone porosity, permeability and interstitial fluid flow," *Journal of Biomechanics*, vol. 46, no. 2, pp. 253–265, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.jbiomech.2012.10.025

[30] Y. Carter, C. D. L. Thomas, J. G. Clement, and D. M. Cooper, "Femoral osteocyte lacunar density, volume and morphology in women across the lifespan," *Journal of Structural Biology*, vol. 183, no. 3, pp. 519–526, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047847713001846

[31] K. M. Hannah, C. D. L. Thomas, J. G. Clement, F. De Carlo, and A. G. Peele, "Bimodal distribution of osteocyte lacunar size in the human femoral cortex as revealed by micro-CT," *Bone*, vol. 47, no. 5, pp. 866–871, 2010. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/20691298 http://linkinghub.elsevier.com/retrieve/pii/S8756328210013906

[32] A. Vatsa, R. G. Breuls, C. M. Semeins, P. L. Salmon, T. H. Smit, and J. Klein-Nulend, "Osteocyte morphology in fibula and calvaria - Is there a role for mechanosensing?" *Bone*, vol. 43, no. 3, pp. 452–458, 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/18625577 http://linkinghub.elsevier.com/retrieve/pii/S8756328208000793

[33] Y. Sugawara, H. Kamioka, T. Honjo, K. I. Tezuka, and T. Takano-Yamamoto, "Three-dimensional reconstruction of chick calvarial osteocytes and their cell processes using confocal microscopy," *Bone*, vol. 36, no. 5, pp. 877–883, 2005.

[34] G. Marotti, "The original contributions of the scanning electron microscope to the knowledge of bone structure," in *Ultrastructure of Skeletal Tissues: Bone and Cartilage in Health and Disease*, E. Bonucci and P. M. Motta, Eds. Boston, MA: Springer US, 1990, pp. 19–39.

[35] P. Schneider, M. Meier, R. Wepf, and R. Müller, "Serial FIB/SEM imaging for quantitative 3D assessment of the osteocyte lacuno-canalicular network," *Bone*, vol. 49, no. 2, pp. 304–311, 2011. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2011.04.005

[36] L.-D. You, S. Weinbaum, S. C. Cowin, and M. B. Schaffler, "Ultrastructure of the osteocyte process and its pericellular matrix," *The Anatomical Record*, vol. 278A, no. 2, pp. 505–513, 2004. [Online]. Available: http://doi.wiley.com/10.1002/ar.a.20050

[37] D. Sharma, C. Ciani, P. A. R. Marin, J. D. Levy, S. B. Doty, and S. P. Fritton, "Alterations in the osteocyte lacunar-canalicular microenvironment due to estrogen deficiency," *Bone*, vol. 51, no. 3, pp. 488–497, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2012.05.014

[38] Y. Sugawara, H. Kamioka, Y. Ishihara, N. Fujisawa, N. Kawanabe, and T. Yamashiro, "The early mouse 3D osteocyte network in the presence and absence of mechanical loading," *Bone*, vol. 52, no. 1, pp. 189–196, 2013. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2012.09.033

[39] T. Beno, Y. J. Yoon, S. C. Cowin, and S. P. Fritton, "Estimation of bone permeability using accurate microstructural measurements," *Journal of Biomechanics*, vol. 39, no. 13, pp. 2378–2387, 2006.

[40] B. R. McCreadie, S. J. Hollister, M. B. Schaffler, and S. A. Goldstein, "Osteocyte lacuna size and shape in women with and without osteoporotic fracture," *Journal of Biomechanics*, vol. 37, no. 4, pp. 563–572, 2004.

[41] P. E. Palacio-Mancheno, A. I. Larriera, S. B. Doty, L. Cardoso, and S. P. Fritton, "3D assessment of cortical bone porosity and tissue mineral density using high-resolution μcT: Effects of resolution and threshold method," *Journal of Bone and Mineral Research*, vol. 29, no. 1, pp. 142–150, 2014.

[42] P. Dong, S. Haupert, B. Hesse, M. Langer, P. J. Gouttenoire, V. Bousson, and F. Peyrin, "3D osteocyte lacunar morphometric properties and distributions in human femoral cortical bone using synchrotron radiation micro-CT images," *Bone*, vol. 60, pp. 172–185, 2014. [Online]. Available: http://dx.doi.org/10.1016/j.bone.2013.12.008

[43] C. M. Heveran, A. Rauff, K. B. King, R. D. Carpenter, and V. L. Ferguson, "A new open-source tool for measuring 3D osteocyte lacunar geometries from confocal laser scanning microscopy reveals age-related changes to lacunar size and shape in cortical mouse bone," *Bone*, vol. 110, pp. 115–127, 2018. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S8756328218300188

[44] B. Hesse, P. Varga, M. Langer, A. Pacureanu, S. Schrof, A, H. Suhonen, P. Maurer, P. Cloetens, F. Peyrin, and K. Raum, "Canalicular network morphology is the major determinant of the spatial distribution of mass density in human bone tissue: Evidence by means of synchrotron radiation phase-contrast nano-CT," *Journal of Bone and Mineral Research*, vol. 30, no. 2, pp. 346–356, 2015.

[45] R. A. Brand, "Biographical sketch: Julius Wolff, 1836-1902." *Clinical orthopaedics and related research*, vol. 468, no. 4, pp. 1047–1049, 2010.

[46] C. Price, X. Zhou, W. Li, and L. Wang, "Real-time measurement of solute transport within the lacunar-canalicular system of mechanically loaded bone: Direct evidence for load-induced fluid flow," *Journal of Bone and Mineral Research*, vol. 26, no. 2, pp. 277–285, 2011.

[47] K. Piekarski and M. Munro, "Transport mechanism operating between blood supply and osteocytes in long bones," *Nature*, vol. 269, no. 5623, pp. 80–82, 1977.

[48] Y. X. Qin, T. Kaplan, A. Saldanha, and C. Rubin, "Fluid pressure gradients, arising from oscillations in intramedullary pressure, is correlated with the formation of bone and inhibition of intracortical porosity," *Journal of Biomechanics*, vol. 36, no. 10, pp. 1427–1437, 2003.

[49] C. Halperin, S. Mutchnik, A. Agronin, M. Molotskii, P. Urenski, M. Salai, and G. Rosenman, "Piezoelectric Effect in Human Bones Studied in Nanometer Scale," *Nano Letters*, vol. 4, no. 7, pp. 1253–1256, 7 2004. [Online]. Available: http://pubs.acs.org/doi/abs/10.1021/nl049453i

[50] D. Fu, Z. Hou, Q.-H. Qin, L. Xu, and Y. Zeng, "Influence of shear stress on behaviors of piezoelectric voltages in bone," *Journal of Applied Biomechanics*, vol. 28, no. 4, pp. 387–393, 2012.

[51] A. C. Ahn and A. J. Grodzinsky, "Relevance of collagen piezoelectricity to Wolff's Law: A critical review," *Medical Engineering & Physics*, vol. 31, no. 7, pp. 733–741, 2009.

[52] S. Cowin, S. Weinbaum, and Y. Zeng, "A case for bone canaliculi as the anatomical site of strain generated potentials," *Journal of Biomechanics*, vol. 28, no. 11, pp. 1281–1297, 1995.

[53] S. P. Fritton, K. J. McLeod, and C. T. Rubin, "Quantifying the strain history of bone: Spatial uniformity and self-similarity of low-magnitude strains," *Journal of Biomechanics*, vol. 33, no. 3, pp. 317–325, 2000.

[54] A. A. Biewener and J. E. A. Bertram, "SKELETAL STRAIN PATTERNS IN RELATION TO EXERCISE TRAINING DURING GROWTH," *J. exp. Biol*, vol. 185, pp. 51–69, 1993. [Online]. Available: http://jeb.biologists.org/content/jexbio/185/1/51.full.pdf

[55] C. T. Rubin and L. E. Lanyon, "Dynamic strain similarity in vertebrates; an alternative to allometric limb bone scaling." *Journal of theoretical biology*, vol. 107, no. 2, pp. 321–7, 3 1984. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/6717041

[56] S. Scheiner, P. Pivonka, and C. Hellmich, "Poromicromechanics reveals that physiological bone strains induce osteocyte-stimulating lacunar pressure," *Biomechanics and Modeling in Mechanobiology*, vol. 15, no. 1, pp. 9–28, 2016.

[57] S. C. Cowin, "Bone Poroelasticity," vol. 32, pp. 1–25, 1999.

[58] S. Cowin, "Summary of discussions on fluid infiltrated geomaterials," in *Mechanics of Geomaterials*, Z. Bazant, Ed. New York: Wiley, 1985, pp. 581–582.

[59] X. Wu, N. Wang, Z. Wang, W. Yu, Y. Wang, Y. Guo, and W. Chen, "Mathematically modeling fluid flow and fluid shear stress in the canaliculi of a loaded osteon," *BioMedical Engineering OnLine*, vol. 15, no. S2, p. 149, 2016. [Online]. Available: http://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-016-0267-x

[60] E. J. Anderson, S. Kaliyamoorthy, J. I. D. Alexander, and M. L. K. Tate, "Nano-microscale models of periosteocytic flow show differences in stresses imparted to cell body and processes," *Ann. Biomed. Eng.*, vol. 33, no. 1, pp. 52–62, 2005.

[61] L. Wang, C. Ciani, S. B. Doty, and S. P. Fritton, "Delineating bone's interstitial fluid pathway in vivo," *Bone*, vol. 34, no. 3, pp. 499–509, 2004.

[62] L. M. McNamara, R. J. Majeska, S. Weinbaum, V. Friedrich, and M. B. Schaffler, "Attachment of osteocyte cell processes to the bone matrix," *Anatomical Record*, vol. 292, no. 3, pp. 355–363, 2009.

[63] Y. Wang, L. M. McNamara, M. B. Schaffler, and S. Weinbaum, "A model for the role of integrins in flow-induced mechanotransduction in osteocytes," *Proceedings of the National Academy of Sciences*, vol. 104, no. 40, pp. 15 941–15 946, 2007.

[64] A. Miyauchi, M. Gotoh, H. Kamioka, K. Notoya, H. Sekiya, Y. Takagi, Y. Yoshimoto, H. Ishikawa, K. Chihara, T. Takano-Yamamoto, T. Fujita, and Y. Mikuni-Takagaki, "αVβ3 Integrin ligands enhance volume-sensitive calcium influx in mechanically stretched osteocytes," *Journal of Bone and Mineral Metabolism*, vol. 24, no. 6, pp. 498–504, 2006.

[65] K. J. Noonan, J. W. Stevens, R. Tammi, M. Tammi, J. A. Hernandez, and R. J. Midura, "Spatial distribution of CD44 and hyaluronan in the proximal tibia of the growing rat," *Journal of Orthopaedic Research*, vol. 14, no. 4, pp. 573–581, 1996.

[66] M. M. Schuff, J. P. Gore, and E. A. Nauman, "A mixture theory model of fluid and solute transport in the microvasculature of normal and malignant tissues. I. Theory," *Journal of Mathematical Biology*, vol. 66, no. 6, pp. 1179–1207, 2013.

[67] S. C. Cotter, "A general method of confounding for symmetrical factorial experiments," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 36, no. 2, pp. 267–276, 1974.

[68] E. A. Sander, D. A. Shimko, K. C. Dee, and E. A. Nauman, "Examination of continuum and micro-structural properties of human vertebral cancellous bone using combined cellular solid models," *Biomechanics and modeling in mechanobiology*, vol. 2, no. 2, pp. 97–107, 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/14586811

[69] B. Wang, X. Zhou, C. Price, W. Li, J. Pan, and L. Wang, "Quantifying load-induced solute transport and solute-matrix interaction within the osteocyte lacunar-canalicular system," *Journal of Bone and Mineral Research*, vol. 28, no. 5, pp. 1075–1086, 2013.

[70] M. L. Knothe Tate, P. Niederer, and U. Knothe, "In vivo tracer transport through the lacunocanalicular system of rat bone in an environment devoid of mechanical loading," *Bone*, vol. 22, no. 2, pp. 107–117, 1998.

[71] F. A. Shah, B. R. Johansson, P. Thomsen, and A. Palmquist, "Ultrastructural evaluation of shrinkage artefacts induced by fixatives and embedding resins on osteocyte processes and pericellular space dimensions," *Journal of Biomedical Materials Research - Part A*, vol. 103, no. 4, pp. 1565–1576, 2015.

[72] S. C. Cowin and S. B. Doty, *Tissue Mechanics*. Springer Science +Business Media, LLC, 2007.

[73] R. F. Capozza, S. Feldman, P. Mortarino, P. S. Reina, H. Schiessl, J. Rittweger, J. L. Ferretti, and G. R. Cointry, "Structural analysis of the human tibia by tomographic (pQCT) serial scans," *Journal of Anatomy*, vol. 216, no. 4, pp. 470–481, 2010.

[74] K. A. Verner, "Skeletal biomechanics and response to mechanical load: A comparative approach in the mouse and chukar partridge," Ph.D. dissertation, Purdue University, 2017. [Online]. Available: https://search.proquest.com/docview/2008985108?fromunauthdoc=true

[75] B. Galle, "A combined experimental and computational approach to investigate the mechanism of spinal cord slow compression primary cellular injury," Ph.D. dissertation, Purdue University, 2008.

[76] D.-J. Kroon, "Fast Non-Local Means 1D, 2D Color and 3D," 2010. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/27395-fast-non-local-means-1d–2d-color-and-3d

[77] A. Buades, B. Coll, and J.-M. Morel CMLA, "A non-local algorithm for image denoising," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2005, pp. 60–65. [Online]. Available: http://www.coe.utah.edu/ cs7640/readings/NL-means.pdf

[78] J. V. Manjon-Herrera, "Non-Local Means Filter," 2007. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/13176-non-local-means-filter

[79] D.-J. Kroon, "Hessian based Frangi Vesselness filter," 2009. [Online]. Available: http://ieeexplore.ieee.org/document/534065/

[80] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," vol. 1496, pp. 130–137, 1998. [Online]. Available: http://link.springer.com/10.1007/BFb0056195

[81] R. Manniesing and W. Niessen, "Multiscale Vessel Enhancing Diffusion in CT Angiography Noise Filtering," in *Information Processing in Medical Imaging. IPMI 2005. Lecture Notes in Computer Science*, G. Christensen and M. Sonka, Eds. Springer, Berlin, Heidelberg, 2005, pp. 138–149.

[82] Keiichiro, "Anisotropic morphological opening," 2014. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/45656-anisotropic-morphological-opening

[83] K. Shirai, Y. Endo, A. Kitadai, S. Inoue, N. Kurushima, H. Baba, A. Watanabe, and M. Nakagawa, "Character Shape Restoration of Binarized Historical Documents by Smoothing via Geodesic Morphology," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 8 2013, pp. 1285–1289. [Online]. Available: http://ieeexplore.ieee.org/document/6628821/

[84] D. Tschumperle and R. Deriche, "Vector-valued image regularization with PDEs: a common framework for different applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 506–517, 4 2005. [Online]. Available: http://ieeexplore.ieee.org/document/1401905/

[85] N. Howe, "Better Skeletonization," 2007. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/11123-better-skeletonization

[86] L. Xie, "3D structuring element (sphere)," 2014. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/47937-3d-structuring-element–sphere-

[87] Y. Petrov, "Ellipsoid fit," 2015. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/24693-ellipsoid-fit

[88] D.-J. Kroon, "Polygon2Voxel," 2012. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/24086-polygon2voxel

[89] P. Kollmannsberger, "Skeleton3D," 2018. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/43400-skeleton3d

[90] ——, "Skel2Graph 3D," 2018. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/43527-skel2graph-3d

[91] S. M. Tommasini, A. Trinward, A. S. Acerbo, F. De Carlo, L. M. Miller, and S. Judex, "Changes in intracortical microporosities induced by pharmaceutical treatment of osteoporosis as detected by high resolution micro-CT," *Bone*, vol. 50, no. 3, pp. 596–604, 3 2012. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S8756328211013962

[92] R. L. Hunter and A. M. Agnew, "Intraskeletal variation in human cortical osteocyte lacunar density: Implications for bone quality assessment." *Bone reports*, vol. 5, pp. 252–261, 12 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/28580394 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5441015

[93] B. Busse, D. Djonic, P. Milovanovic, M. Hahn, K. Püschel, R. O. Ritchie, M. Djuric, and M. Amling, "Decrease in the osteocyte lacunar density accompanied by hypermineralized lacunar occlusion reveals failure and delay of remodeling in aged human bone," *Aging Cell*, vol. 9, no. 6, pp. 1065–1075, 2010.

[94] L. M. Tiede-Lewis, Y. Xie, M. A. Hulbert, R. Campos, M. R. Dallas, V. Dusevich, L. F. Bonewald, and S. L. Dallas, "Degeneration of the osteocyte network in the C57BL/6 mouse model of aging." *Aging*, vol. 9, no. 10, pp. 2190–2208, 10 2017. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/29074822 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC5680562

[95] X. Lai, C. Price, S. Modla, W. R. Thompson, J. Caplan, C. B. Kirn-Safran, and L. Wang, "The dependences of osteocyte network on bone compartment, age, and disease," *Bone Res.*, vol. 3, no. November 2014, p. 15009, 2015.

[96] F. L. Bach-Gansmo, J. C. Weaver, M. H. Jensen, H. Leemreize, K. S. Mader, M. Stampanoni, A. Brüel, J. S. Thomsen, and H. Birkedal, "Osteocyte lacunar properties in rat cortical bone: Differences between lamellar and central bone," *Journal of Structural Biology*, vol. 191, no. 1, pp. 59–67, 7 2015.

[97] L. Wang, Y. Wang, Y. Han, S. C. Henderson, R. J. Majeska, S. Weinbaum, and M. B. Schaffler, "In situ measurement of solute transport in the bone lacunar-canalicular system," *Proceedings of the National Academy of Sciences*, vol. 102, no. 33, pp. 11 911–11 916, 8 2005. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/16087872 http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC1187997 http://www.pnas.org/cgi/doi/10.1073/pnas.0505193102

[98] Y. Sugawara, R. Ando, H. Kamioka, Y. Ishihara, T. Honjo, N. Kawanabe, H. Kurosaka, T. Takano-Yamamoto, and T. Yamashiro, "The Three-Dimensional Morphometry and CellCell Communication of the Osteocyte Network in Chick and Mouse Embryonic Calvaria," *Calcified Tissue International*, vol. 88, no. 5, pp. 416–424, 5 2011.

APPENDICES

A. MATHEMATICAL MODEL SOURCE CODE

A.1 γ Function Sensitivity Analysis

```
%Variable input: Low, high inputs for each n number variables. Input such
%that first and second inputs are low and then high value of 1st variable (respectively),
%third and fourth inputs are the low and high of the 2nd variable, etc.

varargin = [75e-9 422e-9 0.5e-9 4.9e-9 6e-9 10e-9 0.5e-9 4.9e-9 6e-9 10e-9];
variables = length(varargin)/2; %number of variables is equal to the number of inputs/2
n = 2*variables + 2; %number of trials or simulations that need to be performed

%Creates a matrix to be used to extract the low variables
all_low = zeros(1, variables); %creates row vector of just zeros
low_identity = eye((n/2)-1, variables); %creates identity matrix
low_matrix = vertcat(all_low, low_identity); %combines zero and identity matrices

%Creates a matrix to be used to extract the high variables
```

```
15 all_high = zeros(1, variables); %creates row vector of just
       zeros
16 high_identity = -1*eye((n/2)-1, variables); %creates identity
       matrix
17 high_matrix = vertcat(high_identity, all_high); %combines
       zero and identity matrices
18
19 %Creates the matrix consisting of the high and low input
       values, in the
20 %necessary format for Cotter's Method
21 for i = 1:1:n/2
22     for m = 1:1:variables
23         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
24         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
25     end
26 end
27
28 %Looping through all of the trials
29 for i = 1:1:n
30     m = 1;
31     %Depends upon order the variables were input into the
           function
32     b(i,1) = values(i,1); %radius of the osteocyte cell
           process
33     Dv(i,1) = values(i,2); %diameter of vertical fibers
34     Lv(i,1) = values(i,3); %length of vertical fibers
35     Dh(i,1) = values(i,4); %diameter of horizontal fibers
```

```
36      r(i,1) = values(i,5); %length of horizontal fibers
37
38      %Equations from Sander et al. 2003
39      %Determine volume fractions based on glycocalyx
            nanostructure (rectangular
40      %unit cell)
41      phi_s(i,1) = pi/4*((Dv(i,1)^2*Lv(i,1) + (2*Dh(i,1)^2*(r(i
            ,1) - Dv(i,1))))/(r(i,1)^2*Lv(i,1))); %volume fraction
             of the solid glycocalyx
42      phi_f(i,1) = 1 - phi_s(i,1); %volume of the interstitial
            fluid flow
43
44      %Calculate vertical permeability of glycocalyx given
            defined nanostructure
45      Aw(i,1) = pi*(Dv(i,1)*(Lv(i,1)-Dh(i,1)) + 2*Dh(i,1)*(r(i
            ,1)-Dv(i,1))); %Wetted surface of the unit cell
46      Cl(i,1) = (4*pi*(Lv(i,1)-Dh(i,1)))/(log(r(i,1)^2/(pi*(Dv(
            i,1)/2))) - 1.476 + (2*pi*(Dv(i,1)/2)^2/r(i,1)^2) - (
            pi^2*(Dv(i,1)/2)^4/(2*r(i,1)^4)) - 0.0150*((pi*(Dv(i
            ,1)/2)^2/r(i,1)^2)^4/(1 + (1.520*(pi*(Dv(i,1)/2)^2/r(i
            ,1)^2)^4))));
47      Ct(i,1) = (4*pi*(r(i,1)-Dv(i,1)))/(log(r(i,1)/(Dh(i,1)/2)
            ) -1.311 + (pi*(Dh(i,1)/2)^2/r(i,1)^2) - (pi^2*(Dh(i
            ,1)/2)^4/(2*r(i,1)^4)) - 8.756*((Dh(i,1)/2)/r(i,1))^4
           + 63.212*((Dh(i,1)/2)/r(i,1))^6);
48      Cd(i,1) = (128*r(i,1)^2*Aw(i,1)*Lv(i,1))/(pi*(r(i,1)-Dv(i
            ,1))^2);
```

```
49      K(i,1) = r(i,1)^2*Lv(i,1)/(Cl(i,1) + 2*Ct(i,1) + Cd(i,1))
            ; %permeability of the glycocalyx in the vertical
            direction
50
51      gamma(i,1) = (b(i,1))/(sqrt(phi_f(i,1)*K(i,1)));
52  end
53
54  sum_measures = 0;
55  for j = 1:1:variables
56      contrast_odd(1,j) = 1/4*((gamma(2*variables+2,1) - gamma(
            j+variables+1,1)) + (gamma(j+1,1)- gamma(1,1)));
57      contrast_even(1,j) = 1/4*((gamma(2*variables+2,1) - gamma
            (j+variables+1,1)) - (gamma(j+1,1)- gamma(1,1)));
58      measure(1,j) = abs(contrast_odd(1,j)) + abs(contrast_even
            (1,j));
59      sum_measures = sum_measures + measure(1,j);
60  end
61
62  for j = 1:1:variables
63       sens(1,j) = measure(1,j)/sum_measures;
64  end
65
66  %Sensitivity analysis for the gamma function
67  names = {'b', 'Dv', 'Lv', 'Dh', 'Lh'};
68  bar(sens(1,:), 0.4)
69  xlabel('Parameter')
70  ylabel('Sensitivity')
71  title('\gamma Term Sensitivity Analysis')
72  hold on
```

```
plot(xlim, [.20, .20], ':')
set(gca, 'xticklabel', names)
```

A.2 ϕ^s Sensitivity Analysis

```
%Variable input: Low, high inputs for each n number variables. Input such
%that first and second inputs are low and then high value of 1st variable (respectively),
%third and fourth inputs are the low and high of the 2nd variable, etc.

%Variables for Cotter's Method Sensitivity analysis
varargin = [0.5e-9 5.9e-9 6e-9 10e-9 4.75e-9 5.9e-9 6e-9 7.25e-9];
variables = length(varargin)/2; %number of variables is equal to the number of inputs/2
n = 2*variables + 2; %number of trials or simulations that need to be performed

%Creates a matrix to be used to extract the low variables
all_low = zeros(1, variables); %creates row vector of just zeros
low_identity = eye((n/2)-1, variables); %creates identity matrix
low_matrix = vertcat(all_low, low_identity); %combines zero and identity matrices

%Creates a matrix to be used to extract the high variables
```

```
16 all_high = zeros(1, variables); %creates row vector of just
       zeros
17 high_identity = -1*eye((n/2)-1, variables); %creates identity
       matrix
18 high_matrix = vertcat(high_identity, all_high); %combines
       zero and identity matrices
19
20 %Creates the matrix consisting of the high and low input
       values, in the
21 %necessary format for Cotter's Method
22 for i = 1:1:n/2
23     for m = 1:1:variables
24         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
25         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
26     end
27 end
28
29 %Define constants within the system (fluid = PBS)
30 mu = 0.0007; %viscosity of the fluid
31 rho_f_true = 1000; %true density of the fluid
32
33 %Looping through all of the trials
34 for i = 1:1:n
35     m = 1;
36     %Depends upon order the variables were input into the
           function
37     Dv(i,1) = values(i,1); %diameter of vertical fibers
```

```
38      Lv(i,1) = values(i,2); %length of vertical fibers
39      Dh(i,1) = values(i,3); %diameter of horizontal fibers
40      Lh(i,1) = values(i,4); %length of horizontal fibers
41
42      %Equations from Sander et al. 2003
43      %Determine volume fractions based on glycocalyx
            nanostructure (rectangular
44      %unit cell)
45      phi_s(i,1) = pi/4*((Dv(i,1)^2*Lv(i,1) + (2*Dh(i,1)^2*(Lh(
            i,1) - Dv(i,1))))/(Lh(i,1)^2*Lv(i,1))); %volume
            fraction of the solid glycocalyx
46  end
47
48  sum_measures = 0;
49  for j = 1:1:variables
50      contrast_odd(1,j) = 1/4*((phi_s(2*variables+2,1) - phi_s(
            j+variables+1,1)) + (phi_s(j+1,1)- phi_s(1,1)));
51      contrast_even(1,j) = 1/4*((phi_s(2*variables+2,1) - phi_s
            (j+variables+1,1)) - (phi_s(j+1,1)- phi_s(1,1)));
52      measure(1,j) = abs(contrast_odd(1,j)) + abs(contrast_even
            (1,j));
53      sum_measures = sum_measures + measure(1,j);
54  end
55
56  for j = 1:1:variables
57       sens(1,j) = measure(1,j)/sum_measures;
58  end
59
60  %Plots sensitivity analysis of the solid volume fraction
```

```
figure(1)
bar(sens(1,:), 0.4)
xlabel('Parameter')
ylabel('Sensitivity')
title('\phi^s Sensitivity Analysis (Glycocaylx)')
hold on
plot(xlim, [.25, .25], ':')
names = {'Dv', 'Lv', 'Dh', 'Lh', 'dP/dz', 'v_{o}'};
set(gca, 'xticklabel', names)
```

A.3 Brinkman Velocity and Shear Stress Sensitivity Analysis

```
%Variable input: Low, high inputs for each n number variables. Input such
%that first and second inputs are low and then high value of 1st variable (respectively),
%third and fourth inputs are the low and high of the 2nd variable, etc.

varargin = [0.245 0.819 75e-9 422e-9 0.5e-9 6e-9 6e-9 10e-9 0.5e-9 4.75e-9 7.25e-9 10e-9 -11.87e3/6e-3 -15.87e3/6e-3 21.6e-6 84e-6];
variables = length(varargin)/2; %number of variables is equal to the number of inputs/2
n = 2*variables + 2; %number of trials or simulations that need to be performed

%Creates a matrix to be used to extract the low variables
all_low = zeros(1, variables); %creates row vector of just zeros
```

```
11 low_identity = eye((n/2)-1, variables); %creates identity
       matrix
12 low_matrix = vertcat(all_low, low_identity); %combines zero
       and identity matrices
13
14 %Creates a matrix to be used to extract the high variables
15 all_high = zeros(1, variables); %creates row vector of just
       zeros
16 high_identity = -1*eye((n/2)-1, variables); %creates identity
        matrix
17 high_matrix = vertcat(high_identity, all_high); %combines
       zero and identity matrices
18
19 %Creates the matrix consisting of the high and low input
       values, in the
20 %necessary format for Cotter's Method
21 for i = 1:1:n/2
22     for m = 1:1:variables
23         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
24         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
25     end
26 end
27
28 %Define constants within the system (fluid = PBS)
29 mu = 0.0007; %viscosity of the fluid
30 rho_f_true = 1000; %true density of the fluid
31
```

```
32 %Looping through all of the trials
33 for i = 1:1:n
34     m = 1;
35     %Depends upon order the variables were input into the
           function
36     ab(i,1) = values(i,1); %radius of the osteocyte cell
           process
37     b(i,1) = values(i,2); %radius of the canaliculus
38     Dv(i,1) = values(i,3); %diameter of vertical fibers
39     Lv(i,1) = values(i,4); %length of vertical fibers
40     Dh(i,1) = values(i,5); %diameter of horizontal fibers
41     Lh(i,1) = values(i,6); %length of horizontal fibers
42     dp_dz(i,1) = values(i,7); %pressure gradient down the
           length of the canaliculus
43     v_o(i,1) = values(i,8); %initial velocity of the fluid
44
45     %Equations from Sander et al. 2003
46     %Determine volume fractions based on glycocalyx
           nanostructure (rectangular
47     %unit cell)
48     phi_s(i,1) = pi/4*((Dv(i,1)^2*Lv(i,1) + (2*Dh(i,1)^2*(Lh(
           i,1) - Dv(i,1))))/(Lh(i,1)^2*Lv(i,1))); %volume
           fraction of the solid glycocalyx
49     phi_f(i,1) = 1 - phi_s(i,1); %volume of the interstitial
           fluid flow
50
51     %Calculate vertical permeability of glycocalyx given
           defined nanostructure
```

```
52  Aw(i,1) = pi*(Dv(i,1)*(Lv(i,1)-Dh(i,1)) + 2*Dh(i,1)*(Lh(i
        ,1)-Dv(i,1))); %Wetted surface of the unit cell
53  Cl(i,1) = (4*pi*(Lv(i,1)-Dh(i,1)))/(log(Lh(i,1)^2/(pi*(Dv
        (i,1)/2))) - 1.476 + (2*pi*(Dv(i,1)/2)^2/Lh(i,1)^2) -
        (pi^2*(Dv(i,1)/2)^4/(2*Lh(i,1)^4)) - 0.0150*((pi*(Dv(i
        ,1)/2)^2/Lh(i,1)^2)^4/(1 + (1.520*(pi*(Dv(i,1)/2)^2/Lh
        (i,1)^2)^4))));
54  Ct(i,1) = (4*pi*(Lh(i,1)-Dv(i,1)))/(log(Lh(i,1)/(Dh(i,1)
        /2)) -1.311 + (pi*(Dh(i,1)/2)^2/Lh(i,1)^2) - (pi^2*(Dh
        (i,1)/2)^4/(2*Lh(i,1)^4)) - 8.756*((Dh(i,1)/2)/Lh(i,1)
        )^4 + 63.212*((Dh(i,1)/2)/Lh(i,1))^6);
55  Cd(i,1) = (128*Lh(i,1)^2*Aw(i,1)*Lv(i,1))/(pi*(Lh(i,1)-Dv
        (i,1))^2);
56  K(i,1) = Lh(i,1)^2*Lv(i,1)/(Cl(i,1) + 2*Ct(i,1) + Cd(i,1)
        ); %permeability of the glycocalyx in the vertical
        direction
57
58  N(i,1) = v_o(i,1)*rho_f_true*sqrt(K(i,1))/mu; %Nauman
        number
59  gamma(i,1) = (b(i,1))/(sqrt(phi_f(i,1)*K(i,1)));
60
61  %Equation for A and B, modified Bessel function constants
        in the equation for the
62  %velocity profile within a canaliculus
63  A(i,1) = (besselk(0, gamma(i,1)) - besselk(0, gamma(i,1)
        *(ab(i,1))))/(besseli(0, gamma(i,1)*(ab(i,1)))*besselk
        (0, gamma(i,1)) - besseli(0, gamma(i,1))*besselk(0,
        gamma(i,1)*(ab(i,1))));
```

```
64      B(i,1) = (besseli(0, gamma(i,1)*(ab(i,1))) - besseli(0,
            gamma(i,1)))/(besseli(0, gamma(i,1)*(ab(i,1)))*besselk
            (0, gamma(i,1)) - besseli(0, gamma(i,1))*besselk(0,
            gamma(i,1)*(ab(i,1))));
65
66      nondim_K(i,1) = K(i,1)/K(i,1);
67      nondim_dp_dz(i,1) = (dp_dz(i,1)*sqrt(K(i,1)))/((v_o(i,1))
            ^2*rho_f_true);
68      nondim_b(i,1) = b(i,1)/sqrt(K(i,1));
69
70      %Determines fluid velocity and shear stress as a function
             of symbolic
71      %radius
72      syms radius;
73      nondim_sym_velocityf(i,1) = N(i,1)*phi_f(i,1)*nondim_K(i
            ,1)*nondim_dp_dz(i,1)*(A(i,1)*besseli(0, gamma(i,1)*
            radius/nondim_b(i,1)) + B(i,1)*besselk(0, gamma(i,1)*
            radius/nondim_b(i,1)) - 1); %creates fluid velocity
            equation as a function of radius
74      nondim_sym_shear(i,1) = (1/rho_f_true)*(1/v_o(i,1))*(1/
            sqrt(K(i,1)))*mu*diff(nondim_sym_velocityf(i,1),
            radius); %differentiates (with respect to radius)
            fluid velocity equation to create shear stress
            equation as a function of radius
75      radiusV = ((b(i,1) + ab(i,1)*b(i,1))/2)/sqrt(K(i,1));
76      radiusS = b(i,1)*ab(i,1)/sqrt(K(i,1)); %
            nondimensionalized
77      nd_velocityf(i,1) = subs(nondim_sym_velocityf(i,1),
            radiusV);
```

```
78      nd_shear(i,1) = subs(nondim_sym_shear(i,1), radiusS); %
            substitutes in cell process radius to numerically
            determine value of shear stress
79      shear(i,1) = nd_shear(i,1)*rho_f_true*v_o(i,1)^2;
80  end
81
82  ndd_velocityf(:,1) = double(nd_velocityf(:,1));
83  ndd_shear(:,1) = double(nd_shear(:,1));
84  actual_shear(:,1) = double(shear(:,1));
85
86  sum_measures_v = 0;
87  for j = 1:1:variables
88      contrast_odd_v(1,j) = 1/4*((ndd_velocityf(2*variables
            +2,1) - ndd_velocityf(j+variables+1,1)) + (
            ndd_velocityf(j+1,1)- ndd_velocityf(1,1)));
89      contrast_even_v(1,j) = 1/4*((ndd_velocityf(2*variables
            +2,1) - ndd_velocityf(j+variables+1,1)) - (
            ndd_velocityf(j+1,1)- ndd_velocityf(1,1)));
90      measure_v(1,j) = abs(contrast_odd_v(1,j)) + abs(
            contrast_even_v(1,j));
91      sum_measures_v = sum_measures_v + measure_v(1,j);
92  end
93
94  for j = 1:1:variables
95       sens_v(1,j) = measure_v(1,j)/sum_measures_v;
96  end
97
98  sum_measures = 0;
99  for j = 1:1:variables
```

```
100     contrast_odd(1,j) = 1/4*((ndd_shear(2*variables+2,1) -
            ndd_shear(j+variables+1,1)) + (ndd_shear(j+1,1)-
            ndd_shear(1,1)));
101     contrast_even(1,j) = 1/4*((ndd_shear(2*variables+2,1) -
            ndd_shear(j+variables+1,1)) - (ndd_shear(j+1,1)-
            ndd_shear(1,1)));
102     measure(1,j) = abs(contrast_odd(1,j)) + abs(contrast_even
            (1,j));
103     sum_measures = sum_measures + measure(1,j);
104 end
105
106 for j = 1:1:variables
107      sens(1,j) = measure(1,j)/sum_measures;
108 end
109
110 %Sensitivity analysis on the brinkman velocity profile
111 figure(1)
112 bar(sens(1,:), 0.4)
113 xlabel('Parameter')
114 ylabel('Sensitivity')
115 title('Brinkman Shear Stress Sensitivity Analysis (Glycocaylx
        )')
116 hold on
117 plot(xlim, [.125, .125], ':')
118 names = {'a/b', 'b', 'Dv', 'Lv', 'Dh', 'Lh', 'dP/dz', 'v_{o}'
        };
119 set(gca, 'xticklabel', names)
120
121 %Sensitivity analysis on the brinkman shear stress
```

```
figure(2)
bar(sens_v(1,:), 0.4)
xlabel('Parameter')
ylabel('Sensitivity')
title('Brinkman Velocity Sensitivity Analysis (Glycocaylx)')
hold on
plot(xlim, [.125, .125], ':')
names = {'a/b', 'b', 'Dv', 'Lv', 'Dh', 'Lh', 'dP/dz', 'v_{o}'};
set(gca, 'xticklabel', names)
```

A.4 Brinkman Velocity

```
%Inputs into Cotter's Method sensitivity analysis
varargin = [0.5e-9 5.9e-9 6.75e-9 10e-9 4.75e-9 5.25e-9 6e-9 7.25e-9 -11.87e3/6e-3 -15.87e3/6e-3 21.6e-6 84e-6];
variables = length(varargin)/2; %number of variables is equal to the number of inputs/2
n = 2*variables + 2; %number of trials or simulations that need to be performed

%Creates a matrix to be used to extract the low variables
all_low = zeros(1, variables); %creates row vector of just zeros
low_identity = eye((n/2)-1, variables); %creates identity matrix
low_matrix = vertcat(all_low, low_identity); %combines zero and identity matrices

%Creates a matrix to be used to extract the high variables
```

```
12 all_high = zeros(1, variables); %creates row vector of just
       zeros
13 high_identity = -1*eye((n/2)-1, variables); %creates identity
       matrix
14 high_matrix = vertcat(high_identity, all_high); %combines
       zero and identity matrices
15
16 %Creates the matrix consisting of the high and low input
       values, in the
17 %necessary format for Cotter's Method
18 for i = 1:1:n/2
19     for m = 1:1:variables
20         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
21         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
22     end
23 end
24
25 %Determine volume fraction and permeability for each
       simulation
26 for i = 1:1:n
27     m = 1;
28     %Depends upon order the variables were input into the
           function
29     Dv(i,1) = values(i,1); %diameter of vertical fibers
30     Lv(i,1) = values(i,2); %length of vertical fibers
31     Dh(i,1) = values(i,3); %diameter of horizontal fibers
32     Lh(i,1) = values(i,4); %length of horizontal fibers
```

```
33      dp_dz(i,1) = values(i,5); %pressure gradient down the
            length of the canaliculus
34      v_o(i,1) = values(i,6); %initial velocity of the fluid
35
36      %Equations from Sander et al. 2003
37      %Determine volume fractions based on glycocalyx
            nanostructure (rectangular unit cell)
38      phi_s(i,1) = pi/4*((Dv(i,1)^2*Lv(i,1) + (2*Dh(i,1)^2*(Lh(
            i,1) - Dv(i,1))))/(Lh(i,1)^2*Lv(i,1))); %volume
            fraction of the solid glycocalyx
39      phi_f(i,1) = 1 - phi_s(i,1); %volume of the interstitial
            fluid flow
40
41      %Calculate vertical permeability of glycocalyx given
            defined nanostructure
42      Aw(i,1) = pi*(Dv(i,1)*(Lv(i,1)-Dh(i,1)) + 2*Dh(i,1)*(Lh(i
            ,1)-Dv(i,1))); %Wetted surface of the unit cell
43      Cl(i,1) = (4*pi*(Lv(i,1)-Dh(i,1)))/(log(Lh(i,1)^2/(pi*(Dv
            (i,1)/2))) - 1.476 + (2*pi*(Dv(i,1)/2)^2/Lh(i,1)^2) -
            (pi^2*(Dv(i,1)/2)^4/(2*Lh(i,1)^4)) - 0.0150*((pi*(Dv(i
            ,1)/2)^2/Lh(i,1)^2)^4/(1 + (1.520*(pi*(Dv(i,1)/2)^2/Lh
            (i,1)^2)^4))));
44      Ct(i,1) = (4*pi*(Lh(i,1)-Dv(i,1)))/(log(Lh(i,1)/(Dh(i,1)
            /2)) -1.311 + (pi*(Dh(i,1)/2)^2/Lh(i,1)^2) - (pi^2*(Dh
            (i,1)/2)^4/(2*Lh(i,1)^4)) - 8.756*((Dh(i,1)/2)/Lh(i,1)
            )^4 + 63.212*((Dh(i,1)/2)/Lh(i,1))^6);
45      Cd(i,1) = (128*Lh(i,1)^2*Aw(i,1)*Lv(i,1))/(pi*(Lh(i,1)-Dv
            (i,1))^2);
```

```
46      K(i,1) = Lh(i,1)^2*Lv(i,1)/(Cl(i,1) + 2*Ct(i,1) + Cd(i,1)
            ); %permeability of the glycocalyx in the vertical
            direction
47  end
48
49  %Defining variables needed to plot velocity profile
50  index = 6e-8;
51  %Canaliculus radius (b) (m)
52  b_low = 75e-9;
53  b_high = 422e-9;
54  %Ratio of osteocyte cell process (a) to canaliculus radius (b
        )
55  ratio_ab_low = 0.245;
56  ratio_ab_high = 0.819;
57  %Diemnsions (length and diameter) of the glycocaylx in two
        directions
58  %(vertical and horizontal) (m)
59  Lv_low = 6e-9;
60  Lv_high = 10e-9;
61  Lh_low = 6e-9;
62  Lh_high = 10e-9;
63  Dv_low = 0.5e-9;
64  Dv_high = 6e-9;
65  Dh_low = 0.5e-9;
66  Dh_high = 6e-9;
67  %Initial velocity (m/s)
68  v_o_low = 21.6e-6;
69  v_o_high = 84e-6;
70  %Pressure gradient down the length of the canaliculus (Pa/m)
```

```
71 dp_dz_low = -11.87e3/6e-3;
72 dp_dz_high = -15.87e3/6e-3;
73 %Viscosity of the fluid (Pa s)
74 mu = 0.0007;
75 %True density of the fluid (kg/m^3)
76 rho_f_true = 1000;
77 %Fluid constituent volume fractions
78 phi_f_high = max(phi_f);
79 phi_f_low = min(phi_f);
80 %Glycocaylx permeability values (m^2)
81 K_low = min(K);
82 K_high = max(K);
83 %Gamma term
84 gamma_high = b_high/sqrt(phi_f_low*K_low);
85 gamma_low = b_low/sqrt(phi_f_high*K_high);
86 r_gamma_low = round(gamma_low);
87
88 gamma_index = 50;
89 i=1;
90 for gamma = r_gamma_low:gamma_index:700
91     j = 1;
92     for ab = ratio_ab_low:0.1:ratio_ab_high
93         if gamma < 700 %Brinkman flow
94             %Use averages to nondimensionalize parameters and
                calculate velocity
95             Kv(i,j) = ((b_high + b_low)/2)^2/(gamma^2);
96             K(i,j) = (Kv(i,j)/((phi_f_high + phi_f_low)/2));
97             N(i,j) = ((v_o_high + v_o_low)/2)*rho_f_true*sqrt(
                K(i,j))/mu;
```

```
98          radius = ((b_high + b_low)/2 + ab*(b_high + b_low)
                /2)/2;
99
100         %Nondimensionalization of variables
101         nondim_radius = radius/sqrt(K(i,j));
102         nondim_b = ((b_high + b_low)/2)/sqrt(K(i,j));
103         nondim_Kv = (Kv(i,j)/K(i,j));
104         nondim_dp = ((dp_dz_high + dp_dz_low)/2)*sqrt(K(i,
                j))/(((v_o_high + v_o_low)/2)^2*rho_f_true);
105
106         %Calculation of bessel constants,
                nondimensionalized maximum
107         %velocity profile (middle of annulus gap), and
108         %re-dimensionalized maximum velocity profile
109         A(i,j) = (besselk(0, gamma) - besselk(0, gamma*ab)
                )/(besseli(0, gamma*ab)*besselk(0, gamma) -
                besseli(0, gamma)*besselk(0, gamma*ab));
110         B(i,j) = (besseli(0, gamma*ab) - besseli(0, gamma)
                )/(besseli(0, gamma*ab)*besselk(0, gamma) -
                besseli(0, gamma)*besselk(0, gamma*ab));
111         nondim_vf(i,j) = N(i,j)*nondim_Kv*nondim_dp*(A(i,j
                )*besseli(0, gamma*nondim_radius/nondim_b) + B(
                i,j)*besselk(0, gamma*nondim_radius/nondim_b) -
                 1); %creates fluid velocity equation as a
                function of radius
112         vf(i,j) = nondim_vf(i,j)*((v_o_high + v_o_low)/2)
                *1e6;
113         j = j+1;
114     end
```

```
115     end
116     i = i+1;
117 end
118
119 ab = ratio_ab_low:0.1:ratio_ab_high;
120 gamma = r_gamma_low:gamma_index:700;
121
122 %Plots nondimensionalized maximum velocity
123 figure(1)
124 hold on
125 for m = 1:1:(i-1)
126     plot(ab(:), nondim_vf(m,:), 'DisplayName', ['gamma = '
          num2str(gamma(m))]);
127 end
128 xlabel('Ratio of Osteocyte Process to Canaliculus (a/b)')
129 ylabel('Nondimensionalized Maximum Velocity')
130 title('Nondimensionalized Brinkman Velocity Profile (
    Glycocaylx)')
131 legend(gca, 'show')
132
133 %Plots re-dimensionalized maximum velocity
134 figure(2)
135 hold on
136 for m = 1:1:(i-1)
137     plot(ab(:), vf(m,:), 'DisplayName', ['gamma = ' num2str(
          gamma(m))]);
138 end
139 xlabel('Ratio of Osteocyte Process to Canaliculus (a/b)')
140 ylabel('Maximum Velocity (\mum/s)')
```

```
141 title('Brinkman Velocity Profile (Glycocaylx)')
142 legend(gca, 'show')
```

A.5 Darcy Velocity Sensitivity Analysis

```
1  %Variable input: Low, high inputs for each n number variables
       . Input such
2  %that first and second inputs are low and then high value of
       1st variable (respectively),
3  %third and fourth inputs are the low and high of the 2nd
       variable, etc.
4
5  varargin = [0.5e-9 5.9e-9 6.75e-9 10e-9 4.75e-9 5.25e-9 6e-9
       7.25e-9 -11.87e3/6e-3 -15.87e3/6e-3 21.6e-6 84e-6];
6  variables = length(varargin)/2; %number of variables is equal
        to the number of inputs/2
7  n = 2*variables + 2; %number of trials or simulations that
       need to be performed
8
9  %Creates a matrix to be used to extract the low variables
10 all_low = zeros(1, variables); %creates row vector of just
       zeros
11 low_identity = eye((n/2)-1, variables); %creates identity
       matrix
12 low_matrix = vertcat(all_low, low_identity); %combines zero
       and identity matrices
13
14 %Creates a matrix to be used to extract the high variables
15 all_high = zeros(1, variables); %creates row vector of just
       zeros
```

```
16 high_identity = -1*eye((n/2)-1, variables); %creates identity
       matrix
17 high_matrix = vertcat(high_identity, all_high); %combines
      zero and identity matrices
18
19 %Creates the matrix consisting of the high and low input
      values, in the
20 %necessary format for Cotter's Method
21 for i = 1:1:n/2
22     for m = 1:1:variables
23         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
24         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
25     end
26 end
27
28 %Define constants within the system (fluid = PBS)
29 mu = 0.0007; %viscosity of the fluid
30 rho_f_true = 1000; %true density of the fluid
31
32 %Looping through all of the trials
33 for i = 1:1:n
34     m = 1;
35     %Depends upon order the variables were input into the
          function
36     Dv(i,1) = values(i,1); %diameter of vertical fibers
37     Lv(i,1) = values(i,2); %length of vertical fibers
38     Dh(i,1) = values(i,3); %diameter of horizontal fibers
```

```
39      Lh(i,1) = values(i,4); %length of horizontal fibers
40      dp_dz(i,1) = values(i,5); %pressure gradient down the
            length of the canaliculus
41      v_o(i,1) = values(i,6); %initial velocity of the fluid
42
43      %Equations from Sander et al. 2003
44      %Determine volume fractions based on glycocalyx
            nanostructure (rectangular
45      %unit cell)
46      phi_s(i,1) = pi/4*((Dv(i,1)^2*Lv(i,1) + (2*Dh(i,1)^2*(Lh(
            i,1) - Dv(i,1))))/(Lh(i,1)^2*Lv(i,1))); %volume
            fraction of the solid glycocalyx
47      phi_f(i,1) = 1 - phi_s(i,1); %volume of the interstitial
            fluid flow
48
49      %Calculate vertical permeability of glycocalyx given
            defined nanostructure
50      Aw(i,1) = pi*(Dv(i,1)*(Lv(i,1)-Dh(i,1)) + 2*Dh(i,1)*(Lh(i
            ,1)-Dv(i,1))); %Wetted surface of the unit cell
51      Cl(i,1) = (4*pi*(Lv(i,1)-Dh(i,1)))/(log(Lh(i,1)^2/(pi*(Dv
            (i,1)/2))) - 1.476 + (2*pi*(Dv(i,1)/2)^2/Lh(i,1)^2) -
            (pi^2*(Dv(i,1)/2)^4/(2*Lh(i,1)^4)) - 0.0150*((pi*(Dv(i
            ,1)/2)^2/Lh(i,1)^2)^4/(1 + (1.520*(pi*(Dv(i,1)/2)^2/Lh
            (i,1)^2)^4))));
52      Ct(i,1) = (4*pi*(Lh(i,1)-Dv(i,1)))/(log(Lh(i,1)/(Dh(i,1)
            /2)) -1.311 + (pi*(Dh(i,1)/2)^2/Lh(i,1)^2) - (pi^2*(Dh
            (i,1)/2)^4/(2*Lh(i,1)^4)) - 8.756*((Dh(i,1)/2)/Lh(i,1)
            )^4 + 63.212*((Dh(i,1)/2)/Lh(i,1))^6);
```

```
53      Cd(i,1) = (128*Lh(i,1)^2*Aw(i,1)*Lv(i,1))/(pi*(Lh(i,1)-Dv
            (i,1))^2);
54      K(i,1) = Lh(i,1)^2*Lv(i,1)/(Cl(i,1) + 2*Ct(i,1) + Cd(i,1)
            ); %permeability of the glycocalyx in the vertical
            direction
55
56      N(i,1) = v_o(i,1)*rho_f_true*sqrt(K(i,1))/mu; %Nauman
            number
57      nondim_dp(i,1) = dp_dz(i,1)*sqrt(K(i,1))/((v_o(i,1))^2*
            rho_f_true);
58      velocityf_darcy_nd(i,1) = -N(i,1)*phi_f(i,1)*nondim_dp(i
            ,1);
59      velocityf_darcy(i,1) = velocityf_darcy_nd(i,1)*values(i,
            6)*1e6;
60  end
61
62  sum_measures = 0;
63  for j = 1:1:variables
64      contrast_odd(1,j) = 1/4*((velocityf_darcy_nd(2*variables
            +2,1) - velocityf_darcy_nd(j+variables+1,1)) + (
            velocityf_darcy_nd(j+1,1)- velocityf_darcy_nd(1,1)));
65      contrast_even(1,j) = 1/4*((velocityf_darcy_nd(2*variables
            +2,1) - velocityf_darcy_nd(j+variables+1,1)) - (
            velocityf_darcy_nd(j+1,1)- velocityf_darcy_nd(1,1)));
66      measure(1,j) = abs(contrast_odd(1,j)) + abs(contrast_even
            (1,j));
67      sum_measures = sum_measures + measure(1,j);
68  end
69
```

```
for j = 1:1:variables
      sens(1,j) = measure(1,j)/sum_measures;
end

%Sensitivity analysis for the darcy velocity when the
   glycocaylx is very
%small
figure(1)
bar(sens(1,:), 0.4)
xlabel('Parameter')
ylabel('Sensitivity')
title('Darcy Velocity Sensitivity Analysis (Glycocaylx)')
hold on
plot(xlim, [.1667, .1667], ':')
names = {'Dv', 'Lv', 'Dh', 'Lh', 'dP/dz', 'v_{o}'};
set(gca, 'xticklabel', names)
```

A.6 Poiseuille Velocity and Shear Stress Sensitivity Analysis

```
%Variable input: Low, high inputs for each n number variables
   . Input such
%that first and second inputs are low and then high value of
   1st variable (respectively),
%third and fourth inputs are the low and high of the 2nd
   variable, etc.

varargin = [0.245 0.819 75e-9 422e-9 -11.87e3/6e-3 -15.87e3/6
   e-3 21.6e-6 84e-6];
variables = length(varargin)/2; %number of variables is equal
    to the number of inputs/2
```

```
7  n = 2*variables + 2; %number of trials or simulations that
       need to be performed
8
9  %Creates a matrix to be used to extract the low variables
10 all_low = zeros(1, variables); %creates row vector of just
       zeros
11 low_identity = eye((n/2)-1, variables); %creates identity
       matrix
12 low_matrix = vertcat(all_low, low_identity); %combines zero
       and identity matrices
13
14 %Creates a matrix to be used to extract the high variables
15 all_high = zeros(1, variables); %creates row vector of just
       zeros
16 high_identity = -1*eye((n/2)-1, variables); %creates identity
        matrix
17 high_matrix = vertcat(high_identity, all_high); %combines
       zero and identity matrices
18
19 %Creates the matrix consisting of the high and low input
       values, in the
20 %necessary format for Cotter's Method
21 for i = 1:1:n/2
22     for m = 1:1:variables
23         values(i, m) = varargin(m + (m-1) + low_matrix(i, m))
               ;
24         values(i + n/2, m) = varargin(2*m + high_matrix(i, m)
               );
25     end
```

```
26 end
27
28 %Define constants within the system (fluid = PBS)
29 mu = 0.0007; %viscosity of the fluid
30 rho_f_true = 1000; %true density of the fluid
31
32 %Looping through all of the trials
33 for i = 1:1:n
34     %Depends upon order the variables were input into the
       function
35     ab(i,1) = values(i,1); %radius of the osteocyte cell
       process
36     b(i,1) = values(i,2); %radius of the canaliculus
37     dp_dz(i,1) = values(i,3); %pressure gradient down the
       length of the canaliculus
38     v_o(i,1) = values(i,4); %initial velocity of the fluid
39
40     a(i,1) = ab(i,1)*b(i,1);
41     nondim_dp_dz(i,1) = (dp_dz(i,1)*a(i,1))/(rho_f_true*v_o(i
       ,1)^2);
42     nondim_b(i,1) = b(i,1)/a(i,1);
43     nondim_a(i,1) = a(i,1)/a(i,1);
44
45     %Determines fluid velocity and shear stress as a function
        of symbolic radius
46     %syms radius;
47     syms nondim_radius;
48     h1(i,1) = -a(i,1)*(nondim_b(i,1))^2*rho_f_true*v_o(i,1)*
       nondim_dp_dz(i,1)/4/mu;
```

```
49      h2(i,1) = (nondim_radius/nondim_b(i,1))^2;
50      h3(i,1) = ((nondim_a(i,1)/nondim_b(i,1))^2 - 1)/(log(
            nondim_radius/nondim_b(i,1)));
51      h4(i,1) = log(nondim_radius/nondim_b(i,1));
52
53      nondim_sym_velocityf(i,1) = h1(i,1)*(h2(i,1)-(h3(i,1)*h4(
            i,1))-1);
54      nondim_sym_shear(i,1) = (1/rho_f_true)*(1/v_o(i,1))*(1/a(
            i,1))*mu*diff(nondim_sym_velocityf(i,1), nondim_radius
            ); %differentiates (with respect to radius) fluid
            velocity equation to create shear stress equation as a
             function of radius
55
56      nondim_radiusV(i,1) = (nondim_b(i,1) + nondim_a(i,1)/2);
57      nondim_radiusS(i,1) = a(i,1)/a(i,1);
58
59      nd_velocityf(i,1) = subs(nondim_sym_velocityf(i,1),
            nondim_radiusV(i,1));
60      velocityf(i,1) = nd_velocityf(i,1)*v_o(i,1);
61      nd_shear(i,1) = subs(nondim_sym_shear(i,1),
            nondim_radiusS(i,1)); %substitutes in cell process
            radius to numerically determine value of shear stress
62      shear(i,1) = nd_shear(i,1)*rho_f_true*v_o(i,1)^2;
63  end
64  actual_velocityf(:,1) = double(velocityf(:,1))*1e6;
65  ndd_velocityf(:,1) = double(nd_velocityf(:,1));
66  actual_shear(:,1) = double(shear(:,1));
67  ndd_shear(:,1) = double(nd_shear(:,1));
68
```

```
69 sum_measures_v = 0;
70 for j = 1:1:variables
71     contrast_odd_v(1,j) = 1/4*((ndd_velocityf(2*variables
          +2,1) - ndd_velocityf(j+variables+1,1)) + (
          ndd_velocityf(j+1,1)- ndd_velocityf(1,1)));
72     contrast_even_v(1,j) = 1/4*((ndd_velocityf(2*variables
          +2,1) - ndd_velocityf(j+variables+1,1)) - (
          ndd_velocityf(j+1,1)- ndd_velocityf(1,1)));
73     measure_v(1,j) = abs(contrast_odd_v(1,j)) + abs(
          contrast_even_v(1,j));
74     sum_measures_v = sum_measures_v + measure_v(1,j);
75 end
76
77 for j = 1:1:variables
78      sens_v(1,j) = measure_v(1,j)/sum_measures_v;
79 end
80
81 sum_measures = 0;
82 for j = 1:1:variables
83     contrast_odd(1,j) = 1/4*((ndd_shear(2*variables+2,1) -
          ndd_shear(j+variables+1,1)) + (ndd_shear(j+1,1)-
          ndd_shear(1,1)));
84     contrast_even(1,j) = 1/4*((ndd_shear(2*variables+2,1) -
          ndd_shear(j+variables+1,1)) - (ndd_shear(j+1,1)-
          ndd_shear(1,1)));
85     measure(1,j) = abs(contrast_odd(1,j)) + abs(contrast_even
          (1,j));
86     sum_measures = sum_measures + measure(1,j);
87 end
```

```
88
89  for j = 1:1:variables
90        sens(1,j) = measure(1,j)/sum_measures;
91  end
92
93  %Sensitivity analysis for the shear stress due to poiseuille
       flow
94  figure(1)
95  bar(sens(1,:), 0.4)
96  xlabel('Parameter')
97  ylabel('Sensitivity')
98  title('Poiseuille Shear Stress Sensitivity Analysis (No
       Glycocaylx)')
99  hold on
100 plot(xlim, [.25, .25], ':')
101 names = {'a/b', 'b', 'dP/dz', 'v_{o}'};
102 set(gca, 'xticklabel', names)
103
104 %Sensitivity analysis for the poiseuille velocity
105 figure(2)
106 bar(sens_v(1,:), 0.4)
107 xlabel('Parameter')
108 ylabel('Sensitivity')
109 title('Poiseuille Velocity Sensitivity Analysis (No
       Glycocaylx)')
110 hold on
111 plot(xlim, [.25, .25], ':')
112 names = {'a/b', 'b', 'dP/dz', 'v_{o}'};
113 set(gca, 'xticklabel', names)
```

A.7 Poiseuille Velocity

```
%Defining variables needed for poiseuille velocity profile
%Canaliculus radius (b) (m)
b_low = 75e-9;
b_high = 422e-9;
b_avg = (b_low + b_high)/2;

%Ratio of osteocyte cell process (a) to canaliculus radius (b
    )
ratio_ab_low = 0.245;
ratio_ab_high = 0.819;

%Initial velocity (m/s)
v_o_low = 21.6e-6;
v_o_high = 84e-6;
v_o_avg = (v_o_low + v_o_high)/2;

%Pressure gradient down the canaliculus (Pa/m)
dp_dz_low = -11.87e3/6e-3;
dp_dz_high = -15.87e3/6e-3;
dp_dz_avg = (dp_dz_low + dp_dz_high)/2;

%Viscosity of the fluid (Pa s)
mu = 0.0007;
%True density of the fluid (kg/m^3)
rho_f_true = 1000;

i=1;
```

```
27 for ab = ratio_ab_low:0.05:ratio_ab_high
28     ratio(i,1) = ab;
29     %Nondimensionalization of input variables
30     nondim_dp_dz(i,1) = (dp_dz_avg*ratio(i,1)*b_avg)/(
           rho_f_true*v_o_avg^2);
31     nondim_b(i,1) = b_avg/(ratio(i,1)*b_avg);
32     nondim_a(i,1) = (ratio(i,1)*b_avg)/(ratio(i,1)*b_avg);
33
34     %Calculates nondimensionalized and dimensionalized
           maximum velocity in
35     %the middle of the annulus gap as a function of a
           decreasing gap
36     nondim_radius(i,1) = ((b_avg + ratio(i,1)*(b_avg))/2)/(
           ratio(i,1)*b_avg);
37     h1(i,1) = -(ratio(i,1)*b_avg)*(nondim_b(i,1))^2*
           rho_f_true*v_o_avg*nondim_dp_dz(i,1)/4/mu;
38     h2(i,1) = (nondim_radius(i,1)/nondim_b(i,1))^2;
39     h3(i,1) = ((nondim_a(i,1)/nondim_b(i,1))^2 - 1)/(log(
           nondim_radius(i,1)/nondim_b(i,1)));
40     h4(i,1) = log(nondim_radius(i,1)/nondim_b(i,1));
41     nondim_velocityf(i,1) = h1(i,1)*(h2(i,1)-(h3(i,1)*h4(i,1)
           )-1);
42     velocityf(i,1) = nondim_velocityf(i,1)*v_o_avg*1e6;
43     i=i+1;
44 end
45
46 ab = ratio_ab_low:0.05:ratio_ab_high;
47
48 %Nondimensionalized maximum poiseuille velocity
```

```
49 figure(1)
50 plot(ab, nondim_velocityf);
51 xlabel('Ratio of Osteocyte Process to Canaliculus (a/b)')
52 ylabel('Nondimensionalized Maximum Velocity')
53 title('Nondimensionalized Poiseuille Velocity Profile (No
       Glycocaylx)')
54
55 %Re-dimensionalized maximum poiseuille velocity
56 figure(2)
57 plot(ab, velocityf);
58 xlabel('Ratio of Osteocyte Process to Canaliculus (a/b)')
59 ylabel('Maximum Velocity (\mum/s)')
60 title('Poiseuille Velocity Profile (No Glycocaylx)')
```

B. 3D RECONSTRUCTION SOURCE CODE

B.1 LCS 3D Reconstruction

```
1  function [ImgBlock, blobBlock,fiberBinary] = TiffStacker( )
2  %Reconstructs CLSM imaging z-stacks into 3D volume
3  %   Using TIF images from CLSM imaging of the lacunar-
       canalicular system (LCS) of cortical
4  %   bone, the 2D data is noise and intensity filtered before
       segmentation into lacunae and
5  %   canaliculi. Validation of this segmentation process can
       be validated using a raw
6  %   dataset and its manual reconstruction. The LCS is then
       reconstructed in 3D and
7  %   important morphological characteristics are output to a .
       txt file. In
8  %   this code, lacunae are typically referred to as blobs and
        canaliculi
9  %   are referred to as fibers.
10
11     %% Parameters for dataset under analysis (can be moved to
           function inputs if desired)
12     xyRes = 0.04; % um   (assumes x & y axis have the same
          resolution)
13     zRes = 0.394;  % um
14     channel2use = 2; % Channel of tiff image(s) where data is
           located
```

```
canFiltDia = 0.8;  % um Max diameter of canaliculai in the dataset under
                   %   analysisThis value is used to segment lacunae
                   %   from canaliculai
blobMinArea = 5000; % Minimum area that a lacunae must have to not be noise

display3D = 0;      % Display 3D representation of the target lacunae and neighboring fibers
displayRaw = 0;     % Display stack of raw data slices (takes a lot of memory, never save if you can avoid it)
display3DCan=1;     % Display 3D represntation of final canaliculi and lacuna (takes hours)
runValidation = 0;  % Perform validation on current dataset with a second 'true' dataset
smooth3D = 1;       % Performs 3D smoothing on the dataset before 3D skeletonization

%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Load all images in the folder and calculate thresholds
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Load local supporting files to path
addpath(genpath('.'));
```

```
32
33      % Get data set from user selection
34      [~,PathName] = uigetfile({'*.tiff;*.tif'},'Select the
            first image file in sequence to load');
35      dirData = dir(PathName);                    % Get the data
            for the current directory
36      dirIndex = [dirData.isdir];                 % Find the index
            for directories
37      fileList = {dirData(~dirIndex).name}';      % Get a list of
            the files
38      numSlices = numel(fileList);
39      tic
40
41      ImgStack = cell(numSlices,1);
42      ImgStackOrig = cell(numSlices,1);
43
44      for i = 1:numSlices
45          % Read in slice image file
46          curImgName = strcat(PathName,fileList{i});
47
48          ImgOrig = imread(curImgName);
49          ImgOrig = im2double(ImgOrig);
50
51          % Flatten Tiff images to greyscale values
52          curImgSize = size(ImgOrig);
53          sizeCheck = size(curImgSize);
54          if(sizeCheck(2)>2)
55              if(channel2use==0)
56                  ImgNew = ImgOrig(:,:,1);
```

```
                ImgNew = ImgNew + ImgOrig(:,:,2);
                ImgNew = ImgNew + ImgOrig(:,:,3);
                ImgOrig = ImgNew;
            else
                ImgOrig = ImgOrig(:,:,channel2use);
            end
        end

        % Save slice to stack
        ImgStack{i} = ImgOrig;
        ImgStackOrig{i} = ImgOrig;
    end

    %% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % Perform simple intensity correction
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    ImgBlock = zeros(size(ImgStack{1},1),size(ImgStack{1},2),numSlices);
    for i = 1:numel(ImgStack)
        ImgBlock(:,:,i) = ImgStack{i};
    end
    clear ImgStack


    for i = 1:size(ImgBlock,3)
        ImgBlock(:,:,i) = imadjust(ImgBlock(:,:,i));
    end


```

```
85      sliceAve = mean(mean(ImgBlock,1),2);
86      sliceAve = reshape(sliceAve,size(ImgBlock,3),1);
87      stackAve = mean(mean(mean(ImgBlock)));
88      sliceAdj = stackAve./sliceAve;
89      for i = 1:size(ImgBlock,3)
90          ImgBlock(:,:,i) = ImgBlock(:,:,i).*sliceAdj(i);
91      end
92
93      mVal = max(max(max(ImgBlock)));
94      ImgBlock = ImgBlock./mVal;
95
96
97      opt.kernelratio=4;
98      opt.windowratio=4;
99      opt.verbose=false;
100     for i = 1:size(ImgBlock,3)
101         ImgBlock(:,:,i) =NLMF(ImgBlock(:,:,i),opt);
102     end
103
104     if(displayRaw)
105         figure('Name','Raw Image Stack','NumberTitle','off')
              % Display processed fibers
106         imshow3D(ImgBlock)
107         drawnow
108     end
109
110     %% Segment blobs (lacunae) and fibers (canaliculi)
111     % Analyze fibers sections of image utilizing a Hessian-
          based multiscale filter
```

```
112     fibreBlock = ImgBlock;
113     fibreBlock = fibreBlock*255; % Adjust fiber block values
114     fibBlock2D = ones(size(fibreBlock));
115     fibBlock2DInverse = ones(size(fibreBlock));
116
117
118    % Set up parameters for multiscale filter
119     options.FrangiScaleRange = [1 8];
120     options.FrangiScaleRatio= 0.5;
121     options.FrangiBetaOne= 2;
122     options.FrangiBetaTwo= 15;
123     options.BlackWhite= true;
124     options.verbose= false;
125     for i = 1:size(fibreBlock,3)
126         slice = FrangiFilter2D(fibreBlock(:,:,i),options);
127         fibBlock2D(:,:,i) = slice;
128     end
129     options.BlackWhite= false;
130     for i = 1:size(fibreBlock,3)
131         slice = FrangiFilter2D(fibreBlock(:,:,i),options);
132         fibBlock2DInverse(:,:,i) = slice;
133     end
134
135
136     fiberBinary = false(size(fibreBlock)); % Convert to
        binary block
137     fiberBinary(fibBlock2D==0)=1;
138     fibFix = strel('disk',4);
139     fiberBinary = imclose(fiberBinary,fibFix);
```

```
140
141
142     maskingBinary = false(size(fiberBinary));
143     maskingBinary(fibBlock2DInverse>0.005)=1;
144     fiberBinary(maskingBinary==0)=0;
145     clear maskingBinary
146
147
148     params.bz_thresh = -0; % Perform smoothing on fiber
            regions
149     params.as_scale = 1/4;
150     params.debug = 0;
151     for i = 1:size(fibreBlock,3)
152         fiberBinary(:,:,i) = bwsmooth(fiberBinary(:,:,i),
                params);
153     end
154
155     % Save full combined mask
156     BinaryStack = fiberBinary;
157
158     % Separate blob and fiber regions
159     voxFiltDia = round(canFiltDia/xyRes);
160     blobSe1 = strel('disk',voxFiltDia);
161     blobBlockStack = fiberBinary;
162     blobMask = imopen(fiberBinary,blobSe1);
163     blobSe12 = strel('disk',round(voxFiltDia/2));
164     blobMask = imdilate(blobMask,blobSe12);
165     blobBlockStack(blobMask==0)=0;
166     blobBlockStack = bwareaopen(blobBlockStack,blobMinArea);
```

```
167     clear fibreBlock
168     clear fibBlock2DInverse
169     clear fibBlock2D
170
171     fiberBinary(blobBlockStack==1)=0; % Remove blob regions
            and clean up fibers
172
173     figure('Name','Processed canaliculi','NumberTitle','off')
            % Display processed fibers
174     imshow3D(fiberBinary)
175     drawnow
176
177     figure('Name','Processed lacunae','NumberTitle','off') %
            Display processed fibers
178     imshow3D(blobBlockStack)
179     drawnow
180
181     %% Run validation comparison
182
183     if(runValidation) % Display match with validation dataset
184         expBlock = fiberBinary; % merge analyzed blobs and
                fibers
185         expBlock(blobBlockStack)=1;
186         canDil = strel('disk',5);
187
188
189         % Load and prepare validtation dataset
190         [~,PathName] = uigetfile({'*.tiff;*.tif;*.jpg'},'
                Select the first image file in sequence to load');
```

```
191         dirData = dir(PathName);                    % Get the
               data for the current directory
192         dirIndex = [dirData.isdir];                 % Find the
               index for directories
193         fileList = {dirData(~dirIndex).name}';      % Get a list
               of the files
194         numSlices = numel(fileList);
195
196         %Load images in validation dataset folder
197         ImgStackV = cell(numSlices,1);
198
199         ImgStackOrig = cell(numSlices,1);
200         for i = 1:numSlices
201             % Read in slice image file
202             curImgName = strcat(PathName,fileList{i});
203
204             ImgOrig = imread(curImgName);
205             ImgOrig = im2double(ImgOrig);
206
207             % Flatten Tiff images to greyscale values
208             curImgSize = size(ImgOrig);
209             sizeCheck = size(curImgSize);
210             if(sizeCheck(2)>2)
211                 ImgOrig = ImgOrig(:,:,1);
212             end
213
214             % Save slice to stack
215             ImgStackV{i} = ImgOrig;
216             ImgStackOrig{i} = ImgOrig;
```

```
217         end
218
219         ImgBlockV = false(size(ImgStackV{1},1),size(ImgStackV
                {1},2),numSlices);
220         for i = 1:numSlices
221             ImgBlockV(:,:,i) = ImgStackV{i};
222         end
223
224         allVals= reshape(ImgBlockV,1,[]);
225         allVals(isnan(allVals))=[];
226         stackMean = mean(allVals);
227
228         % Perform thresholding on the image block
229         fiberValid = false(size(ImgBlockV));
230         fiberValid(ImgBlockV>=stackMean)=1;
231         fiberValid = imopen(fiberValid,strel('disk',3));
232
233
234         % Segment cell bodies, 'aka' blobs
235         blobValid = imopen(fiberValid,blobSe1);
236         blobValid = imdilate(blobValid,canDil);
237         fiberValid(blobValid)=0;
238
239
240         % Compare processed blobs and fibers to validation
                dataset
241         validationBlock = fiberValid; % merge analyzed blobs
                and fibers
242         validationBlock(blobValid)=1;
```

```
243
244
245         % Check accuracy of entire segmentation to raw data
246         overlapBlock = expBlock;
247         overlapBlock(validationBlock==0)=0; % overlap of two
                datasets
248         evBlock = expBlock;
249         evBlock(overlapBlock==1)=0;
250         veBlock = validationBlock;
251         veBlock(overlapBlock==1)=0;
252         overlap = sum(sum(sum(overlapBlock)));
253         expAlone = sum(sum(sum(evBlock)));
254         validAlone = sum(sum(sum(veBlock)));
255         overalAccuracy = overlap/(overlap+expAlone+validAlone
                );
256         textO = ['Overall Accuracy: ',num2str(overalAccuracy)
                ];
257         disp(textO);
258
259         % Display overall overlap
260         figure('Name','Dataset Overlap [Red-Validation, Blue-
                Processed]','NumberTitle','off')
261         DataBlock = ones([size(ImgBlock),3]);
262         DataBlock(:,:,:,1) = validationBlock;
263         DataBlock(:,:,:,2) = expBlock;
264         imshow3D(DataBlock)
265         clear DataBlock
266
267
```

```
268         % Check accuracy of fibers
269         overlapBlockF = fiberBinary;
270         overlapBlockF(fiberValid==0)=0; % overlap of two
                fiber datasets
271         evBlockF = fiberBinary;
272         evBlockF(overlapBlockF==1)=0;
273         veBlockF = fiberValid;
274         veBlockF(overlapBlockF==1)=0;
275         overlapF = sum(sum(sum(overlapBlockF)));
276         expAloneF = sum(sum(sum(evBlockF)));
277         validAloneF = sum(sum(sum(veBlockF)));
278         fiberAccuracy = overlapF/(overlapF+expAloneF+
                validAloneF);
279         textO = ['Fiber Accuracy: ',num2str(fiberAccuracy)];
280         disp(textO);
281
282         % Display fiber overlap
283         figure('Name','Fiber Overlap [Red-Validation, Blue-
                Processed]');%,'NumberTitle','off')
284         DataBlock = ones([size(ImgBlock),3]);
285         DataBlock(:,:,:,1) = fiberValid;
286         DataBlock(:,:,:,2) = fiberBinary;
287         imshow3D(DataBlock)
288         clear DataBlock
289
290
291         % Check accuracy of blobs
292         overlapBlockB = blobBlockStack;
```

```
293         overlapBlockB(blobValid==0)=0; % overlap of two blob
                datasets
294         evBlockB = blobBlockStack;
295         evBlockB(overlapBlockB==1)=0;
296         veBlockB = blobValid;
297         veBlockB(overlapBlockB==1)=0;
298         overlapB = sum(sum(sum(overlapBlockB)));
299         expAloneB = sum(sum(sum(evBlockB)));
300         validAloneB = sum(sum(sum(veBlockB)));
301         blobAccuracy = overlapB/(overlapB+expAloneB+
                validAloneB);
302         textO = ['Blob Accuracy: ',num2str(blobAccuracy)];
303         disp(textO);
304
305         % Display blob overlap
306         figure('Name','Blob Overlap [Red-Validation, Blue-
                Processed]','NumberTitle','off')
307         DataBlock = ones([size(ImgBlock),3]);
308         DataBlock(:,:,:,1) = blobValid;
309         DataBlock(:,:,:,2) = blobBlockStack;
310         imshow3D(DataBlock)
311         clear DataBlock
312     end
313     clear ImgBlock
314     clear ImgOrig
315
316     % Skeletonize 2D Sices to determine diameter of fibers
317     skelStack = false(size(fiberBinary));
318
```

```
319     for i = 1:size(fiberBinary,3)
320         skelStack(:,:,i) = bwmorph(skeleton(fiberBinary(:,:,i
                ))>35,'skel',Inf); % Code spends 30 minutes here
321     end
322     fullCNT =sum(sum(sum(fiberBinary)));
323     skelCNT = sum(sum(sum(skelStack)));
324
325     % assuming normally circular radius the diameter shoud be
            the total
326     % fiber volume divide by the length of the skeleton
327     fibDia = round(fullCNT/skelCNT);
328
329
330     %% Calculate fibre diameter for regions of source image
            dataset
331     % Find nodes
332     SegmentData = cell(size(skelStack,3),1);
333     maskDil = strel('disk',round(fibDia*1.5));
334     for sli = 1:size(skelStack,3)
335         branchPoints = bwmorph(skelStack(:,:,sli),'
                branchpoints');
336
337         % Find sections (construct sectional list)
338         allSegment = skelStack(:,:,sli);
339         allSegment(branchPoints==1)=0;
340         segments = bwconncomp(allSegment);
341         sliceData = zeros(segments.NumObjects,2);
342
343         for i = 1:segments.NumObjects
```

```
344             % Mask region for analysis
345             curSkel = false(size(allSegment));
346             curSkel(segments.PixelIdxList{i}) = 1;
347             curMask = imdilate(curSkel,maskDil);
348
349             curSegment = fiberBinary(:,:,sli);
350             curSegment(curMask==0)=0;
351
352             % Make sure analysis is capturing only one region
                    (no neighbors)
353             stats = regionprops(curSegment,'PixelList','Area'
                   );
354             if(numel(stats)>2)
355                 curSegment = false(size(curSegment));
356                 largest = 1;
357                 curStats = stats(1);
358                 area1 = curStats.Area;
359                 for k = 2:size(stats,1)
360                     curStats = stats(k);
361                     if(curStats.Area>area1)
362                         largest = k;
363                         area1 = curStats.Area;
364                     end
365                 end
366                 curPixels = stats(largest).PixelList;
367                 for m = 1:size(curPixels,1)
368                     curSegment(curPixels(m,2),curPixels(m,1))
                            =1;
369                 end
```

```
370                 end
371                 segDiam = sum(sum(curSegment))/sum(sum(curSkel));
372
373                 sliceData(i,1) = sum(sum(curSkel));
374                 sliceData(i,2) = segDiam;
375             end
376             SegmentData{sli} = sliceData;
377         end
378         % Convert data structure to array
379         fiberLenDia = cell2mat(SegmentData);
380
381         % Remove short fiber lengths, invalid readings/noise
382         lenThresh = 10;
383         fiberLenDia(fiberLenDia(:,1)<lenThresh,:) = [];
384
385         % Convert values from pixels to um
386         fiberLenDiaUM = fiberLenDia.*xyRes;
387         disp(['Average canaliculi diameter: ',num2str(mean(
                fiberLenDiaUM(:,2))),' um'])
388         OuputData.fiberLenDiaUM = fiberLenDiaUM; % Save fiber
                diameter data
389
390         clear fiberBinary
391         clear skelStack
392
393         %% Adjust data to be isometric
394         % Thicken binary image blocks to make voxels isotropic
395         sFactor = round(zRes/xyRes);
```

```
396     newBlock = false(size(BinaryStack,1),size(BinaryStack,2),
            size(BinaryStack,3)*sFactor);
397     for i = 1:numSlices
398         startInd = ((i-1)*sFactor)+1;
399         endInd = (i*sFactor)+1;
400         for k = startInd:endInd
401             newBlock(:,:,k) = BinaryStack(:,:,i);
402         end
403     end
404     BinaryStack = newBlock;
405     clear newBlock
406
407
408
409     %% Smooth the 3D fibres & Blobs in the datablock
410     if(smooth3D)
411         BinaryStack = imclose(BinaryStack,strel3d(round(
                sFactor/2))); % smooth 3D geometry
412     end
413     for i = 1:size(BinaryStack,1)
414         temp = BinaryStack(i,:,:);
415         temp = reshape(temp,size(temp,2),size(temp,3));
416         BinaryStack(i,:,:) = bwsmooth(temp, params);
417     end
418     for i = 1:size(BinaryStack,2)
419         temp = BinaryStack(:,i,:);
420         temp = reshape(temp,size(temp,1),size(temp,3));
421         BinaryStack(:,i,:) = bwsmooth(temp, params);
422     end
```

```
423
424     %% Re-segment blobs from fibers in 3D post smoothing
425     % Have to do this post 3D thickening and smoothing so
            that the fibers
426     % don't end up massively disconnected from the blob
            regions
427     blobMask = imopen(BinaryStack,strel3d(voxFiltDia)); %
            Segment blobs
428     blobMask = fastDilate( blobMask,sFactor);
429     blobBlock = BinaryStack;
430     blobBlock(blobMask==0)=0;
431     blobBlock = bwareaopen(blobBlock,blobMinArea*sFactor);
432     fiberBinaryIso = BinaryStack; % Assign fibers
433     fiberBinaryIso(blobBlock)=0;
434     clear blobMask
435     clear BinaryStack
436
437     %% Blob Ellipsoid analysis
438
439     blobBlock = bwareaopen(blobBlock,5000);
440     blobBlock = imopen(blobBlock,strel3d(sFactor));
441     blobBlock(1:2*sFactor,:,:)=0;
442     blobBlock(size(blobBlock,1)-2*sFactor:end,:,:)=0;
443     blobBlock(:,1:2*sFactor,:)=0;
444     blobBlock(:,size(blobBlock,2)-2*sFactor:end,:)=0;
445
446     % Select largest blob for analysis
447     stats = regionprops(blobBlock,'PixelList','Area');
448     if(numel(stats)>2)
```

```
449             blobBlock = false(size(blobBlock));
450             largest = 1;
451             curStats = stats(1);
452             area1 = curStats.Area;
453             for k = 2:size(stats,1)
454                 curStats = stats(k);
455                 if(curStats.Area>area1)
456                     largest = k;
457                     area1 = curStats.Area;
458                 end
459             end
460             curPixels = stats(largest).PixelList;
461             for m = 1:size(curPixels,1)
462                 blobBlock(curPixels(m,2),curPixels(m,1),curPixels
                        (m,3))=1;
463             end
464         end
465
466
467         % Find edge voxels
468         curEdge = imerode(blobBlock,strel3d(3));
469         blobBlockShell=blobBlock;
470         blobBlockShell(curEdge==1)=0; %change MV
471         clear curEdge
472         ind = find(blobBlockShell); %change MV
473         [yInd, xInd, zInd] = ind2sub(size(blobBlockShell), ind);
                %change MV
474
```

```
475     [ ~, radii, evecs, v, ~ ] = ellipsoid_fit( [ xInd yInd
            zInd ], '' );
476
477     mind = [ 0 0 0 ];
478     maxd = size(blobBlockShell); %change MV
479     clear blobBlockShell
480     nsteps = 100;
481     step = ( maxd - mind ) / nsteps;
482     [ x, y, z ] = meshgrid( linspace( mind(1) - step(1), maxd
            (1) + step(1), nsteps ), linspace( mind(2) - step(2),
            maxd(2) + step(2), nsteps ), linspace( mind(3) - step
            (3), maxd(3) + step(3), nsteps ) );
483
484     Ellipsoid = v(1) *x.*x +   v(2) * y.*y + v(3) * z.*z +
            ...
485         2*v(4) *x.*y + 2*v(5)*x.*z + 2*v(6) * y.*z + ...
486         2*v(7) *x    + 2*v(8)*y    + 2*v(9) * z;
487     ellipSurf = isosurface( x, y, z, Ellipsoid, -v(10));
488
489     % Display ellipsoid and sample of source datapoints
490     figure()
491     hold on
492     temp = [xInd, yInd, zInd];
493     if(numel(xInd)>2000)
494         temp = datasample(temp,2000);
495     end
496     scatter3(temp(:,1),temp(:,2),temp(:,3));
497     p = patch(ellipSurf);
498     set( p, 'FaceColor', 'g', 'EdgeColor', 'none' );
```

```
499     camlight;
500     lighting phong;
501     daspect([1,1,1])
502     axis equal
503     hold off
504     drawnow
505
506     radiiO = radii;
507     radii = abs(radii);
508     if(radii(1)>radii(2) && radii(1)>radii(3)) % 1 is long
            axis
509         longAxis = radii(1);
510         if(radii(2)>radii(3))
511             sAxis1 = radii(2);
512             sAxis2 = radii(3);
513         else
514             sAxis1 = radii(3);
515             sAxis2 = radii(2);
516         end
517         longOrient = evecs(:,1);
518     elseif(radii(2)>radii(1) && radii(2)>radii(3)) % 2 is
            long axis
519         longAxis = radii(2);
520         if(radii(1)>radii(3))
521             sAxis1 = radii(1);
522             sAxis2 = radii(3);
523         else
524             sAxis1 = radii(3);
525             sAxis2 = radii(1);
```

```
526         end
527         longOrient = evecs(:,1);
528     else % 3 is long axis
529         longAxis = radii(3);
530         if(radii(1)>radii(2))
531             sAxis1 = radii(1);
532             sAxis2 = radii(2);
533         else
534             sAxis1 = radii(2);
535             sAxis2 = radii(1);
536         end
537         longOrient = evecs(:,1);
538     end
539
540     longAxisUM = longAxis*xyRes;
541     sAxis1UM = sAxis1*xyRes;
542     sAxis2UM = sAxis2*xyRes;
543
544     % Orientation of long axis with respect to z-axis (in
          degrees)
545     theta = atan((sqrt(longOrient(1,1)^2 + longOrient(2,1)^2)
          )/longOrient(3,1))*57.2958;
546
547     % Calculate sphericity
548         lacSphericity = sAxis2/longAxis;
549     lacSurfArea = 4*pi*((((longAxisUM*sAxis1UM)^1.6+(
          longAxisUM*sAxis2UM)^1.6+(sAxis1UM*sAxis2UM)^1.6)/3)
          ^(1/1.6));
550     lacVolume = (4/3)*pi*longAxisUM*sAxis1UM*sAxis2UM;
```

```
551     lacVolumeTrue = sum(sum(sum(blobBlock)))*xyRes*xyRes*
            xyRes;
552     lacOblat = 2*((sAxis1-longAxis)/(sAxis2-longAxis)) -1;
553     disp(['Lacunae Sphericity: ',num2str(lacSphericity)])
554     disp(['Lacunae Surface Area: ',num2str(lacSurfArea),' um
            ^2'])
555     disp(['Lacunae Ellip Volume: ',num2str(lacVolume),' um^3'
            ])
556     disp(['Lacunae True Volume: ',num2str(lacVolumeTrue),' um
            ^3'])
557     disp(['Lacunae Oblateness: ', num2str(lacOblat)])
558
559     % Construct volume block of ellipsoid
560     ellipseSolid = polygon2voxel(ellipSurf,size(blobBlock),'
            none');
561     for i = 1:size(ellipseSolid,3)
562         ellipseSolid(:,:,i) = imfill(ellipseSolid(:,:,i),'
                holes');
563     end
564     clear ellipSurf
565
566     % Create shell region around ellipsoid to search for
            connecting
567     % canaliculi
568     shellThick = 2*round(sAxis2);
569     % lacunaShell = imdilate(blobBlock,strel3d(shellThick));
570     lacunaShell = fastDilate( blobBlock,shellThick);
571
572
```

```
573     % Skeletonize the fibres
574     fSkel = Skeleton3D(fiberBinaryIso); % Code spends 26
            minutes here
575
576     w = size(fSkel,1);
577     l = size(fSkel,2);
578     h = size(fSkel,3);
579
580     % Initial step: condense, convert to voxels and back
581     minBranchLen = 4*sFactor;
582     [~,node,link] = Skel2Graph3D(fSkel,minBranchLen);
583
584     % Total length of network
585     wl = sum(cellfun('length',{node.links}));
586
587     skel2 = Graph2Skel3D(node,link,w,l,h);
588     [~,canNode,canLink] = Skel2Graph3D(skel2,0);
589
590     % Calculate new total length of network
591     wl_new = sum(cellfun('length',{canNode.links}));
592
593     % Iterate the same steps until network length changed by
            less than 0.5%
594     while(wl_new~=wl)
595         wl = wl_new;
596
597         skel2 = Graph2Skel3D(canNode,canLink,w,l,h);
598         [~,canNode,canLink] = Skel2Graph3D(skel2,
                minBranchLen);
```

```
599
600             wl_new = sum(cellfun('length',{canNode.links}));
601         end
602         skel3 = Graph2Skel3D(canNode,canLink,w,l,h);
603         OuputData.canNodes = canNode;
604         OuputData.canLinks = canLink;
605
606         % Display result
607         debug = 0;
608         if(debug==1)
609             figure();
610             hold on;
611             for i=1:length(canNode)
612
613                 x1 = canNode(i).comx;
614                 y1 = canNode(i).comy;
615                 z1 = canNode(i).comz;
616
617                 if(canNode(i).ep==1)
618                     ncol = 'c';
619                 else
620                     ncol = 'y';
621                 end
622
623                 for j=1:length(canNode(i).links)     % draw all
                        connections of each node
624                     if(canNode(canNode(i).conn(j)).ep==1)
625                         col='k'; % branches are black
626                     else
```

```
627                     col='r'; % links are red
628                 end
629                 if(canNode(i).ep==1)
630                     col='k';
631                 end
632
633
634                 % Draw edges as lines using voxel positions
635                 for k=1:length(canLink(canNode(i).links(j)).
                        point)-1
636                     [x3,y3,z3]=ind2sub([w,l,h],canLink(
                            canNode(i).links(j)).point(k));
637                     [x2,y2,z2]=ind2sub([w,l,h],canLink(
                            canNode(i).links(j)).point(k+1));
638                     line([y3 y2],[x3 x2],[z3 z2],'Color',col,
                            'LineWidth',2);
639                 end
640             end
641
642             % Draw all nodes as yellow circles
643             plot3(y1,x1,z1,'o','Markersize',9,...
644             'MarkerFaceColor',ncol,...
645             'Color','k');
646
647         end
648         axis image;axis off;
649         set(gcf,'Color','white');
650         drawnow;
651     end
```

```
652
653
654
655     % Find endpoints on canaliculi skeleton
656     endPts =false(size(fiberBinaryIso));
657     for i=1:length(canNode)
658         x1 = round(canNode(i).comx);
659         y1 = round(canNode(i).comy);
660         z1 = round(canNode(i).comz);
661
662         if(canNode(i).ep==1)
663             endPts(x1,y1,z1)=1;
664         end
665     end
666
667
668
669     % Ouput total lengths of all canaliculai
670     linkLengths = zeros(numel(canLink),1);
671     for i = 1:numel(canLink)
672         linkLengths(i) = numel(canLink(i).point);
673     end
674     disp(['Average canaliculi length: ',num2str(mean(
          linkLengths))])
675
676
677     %% Display canuliculai near the lacunae (i.e. within
          shell region)
678     if(display3D)
```

```
679         fibre2show = skel3;
680         ellipseShellLocal = false(size(lacunaShell));
681         index = find(sum(sum(lacunaShell,2),3));
682         xF = index(1)-100;
683         if(xF<1)
684           xF = 1;
685         end
686         xB = index(end)+100;
687         if(xB>size(lacunaShell,1))
688             xB = size(lacunaShell,1);
689         end
690         index = find(sum(sum(lacunaShell,1),3));
691         yF = index(1)-100;
692         if(yF<1)
693           yF = 1;
694         end
695         yB = index(end)+100;
696         if(yB>size(lacunaShell,2))
697             yB = size(lacunaShell,2);
698         end
699
700         ellipseShellLocal(xF:xB,yF:yB,:) = 1;
701         fibre2show(lacunaShell==0)=0;
702
703         figure()
704         sSurfaces = isosurface(fibre2show,0.5); %
              skeletonized fibers
705         xyz = sSurfaces.vertices;
706         tri = sSurfaces.faces;
```

```
707         trisurf(tri,xyz(:,1),xyz(:,2),xyz(:,3),'FaceColor','
               red','EdgeColor','none','FaceAlpha',1)
708         daspect([1,1,1]); view(3); axis tight; camlight;
               lighting gouraud % Format view
709         hold on
710
711         sSurfaces = isosurface(blobBlock,0.5); % simplified
               lacunae
712         xyz = sSurfaces.vertices;
713         tri = sSurfaces.faces;
714         trisurf(tri,xyz(:,1),xyz(:,2),xyz(:,3),'FaceColor','
               yellow','EdgeColor','none','FaceAlpha',0.3)
715
716         fiberBinaryLocal = fiberBinaryIso;
717         fiberBinaryLocal(lacunaShell==0)=0;
718         sSurfaces = isosurface(fiberBinaryLocal,0.5); %
               simplified canaliculi
719         xyz = sSurfaces.vertices;
720         tri = sSurfaces.faces;
721         trisurf(tri,xyz(:,1),xyz(:,2),xyz(:,3),'FaceColor','
               green','EdgeColor','none','FaceAlpha',0.5)
722
723
724 %         for i=1:length(canNode) % end points
725 %             x1 = round(canNode(i).comx);
726 %             y1 = round(canNode(i).comy);
727 %             z1 = round(canNode(i).comz);
728 %
729 %             if(canNode(i).ep==1)
```

```
730 %                     ellipseShellT = lacunaShell;
731 %                     ellipseShellT(x1,y1,z1)=0;
732 %                     if(sum(sum(sum(ellipseShellT)))<sum(sum(sum
    (lacunaShell))))
733 %                         plot3(y1,x1,z1,'o','Markersize',5,...
734 %                         'MarkerFaceColor','c',...
735 %                         'Color','k');
736 %                     end
737 %                 end
738 %             end
739 %
740 %             % color all non-connected endpoints that are nearby
741 %             lacunaLocalCK = ellipseShellLocal;
742 %             lacunaLocalCK(lacunaShell)=0;
743 %             for i=1:length(canNode) % end points
744 %                 x1 = round(canNode(i).comx);
745 %                 y1 = round(canNode(i).comy);
746 %                 z1 = round(canNode(i).comz);
747 %
748 %                 if(canNode(i).ep==1)
749 %                     ellipseShellT = lacunaLocalCK;
750 %                     ellipseShellT(x1,y1,z1)=0;
751 %                     if(sum(sum(sum(ellipseShellT)))<sum(sum(sum
    (lacunaLocalCK))))
752 %                         plot3(y1,x1,z1,'o','Markersize',5,...
753 %                         'MarkerFaceColor','y',...
754 %                         'Color','k');
755 %                     end
756 %                 end
```

```
757 %           end
758             hold off
759             drawnow
760         end
761
762         % Count number of enpoints within the lacunae shell,
                assume those
763         % segments of fibres are connecting the the lacunae
764         endPts(lacunaShell==0)=0;
765         connectedEndPnts = sum(sum(sum(endPts)));
766         disp(['canaliculi connecting to lacunae: ',num2str(
                connectedEndPnts)])
767
768         % calculated connecting lacunar density
769         lacCanDensity = connectedEndPnts/lacVolumeTrue;
770         disp(['canaliculi connection density: ',num2str(
                lacCanDensity),' per um^3'])
771
772         clear lacunaLocalCK
773         clear ellipseShellLocal
774         clear lacunaShell
775         clear endPts
776         clear fiberBinaryIso
777
778         %% Thicken fiber network
                ----------------------------------------------------
779         % Create fibre diameter, either simple dilate or a ring
                around
```

```
780     fiberBinaryDil = fastDilate(skel3,round(mean(fiberLenDia
            (:,2))));
781
782     fiberBinaryDil(blobBlock==1)=0; % Keep original blob
            geometries for this masking
783
784     figure('Name','Final Canaliculi','NumberTitle','off') %
            Display processed fibers
785     imshow3D(fiberBinaryDil)
786     drawnow
787
788
789     %% Create final blocks and save data
            ----------------------------------------
790
791     combineFileName = strcat(PathName,'FiberMask.mat');
792     save(combineFileName,'fiberBinaryDil');
793
794     combineFileName = strcat(PathName,'BlobMask.mat');
795     save(combineFileName,'blobBlock');
796
797     combineFileName = strcat(PathName,'EllipseMask.mat');
798     save(combineFileName,'ellipseSolid');
799
800     combineFileName = strcat(PathName,'LacunaeMask.mat');
801     save(combineFileName,'blobBlock');
802
803     combineFileName = strcat(PathName,'FiberSkeleton.mat');
804     save(combineFileName,'skel3');
```

```
805
806     combineFileName = strcat(PathName,'OutputResults.mat');
807     save(combineFileName,'OuputData');
808
809
810
811     % Save mreasured metrics to output file
812     resultFileName = strcat(PathName,'Results.txt');
813     fid = fopen( resultFileName, 'wt' ); % Check in double
            diameter reporting
814     fprintf(fid, 'Average Canaliculi Diameter: %f um \n',mean
            (fiberLenDiaUM(:,2)));
815     fprintf(fid, 'Average Canaliculi Segment Length (3D): %f
            um \n',mean(linkLengths));
816     fprintf(fid, 'Porosity volume fraction: %f \n', sum(sum(
            sum(fiberBinaryDil)))/(size(fiberBinaryDil,1)*size(
            fiberBinaryDil,2)*size(fiberBinaryDil,3)));
817     fprintf(fid, 'Lacunar ellipsoid axis lengths(long to
            short): %f, %f, %f \n',longAxisUM,sAxis1UM,sAxis2UM);
818     fprintf(fid, 'Lacunar ellipsoid orientation: %f Degrees \
            n', theta);
819     fprintf(fid,'Lacuna Sphericity: %f \n',lacSphericity);
820     fprintf(fid,'Lacuna Oblateness: %f \n', lacOblat);
821     fprintf(fid,'Lacuna Surface Area: %f um^2 \n',lacSurfArea
            );
822     fprintf(fid,'Lacuna Volume: %f um^3 \n',lacVolume);
823     fprintf(fid,'Lacuna True Volume: %f um^3 ',lacVolumeTrue)
            ;
824
```

```
825     fprintf(fid,'Canaliculi connecting to Lacunae: %d \n',
            connectedEndPnts);
826     fprintf(fid,'Canaliculi connection density: %f per um^3
827     \n',lacCanDensity);
828     fclose(fid);
829
830
831     %% Construct 3D represntation of canaliculi and lacunae
            network
832     if(display3DCan)
833         figure()
834         fiberBinaryDisp = padarray(fiberBinaryDil,[1 1 1],0);
835         sSurfaces = isosurface(fiberBinaryDisp,0.5);
836         xyz = sSurfaces.vertices;
837         tri = sSurfaces.faces;
838         trisurf(tri,xyz(:,1),xyz(:,2),xyz(:,3),'FaceColor','
                yellow','EdgeColor','none','FaceAlpha',1)
839         daspect([1,1,1]); view(3); axis tight; camlight;
                lighting gouraud % Format view
840         clear fiberBinaryDisp
841
842         hold on
843         blobBlockDisp = padarray(blobBlock,[1 1 1],0);
844         sSurfaces = isosurface(blobBlockDisp,0.5);
845         xyz = sSurfaces.vertices;
846         tri = sSurfaces.faces;
847         trisurf(tri,xyz(:,1),xyz(:,2),xyz(:,3),'FaceColor','
                green','EdgeColor','none','FaceAlpha',1)
```

```
848             daspect([1,1,1]); view(3); axis tight; camlight;
                    lighting gouraud % Format view
849             clear blobBlockDisp
850             hold off
851         end
852
853         toc % report to user how long analysis took
854     end
```