

University of Denver

Digital Commons @ DU

Electronic Theses and Dissertations


Graduate Studies

8-2023

A Virtual Method for Establishing Femoral Stem Position in Total Hip Arthroplasty

Samuel Mattei
University of Denver

Follow this and additional works at: <https://digitalcommons.du.edu/etd>

 Part of the [Biomechanical Engineering Commons](#), and the [Biomedical Engineering and Bioengineering Commons](#)

Recommended Citation

Mattei, Samuel, "A Virtual Method for Establishing Femoral Stem Position in Total Hip Arthroplasty" (2023). *Electronic Theses and Dissertations*. 2289.
<https://digitalcommons.du.edu/etd/2289>



All Rights Reserved.

This Masters Thesis is brought to you for free and open access by the Graduate Studies at Digital Commons @ DU. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of Digital Commons @ DU. For more information, please contact jennifer.cox@du.edu, dig-commons@du.edu.

A Virtual Method for Establishing Femoral Stem Position in Total Hip Arthroplasty

Abstract

Total hip arthroplasty (THA) is one of the most successful orthopedic surgeries performed, in which the hip joint is reconstructed to improve functionality and decrease pain in the joint. Despite the success of these procedures, femoral stem misalignment remains an area that influences THA success. The relationship between the femoral stem geometry and implanted femoral stem size and orientation is underreported due to limitations in the collection of radiographic data used for clinical pre-operative templating. Furthermore, the influence of anatomic measurements on the direction and size of the femoral stem are not considered due to imaging technique and difficulties associated with reliably measuring, moreover are not reported. The overall goal was to develop a custom automated workflow to segment the intramedullary geometry and extract anatomic parameters from cadaveric femurs and to create an automated workflow for the implantation and evaluation of femoral stem positions.

Document Type

Masters Thesis

Degree Name

M.S.

First Advisor

Chadd W. Clary

Second Advisor

Casey A. Myers

Third Advisor

Paul J. Rullkoetter

Keywords

Anatomic measurements, Total hip arthroplasty, Virtual implantation

Subject Categories

Biomechanical Engineering | Biomedical Engineering and Bioengineering | Engineering | Mechanical Engineering

Publication Statement

Copyright is held by the author. User is responsible for all copyright compliance.

A Virtual Method for Establishing Femoral Stem Position in Total Hip Arthroplasty

A Thesis

Presented to

the Faculty of the Daniel Felix Ritchie School of Engineering and Computer Science

University of Denver

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Samuel Mattei

August 2023

Advisor: Dr. Chadd W. Clary, PhD

©Copyright by Samuel Mattei 2023

All Rights Reserved

Author: Samuel Mattei
Title: A Virtual Method for Establishing Femoral Stem Position in Total Hip Arthroplasty
Advisor: Dr. Chadd W. Clary, PhD
Degree Date: August 2023

Abstract

Total hip arthroplasty (THA) is one of the most successful orthopedic surgeries performed, in which the hip joint is reconstructed to improve functionality and decrease pain in the joint. Despite the success of these procedures, femoral stem misalignment remains an area that influences THA success. The relationship between the femoral stem geometry and implanted femoral stem size and orientation is underreported due to limitations in the collection of radiographic data used for clinical pre-operative templating. Furthermore, the influence of anatomic measurements on the direction and size of the femoral stem are not considered due to imaging technique and difficulties associated with reliably measuring, moreover are not reported. The overall goal was to develop a custom automated workflow to segment the intramedullary geometry and extract anatomic parameters from cadaveric femurs and to create an automated workflow for the implantation and evaluation of femoral stem positions.

Acknowledgments

I want to thank my advisor, Dr. Chadd Clary for helping me by providing guidance and expertise throughout this entire process. I would also like to thank Dr. Casey Myers for giving guidance and feedback at multiple stages throughout this process, your assistance was invaluable. I would also like to thank my other committee members Dr. Paul Rullkoetter and Dr. Matt Rutherford.

I would additionally like to thank the people I have the privilege of calling friends that I've made during this degree program. I'd like to thank Thor Andreassen and Sean Higinbotham for their willingness to assist me regardless of the time of day and to provide expertise and perspective when I needed it the most. I would also like to thank Sam, Ignacio, Gabby, Daniele, Brendan, Hannah, Kathryn, Emma, Ola, and everyone else for creating an unforgettable lab environment. My appreciation for each of you goes beyond words. I would like to thank my undergraduate research assistants Maelah and Jisu for letting me be their mentor and guide them in not only aiding my research journey but exposing them to the world of orthopedic research.

Lastly, I would like to thank my loving family, friends, and pets for keeping my lid tight for every step of this process. Thank you, Dad, Mom, Dom, Vince, Gavin, Mindy, Oink, and Snoopy. I am nothing without the strength of those around me, thank you for supporting me in every step of this journey, none of this would have been possible without you.

Table of Contents

Abstract	ii
Acknowledgments.....	iii
Table of Contents	iv
List of Figures	vi
List of Tables	ix
CHAPTER ONE: Introductory Remarks.....	1
CHAPTER TWO: Review of Pertinent Literature	7
2.1 Review of Bony Anatomy	7
2.1.1 Review of General Bony Anatomy for the Lower Limbs.....	7
2.1.2 Review of Femoral Intercortical Anatomy	14
2.2 Total Hip Arthroplasty General Information and Femoral Stem Design	16
2.2.1 Total Hip Arthroplasty Overview	16
2.2.2 Femoral Stem Design.....	20
2.3 General Preoperative Software Information	29
2.3.1 Overview of Pre-operative Software	30
2.3.2 Differences Between Pre-Operative Software	32
2.3.3 Evaluation of Templating in THA	36
CHAPTER THREE: A Virtual Method for Establishing Stem Placement in Total Hip Arthroplasty	46
3.1 Introduction.....	46
3.2 Methodology	48
Intercortical Calcar Wall Segmentation.....	51
Intramedullary Canal Segmentation	54
Femoral Anatomic Measurements	60
Automated Femoral Stem Implantation.....	64
Implantation Algorithm Validation.....	73
Statistical Analysis.....	75
3.3 Results.....	75
Anatomic Measurement	76
Implanted Restoration of Native Head Center	77
Implantation Algorithm Validation.....	86
3.4 Discussion.....	91

CHAPTER FOUR: Concluding Remarks.....	104
4.1 Final Remarks	104
4.2 Future Work	105
4.3 Key Takeaways	106
References.....	107
Appendix A: Segmentation Algorithm	114
DU_GetCalcar_Final	114
DU_GetIntercorticalMesh_Final	126
DU_MeshDiaphysis_Final.....	135
DU_GetDorr_Final	147
DU_GetBowling_Final	150
Appendix B: Implantation Algorithm.....	159
DU_ImplantFemur_Final.....	159

List of Figures

CHAPTER TWO: Review of Pertinent Literature

Figure 1: Coronal view of the proximal end of a human femur. A-Cortical Bone, B-Cancellous Bone. (image from: Martin et al 2015)	9
Figure 2:Upper) Examples of surgical landmarks on the pelvis (image from: Ait et al 2020). Lower) Degrees of freedom of the hip joint (image from: Zhang et al 2009)	11
Figure 3:Upper) Table of muscle insertions and primary actions from Neuman et al. 2010. Lower) Example of common surgical landmark sites on the proximal femur (image from: https://teachmeanatomy.info/lower-limb/bones/femur/)	13
Figure 4: Appearance of the calcar femorale on posterior dissection. F-calcar femorale: LT.lesser trochanter. (image from: Garden et al. 1961)	14
Figure 5: Structure of a long bone (image from: https://en.wikipedia.org/wiki/Diaphysis)	15
Figure 6: Visualization of THA (image from: https://orthoinfo.aaos.org/en/treatment/total-joint-replacement/)	17
Figure 7:Left) X-ray of the hip in lateral view showing a partial cemented hip replacement Right) X-ray showing an uncemented total hip replacement. (Images from: https://www.cortho.org/hip/uncemented-vs-cemented-hip-replacement/) ..	18
Figure 8: Upper) Examples of cemented femoral stem design (image from Khanuja et al 2019). Lower) Classifications of cementless femoral stems (image from Cassar-Gheiti et al 2020).....	23
Figure 9: Examples of the three Dorr classifications (image from: Khanuja et al 2019)	26
Figure 10: Examples of anatomic stems in comparison to tapered wedge femoral stems (image from Heyland et al. 2019).....	28
Figure 11: Examples of coxa vara and coxa valga femoral neck-shaft angle deformities. (Image from: https://paleyinstitute.org/centers-of-excellence/joint-health/hip-preservation/#/)	29
Figure 12. Examples of Upper) 2D radiographic templating (image from Bachour et al. 2010). Lower) 3D biplanar radiographic templating using the EOS system (image from Knafo et al. 2019).....	31
Figure 13: Upper) 2D frontal plane image (image from: Mirghaderi et al. 2022). Lower) 3D Computed Tomography scan with boney and implant geometry segmented (image from Müller et al 2011).....	34
Figure 14: Upper) Image detailing the calculation of the femoral offset (image from Sarili et al 2009). Lower) Visualization of classification of femoral stem degrees of freedom described in the anatomic coordinate system (image from Belzunce et al. 2020).....	41

CHAPTER THREE: A Virtual Method for Establishing Femoral Stem Placement in Total Hip Arthroplasty

Figure 15: Proposed workflow to accomplish both automatic segmentation of the intercortical femoral anatomy and automated femoral stem orientation and size evaluation..... 50

Figure 16: Sulcus cut and example of resampled external surface nodes on a right femur 52

Figure 17: Upper) Visualization of intercortical node segmentation Lower) The final output of the intercortical node segmentation algorithm..... 53

Figure 18: Visualization of the height range used for diaphyseal and metaphyseal classifications. 1) Lowest node on the calcar resection, 2) intersection of the diaphysis and metaphysis 10-mm distal to the lesser trochanter, 3) distal end of diaphyseal region segmentation..... 54

Figure 19: Example radial profile from a slice in the diaphysis. Hounsfield units were mapped along radial lines extending from the centroid of the bone. Inner wall of the cortex was identified using a 40% threshold between the maximum and minimum HU on the line..... 56

Figure 20: Upper) Refinement of the diaphyseal and metaphyseal medullary canal mesh from identifying raw nodes from the DICOM images, to establishing an initial mesh, to applying smoothing splines along the S-I axis of the mesh, and finally using a Fourier smoothing filter. Lower) Example of femoral segmentation with intercortical segmentation..... 59

Figure 21: Left) CTCR calculation (LT-Lesser Trochanter) Right) CI calculation (LT-Lesser Trochanter)..... 61

Figure 22: Example femur with lateral bowing GCBA and anterior bowing GSBA. (LT – Lesser Trochanter, MDC- Midpoint of Distal Condyles) 62

Figure 23: Example of LBSA (left) and FNSA (right) calculations. 64

Figure 24: Example of FNV calculation on a right femur 64

Figure 25: Size 8 Corail® 135° standard offset with standardized landmark locations labeled: 1. center of the proximal face of the trunnion, 2. the medial aspect of the neck and the junction with the stem, 3. the distal tip of femoral stem and 4. the trunnion normal vector (red). 65

Figure 26: Example of femoral stem geometry aligned with proximal femur axis (shown as black dashed line), rotationally aligned to the native femoral head, and at the inferior sulcus cut height. 66

Figure 27: Examples of valid Size 8 Corail® 135° standard offset (left) and invalid Size 13 Corail® 135° standard offset (right) in the nominal implant positions. Green indicates no overclosure, yellow indicates overclosure less than 0.5mm, and red indicates overclosure greater than 0.5mm. 67

Figure 28: Examples of size 8 Corail® 135° standard offset rotational (top) and translational (bottom) directions of perturbation about the distal tip 69

Figure 29: Examples of a valid femoral stem cross-section within the canal’s cortical wall (left) and invalid femoral stem cross-sections caused by overclosure

of the femoral stem (middle, right). Green nodes represent 85% of the range of external sulcus cut nodes.	70
Figure 30: Example of 20% and 40% fit and fill identified metrics, with no measured overclosures. Positive ML and AP distances are notated by their respective colors.....	71
Figure 31: Illustration showing the process for creating the projected calcar resection coordinate system. An example femoral head center intersection with the projected sulcus cut plane (magenta) is shown with the distances in the coordinate system to the anatomic femoral head center.	72
Figure 32: Upper) Variance in the number of valid implant alignments vs size of the implant. Lower) The number of valid alignments for the terminal stem sizes. Statistical outliers are labeled with red crosses.....	78
Figure 33: Distance to anatomic head center from the implanted head center for T, T-1, and T-2 femoral stems. Red crosses indicate outliers.	81
Figure 34: Example composite plot showing the differences in the implanted femoral head center as femoral stem size increases. An example is within the sulcus cut coordinate system.....	81
Figure 35: Distance from the implanted head to the native head center for the terminal stem size (Upper), Terminal-1 stem size (Middle), and Terminal-2 stem size (Lower)	84
Figure 36: Anatomic description of Alignment 1, Alignment 2, and the experimentally placed femoral stem.	88
Figure 37: Alignment 1 stem position in the frontal (Upper) and sagittal (Lower) planes. Gray represents the experimentally placed stems, cyan represents the color of the Alignment 1 stem for each specimen, and light blue represents the intramedullary canal mesh.	89
Figure 38: Alignment 2 stem position in the frontal (Upper) and sagittal (Lower) planes. Gray represents the experimentally placed stems, cyan represents the color of the Alignment 2 stem for each specimen, and light blue represents the intramedullary canal mesh.	90

List of Tables

CHAPTER TWO: Review of Pertinent Literature

Table 1:Horizontal Femoral Offset and Vertical Femoral Offset Literature Comparison. *Results were compared to the contralateral side of the hip instead of the planned approach with no reported standard deviations. † Results were reported as a median number wit with no standard deviation. Surgical approaches used are reported as follows: PA-Posterior approach, PLA – Posterior lateral approach, AA-anterior approach, ALA-anterolateral approach	42
Table 2:Femoral Stem Anteversion. * Results were reported as a comparison to the contralateral hip Surgical approaches used are as follows: PA-Posterior approach, PLA – Posterior lateral approach, AA-anterior approach, ALA- anterolateral approach, DLA – Direct Lateral Approach.....	43

CHAPTER THREE: A Virtual Method for Establishing Stem Position in Total Hip Arthroplasty

Table 3: Range of LHC perturbation values	69
Table 4: Implant sizes used for each specimen. * Denotes a collared stem, for purposes of this study collared stems were treated as collarless. For S221141 the left femur was excluded due to poor segmentation.	73
Table 5: Boney Parameters for Full Cohort. (CTCR- Canal-to-Calcar ratio, CI – Cortical Index, GCBA – Global Coronal Bowing Angle, GSBA – Global Sagittal Bowing Angle, LSBA – Local Sagittal Bowing Angle, FNSA – Femoral Neck Shaft Angle, FNV – Femoral Neck Version).....	76
Table 6: Upper diagonal of symmetric correlation coefficient. Red text indicates potential association and correlation. (CTCR- Canal-to-Calcar ratio, CI – Cortical Index, GCBA – Global Coronal Bowing Angle, GSBA – Global Sagittal Bowing Angle, LSBA – Local Sagittal Bowing Angle, FNSA – Femoral Neck Shaft Angle, FNV – Femoral Neck Version).....	77
Table 7: Tabulated femoral stem terminal size results. T indicates the terminal stem size, T-1 indicates the next largest fitting stem, and T-2 indicates the next largest fitting stem underneath the T-1 stem size.	79
Table 8: Best-case implantation differences in the implanted femoral head center and anatomic in projected sulcus cut coordinate system for T, T-1, and T-2 femoral stems.	80
Table 9: Discrepancy in implant size and Y direction restoration of the femoral head.....	81
Table 10: Distance between implanted and native femoral head center for T, T-1, and T-2 size stems separated by Dorr classification. The numbers under each Dorr Classification are the size of each cohort.....	82
Table 11: Correlation coefficients between anatomic measurements and changes in head center for the T, T-1, and T-2 stem sizes. Correlations of moderate or higher statistical significance are shown in red.	85

Table 12: Correlation coefficients between the anatomic description of the location of the stem from the nominal position and the anatomic measurements for the T, T-1, and T-2 stem sizes. Correlations of moderate or higher statistical significance are shown in red..... 86

Table 13: Alignment 1 for right and left legs for terminal (T) size of implants... 87

Table 14: Alignment 2 for the terminal (T) size of implants..... 87

Table 15: Differences in experimental cut plane angle and the maximum observed overclosure of the implanted experimental femoral stem..... 87

CHAPTER ONE: Introductory Remarks

1.1 Introduction

Total joint replacements are an effective solution for restoring normal joint function. Two of the most common procedures are Total Knee Arthroplasty (TKA) and Total Hip Arthroplasty (THA). THA surgeries are used as a final effort to treat symptomatic pain and loss of function of the hip joint in cases of late-stage osteoarthritis, rheumatoid arthritis, osteoporosis, osteonecrosis, hip dysplasia, and other diseases of the hip [1]–[3]. Additionally, THA surgeries are common and there is no expected decline in the rate of THR procedures with an increase of 71% expected over the next 10 to 20 years [4].

Learmonth et al. described the recent developments in THA as being aimed at reducing the rate of failure while accommodating the high-activity and increasing the implant longevity to accommodate the modern patient [5]. Many factors influence the success and failures of THR, including the complexity of surgical technique, the volume of surgeries performed by surgeons, the standard of care post-operation, anthropometric variability, and implant geometry [6]–[12]. Jones et al. found that while some patient-reported success metrics are relative to surgical factors or complications, some patients had no identifiable link, suggesting the problem is multifactorial [2]. To improve patient outcomes following THA procedures, it is crucial to minimize the effects of and occurrence of the controllable factors that reduce the success rate of the procedure.

In the event of primary surgery complications, the surgeon can elect for the patient to undertake revision surgery, in which parts or the entirety of implanted hardware are removed and replaced. The leading causes of revision surgeries for THA at any point along the life of the implant are predominantly: aseptic loosening, instability or dislocations, infections, wear, or periprosthetic fractures [6]–[8], [13]. Both periprosthetic fracture and aseptic loosening have been observed to be a common early-to-mid-term cause for revision surgeries in THA [14]–[17]. Aseptic loosening is defined as the failure of the fixation of any portion of the implant in the absence of infection. Periprosthetic fracture is defined as the fracture of the bone surrounding implanted components. Periprosthetic fracture has been hypothesized to be a common source of revision within 2 years of primary surgery occurring in roughly 10.4% of 535 patients [14]. Bozic et al also found that periprosthetic fracture occurred in 6.2% of 51,345 patients [15]. Mäkelä et al. also found that in 3,868 patients aseptic loosening accounted for roughly 5.9% of total reasons for revision [6]. The incidence of aseptic loosening and periprosthetic fractures is thought to be impacted by many factors, one factor that may be critical is the size and orientation of the femoral stem chosen for implantation [18].

In THA, the use of pre-operative software and templating has improved both femoral stem sizing and alignment [19]. Specific software such as TraumaCad® and mediCAD® allow surgeons to establish virtual component placement on a subject-specific basis while allowing for observation of changes in THA component alignment and sizing on relevant clinical measurements (e.g., Limb Length Discrepancy, center of rotation, etc.). Pre-operative templating commonly uses 2D and 3D patient imaging data to establish the

ideal implant size and orientation [19]. Pre-operative templating can evaluate implant fit and orientation while considering unique patient-specific challenges with anatomic and implant geometry. Despite the benefit of pre-operative templating, there are still shortcomings that exist with the accuracy of different software. These inaccuracies are thought to be caused by factors such as patient radiation exposure, lack of meaningful anatomic information based on imaging technique, and patient anthropometric variability (such as BMI) [19]–[22].

The goal of evaluating implant fit and orientation in a templating environment is to identify the optimal location of the implant in a non-surgical environment for the patient. The chosen implant location often restores both the anatomic leg length compared to the contralateral hip and the center of rotation of the operated hip [23]–[25]. The chosen location of the femoral stem often combines the restoration of the femoral head center with good implant seating. Though in cases where neither are fully restored, there is an increased risk of periprosthetic fracture, aseptic loosening, and implant subsidence while also leading to a decrease in initial fixation [16], [17], [26], [27]. Furthermore, the restoration of the femoral head has been shown to drive abductor muscle moment arms, impairment of gait, risk of dislocation, and in some extreme cases is thought to increase the incidence of trochanteric pain and pelvic-trochanteric impingement [12], [25], [28]–[33]. A lack of restoration of the two goals can lead to poor implant seating, which can decrease initial fixation potentially leading to increased rates of aseptic loosening and periprosthetic fracture [16], [17], [26], [27]. The lack of consideration of the patient-

specific proximal femoral morphology specifically, intercortical bony geometry, can also lead to improper restoration, further causing these problems [11], [12], [21], [22].

Furthermore, the templated size of the femoral stem can differ from the implanted size of the femoral stem depending on the imaging technique used and surgeon preferences for femoral stem size and orientation [19], [21], [34]. In a larger retrospective review Mirghaderi et al observed that using mediCAD® preoperative templating, the exact size of surgically implanted femoral stem in patients occurred 27.2% of the time, within one and two sizes of implanted stem 61.0% and 78.6% of 346 femoral stems respectively. In a clinical environment these differences Surgeons can induce changes in the femoral offset by increasing implant size to achieve a better fill of the implant or to improve diaphyseal/metaphyseal contact [35]. Though with an increase in size, there is an inherent reduction in the amount of variability in implant positioning, by achieving better fill the surgeon could compromise the accuracy of femoral head restoration [12], [18], [35]–[37]. Additionally, increasing the size of the implant can influence the 3D alignment of the stem in the intermedullary canal, which is often underreported, producing undesired effects on the version of the implant or the anterior-posterior offset which can induce the complications detailed above.

The impact of implant size on the restoration of the head center has been observed in both 2D and 3D for various implant geometries. However, the relationship between femoral head restoration, patient intermedullary anatomy, and implant geometry has not been explored on a large scale, which is the motivation for this thesis. Presently, there is also a lack of a reliable and efficient workflow to use external manually segmented femur

anatomy to extract the intercortical anatomy and to perform virtual implantations of a variety of femoral stems. The purpose of this thesis was to create an algorithm that is capable of extracting the inner-cortical wall of the femur in the diaphyseal and metaphyseal regions that enable the consideration of implant and femoral geometry in restoring clinically relevant metrics for positioning of the femoral component in THA. The purpose of this thesis is to further explore the relationship between implant geometry and femoral geometry concerning restoring clinically relevant metrics and restoration of the femoral head center. Additionally, to create an algorithm that is capable of extracting the inner-cortical wall of the femur in the diaphyseal and metaphyseal region

1.2 Objectives

The objectives of this thesis are as follows:

1. To develop an algorithm that automatically extracts the femoral intercortical geometry from CT scans.
2. To develop an implant sizing algorithm that predicts the range of femoral stem sizes and alignments that fit within the femoral intercortical bony geometry.
3. Verify the femoral stem alignment predictions through comparisons with implanted cadaveric femoral bones.
4. Demonstrate the ability of a modern total hip replacement system to restore the femoral head center across a population of femoral bones.
5. Identify the anatomic characteristics of the femoral bone (e.g., Dorr classification, femoral neck version, femoral bowing, etc.) that contribute to the ability of the modern femoral stem geometries to accurately restore the femoral head center.

1.3 Thesis Overview

Each chapter of this thesis is written to provide original content to contribute to the field of orthopedic biomechanics with each chapter falling into the chronological order in which the research was performed. Chapter two provides a review of previous work done in the field of total hip replacements to provide background information on the current state of literature. Chapter three describes a study that explored the impact of intercortical variability and implant geometry on restoring the femoral head center in a large cadaveric cohort. Chapter four provides a conclusion with final remarks and recommendations for future work. This is followed by appendices and references to all applicable work and extra figures for each study.

CHAPTER TWO: Review of Pertinent Literature

2.1 Review of Bony Anatomy

The purpose of this section is to provide the reader with a comprehensive review of the pelvic and femoral anatomy as it pertains to this project.

2.1.1 Review of General Bony Anatomy for the Lower Limbs

The organic and mineral composition of bone tissue largely determines the mechanical properties of itself. A distinct separation between the different types of bone tissue that are found in the human body. While bone can be primarily categorized into two categories: denser cortical (non-porous) bone and less dense cancellous (porous) bone. Both the density and porosity of bone tissues have been observed to be on a continuous scale from 0-100% rather than being simply porous or non-porous [38]. The molecular structure of bone can further be categorized as one of two types mineralized and non-mineralized.

About 65% of bone by weight is mineral, 20–25% is mostly Type 1 collagen, roughly 10% is water, and 1-2% is non-collagenous proteins [38]. Type 1 collagen, which is a triple helix that aggregates into an arrangement of parallel spaced molecules, at the microstructural level, forms a crosslinked matrix with nanocrystalline and carbonated apatite to form bone tissue [38]. Cortical bone is denser and has a higher mineralized content, while cancellous bone is less dense and has a lower mineralized content [38].

Similarly, the porosity of each type of bone follows a similar trend with the bone mineral density, with cortical being less porous (5-10% porosity) and cancellous having higher porosity (75-95% porosity) [38]. The primary function of cortical bone is to transfer load

through the body's skeletal structure. While the cancellous function is twofold; one purpose is to provide a matrix for bone marrow and blood vessels, and the second is to provide structural support in addition to the cortical bone by orienting loading along the direction of its fibers (Figure 1). The differences in the microstructure of the calcified bone matrix in cancellous and in cortical bone are responsible for measured differences in the material properties of bone tissue. The material properties of the bone (e.g., strength and stiffness) are anisotropic and strongly correlated to bone mineral density. The cortical bone is oriented to have its collagen matrix oriented to be parallel to the long axis of the bone which lends to higher values of matrix strength and stress [38]. While cancellous bone due to the porosity of the matrix has higher stress and strength when in compression [38].

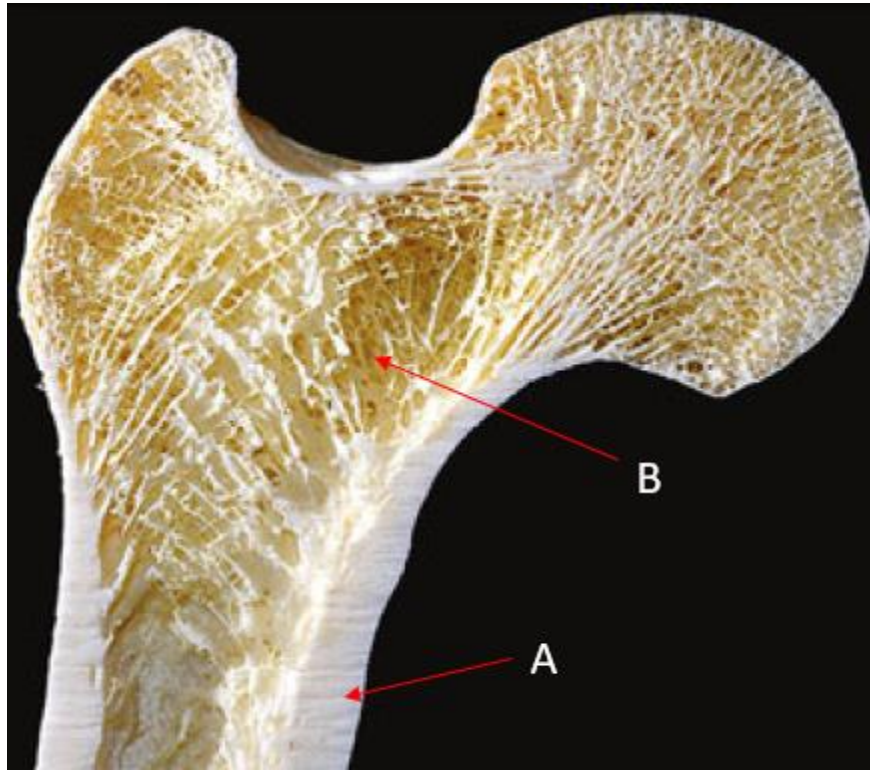
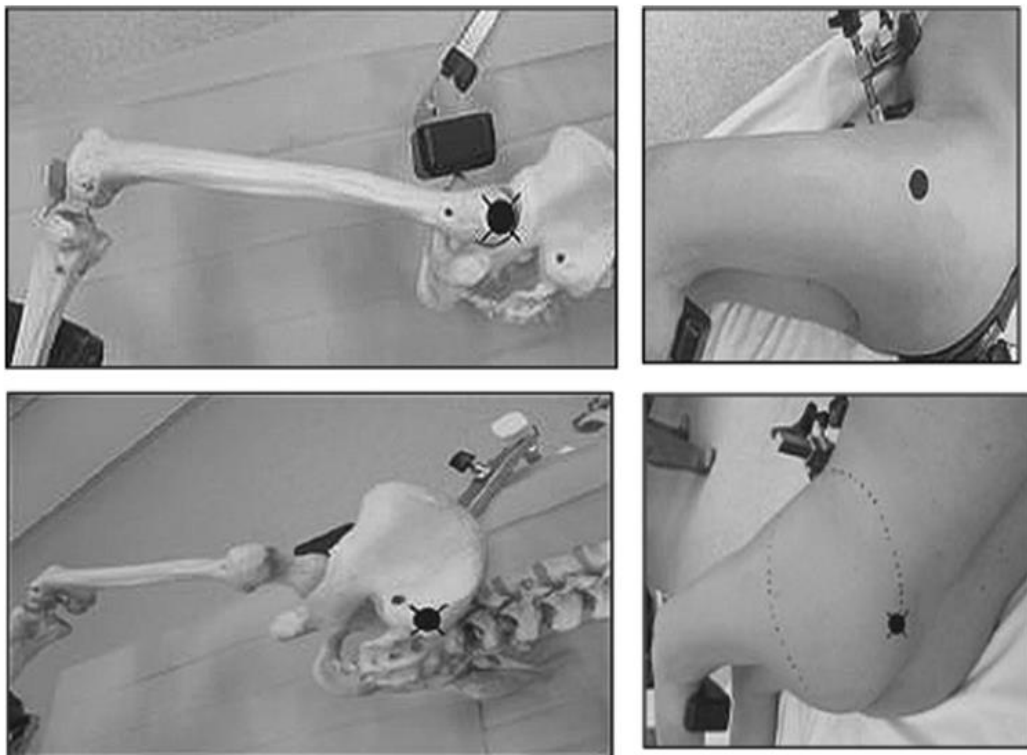


Figure 1: Coronal view of the proximal end of a human femur. A-Cortical Bone, B-Cancellous Bone. (image from: Martin et al 2015)

The hip joint is a ball and socket joint with three degrees of rotational freedom compromised of the femur and pelvic bones. The pelvic bones consist of the ilium, ischium, and pubis for the right and left lower extremities. During a THA surgical exposure of the pelvis or femora is highly likely due to the soft tissue surrounding the hip joint. With the variability present within each surgical approach, there is a need for reliable identification of bony landmarks to aid surgeons in standardizing each approach [39].

These bony landmarks are commonly located at the insertion point of the muscles, prominent portions of bones, and portions of bone that are conveniently located for surgeons and are identified before the start of any operation using various medical

imaging techniques. Focusing specifically on THA, the most common landmarks of the hip joint that are used and identified are the anterior superior iliac spine (ASIS), posterior superior iliac spine (PSIS), greater trochanter, pubic tubercle, and pubic symphysis [40]. There are also other relevant locations on the femur that are commonly used by surgeons and researchers. Some common femoral surgical landmarks are the greater trochanter, lesser trochanter, femoral head, and the calcar femorale.



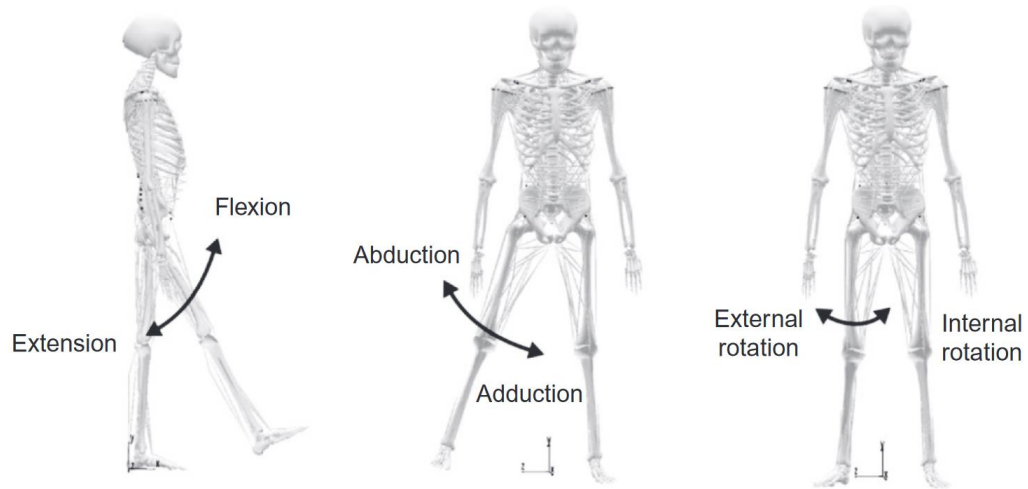


Figure 2: Upper) Examples of surgical landmarks on the pelvis (image from: Ait et al 2020). Lower) Degrees of freedom of the hip joint (image from: Zhang et al 2009)

The hip joint muscles allow for tri-planar rotational movement of the pelvis and trunk relative to the femur [41]. The rotations can be categorized as Flexion-Extension (FE), Internal-External (IE), and Abduction-Adduction (ADAB). Furthermore, each of the muscles in the hip joint can be categorized based on its orientation and line of action relative to the hip joint's primary axis of rotation listed above (Figure 2 Lower) [41].

On the femur, bony surgical landmark sites often occur at the insertion sites of muscles or tendons as these sites are usually points of prominence on the bone. All except the calcar femorale act as insertion sites for musculature and ligaments and act as bony prominence points that surgeons can identify. Looking specifically at the greater trochanter, the muscles that attach on or around the greater trochanter (gluteus medius, gluteus minimus, piriformis, obturator externus, and obturator internus) provide stability and accommodate unconstrained motion in all three rotational degrees of freedom [41]–[43]. While the lesser trochanter muscle attachments (the psoas major and iliacus muscles) accommodate primarily pelvic flexion and extension relative to the trunk [44].

The femoral head is a sphere-like structure found on the superomedial portion of the femur. The femoral head makes up the “ball” portion of the socket joint that is used to classify the hip. There is also a depression superomedially on the femoral head that is the attachment site of the teres ligament which originates in the acetabular notch. Figure 3 lower shows some of the insertion sites of muscles on the proximal femur and labeled boney landmarks.

TABLE 2		MUSCLES OF THE HIP, ORGANIZED ACCORDING TO PRIMARY OR SECONDARY ACTIONS*
Muscles	Primary	Secondary
Flexors	<ul style="list-style-type: none"> • Iliopsoas • Sartorius • Tensor fasciae latae • Rectus femoris • Adductor longus • Pectineus 	<ul style="list-style-type: none"> • Adductor brevis • Gracilis • Gluteus minimus (anterior fibers)
Extensors	<ul style="list-style-type: none"> • Gluteus maximus • Adductor magnus (posterior head) • Biceps femoris (long head) • Semitendinosus • Semimembranosus 	<ul style="list-style-type: none"> • Gluteus medius (middle and posterior fibers) • Adductor magnus (anterior head)
External rotators	<ul style="list-style-type: none"> • Gluteus maximus • Piriformis • Obturator internus • Gemellus superior • Gemellus inferior • Quadratus femoris 	<ul style="list-style-type: none"> • Gluteus medius (posterior fibers) • Gluteus minimus (posterior fibers) • Obturator externus • Sartorius • Biceps femoris (long head)
Internal rotators	Not applicable	<ul style="list-style-type: none"> • Gluteus minimus (anterior fibers) • Gluteus medius (anterior fibers) • Tensor fasciae latae • Adductor longus • Adductor brevis • Pectineus • Adductor magnus (posterior head)
Adductors	<ul style="list-style-type: none"> • Pectineus • Adductor longus • Gracilis • Adductor brevis • Adductor magnus (anterior and posterior heads) 	<ul style="list-style-type: none"> • Biceps femoris (long head) • Gluteus maximus (posterior fibers) • Quadratus femoris • Obturator externus
Abductors	<ul style="list-style-type: none"> • Gluteus medius (all fibers) • Gluteus minimus (all fibers) • Tensor fasciae latae 	<ul style="list-style-type: none"> • Piriformis • Sartorius • Rectus femoris

* Each action assumes a muscle is fully activated from the anatomic position. Several of these muscles may have a different action when they are activated outside of this reference position.

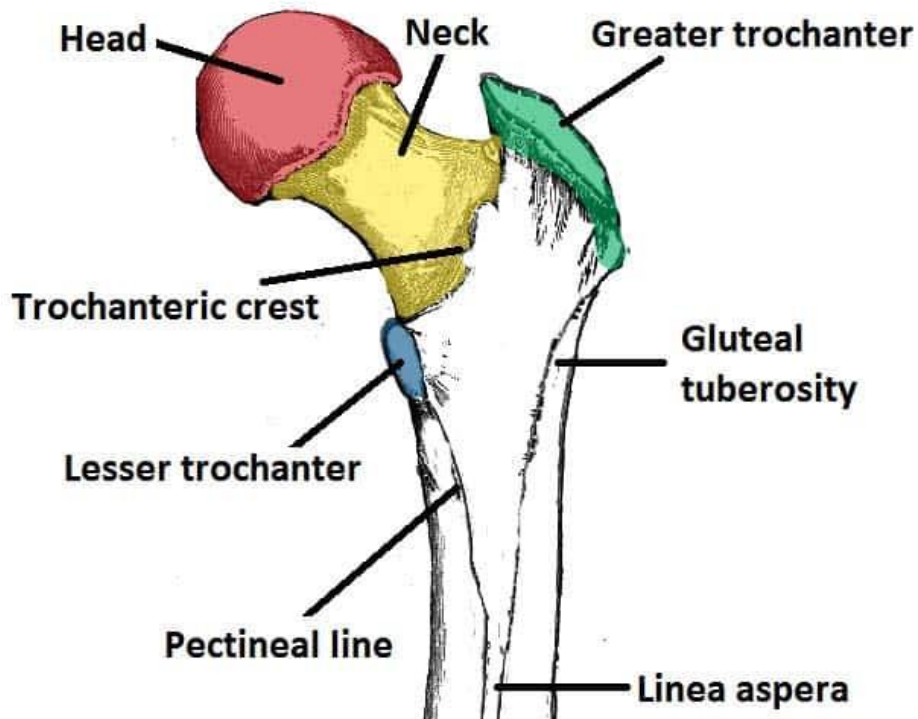


Figure 3:Upper) Table of muscle insertions and primary actions from Neuman et al. 2010. Lower) Example of common surgical landmark sites on the proximal femur (image from: <https://teachmeanatomy.info/lower-limb/bones/femur/>)

The calcar femorale is one region of the femur bone not located on the endosteum; it is found within the medullary cavity of the femur (Figure 4). The calcar femorale is a vertically oriented dense bone region that acts as a natural separation between the medullary cavity and the lesser trochanter. Removal of the posterior cortical bone partially removes the calcar itself and creates the illusion of a spur projecting into the femoral metaphysis [45]. The calcar femorale has been referred to as the true neck of the femur and acts to distribute the weight of the body on the lower limbs. The calcar region of the bone is commonly used as a resection plane for the femoral stem in a THA procedure. Zhang et al. found that the calcar region redistributes the compressive weight

of the body from the posterior and medial femur and into the anterior and lateral femur [46].



Figure 4: Appearance of the calcar femorale on posterior dissection. F-calcar femorale: LT.lesser trochanter. (image from: Garden et al. 1961)

2.1.2 Review of Femoral Intercortical Anatomy

As mentioned in the previous section, the mechanical properties of different bones are dictated by the mineral composition and porosity of each bone. The separation of cortical and cancellous bone exists in all bones in the human body; however, based on the function of bone, the composition of cortical and cancellous bone can vary [38]. This variation is partially because of the function of the skeletal system, which is constantly optimized to maximize the strength of bone while minimizing bone weight under stress and activities of daily living and can cause a constant change in the composition of cortical and cancellous bone found in the body.

For long bones such as the humerus, tibia, and femur, there are three distinct regions where the composition of bone is noticeably different. This is predominantly

caused by the loading of long bones, there is a need for a denser region of the bone to allow for axial loading of the bone without buckling. These regions are referred to as the metaphysis, diaphysis, and epiphysis. The diaphysis is the hollow tube-like shaft between the proximal and distal ends of the bone, the epiphysis is located near the proximal and distal ends and the metaphysis acts as a transition between the diaphysis and epiphysis. In long bones, the long hollow region is also referred to as the intramedullary canal or cavity as there is medullary bone marrow found in this region. Figure 5 illustrates the differences between each region.

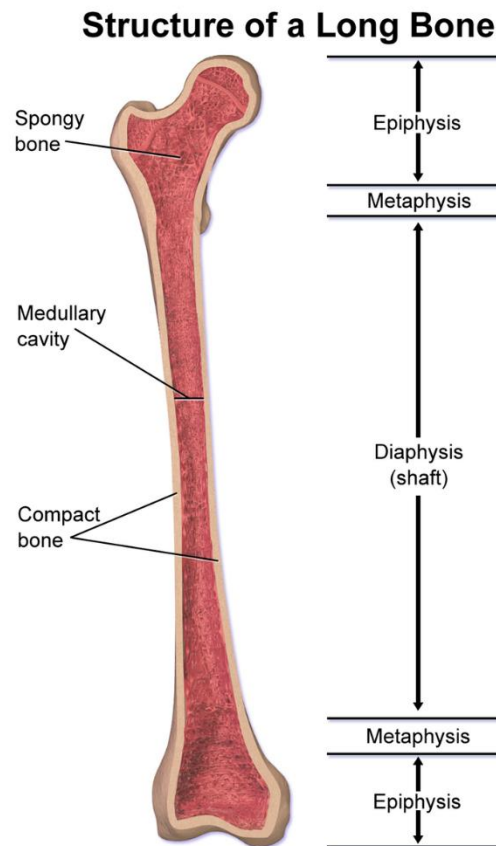


Figure 5: Structure of a long bone (image from: <https://en.wikipedia.org/wiki/Diaphysis>)

2.2 Total Hip Arthroplasty General Information and Femoral Stem Design

The purpose of this section is to introduce the Total Hip Arthroplasty (THA) procedure and to familiarize the reader with the overall procedure. This includes overall goals and rationale for receiving this procedure as well as information detailing the rationale for specific design decisions of components. Additionally, the purpose of this section is to familiarize the reader with the femoral stem specifically.

2.2.1 Total Hip Arthroplasty Overview

Total Hip Arthroplasty (THA) is one of the most common arthroplasty procedures; with more than a million occurring annually [47]. Furthermore, there is also no predicted decline in the rate of expected THA procedures with a projected increase in THA procedures over the next 10 to 20 years by 71% to 635,000 [4]. THA procedures have a low rate of complications and are an effective treatment for most individuals who suffer from pain and loss of hip joint function due to late-stage osteoarthritis [1], [2]. According to the AAOS, a THA procedure can also occur in cases of rheumatoid arthritis, osteoporosis, osteonecrosis, childhood hip disorders, and hip dysplasia to reduce symptomatic pain [3]. THA, like other total joint replacement surgeries, is an elective surgery that can be performed if conservative treatment fails to alleviate joint pain [2]. Generally, THA consists of four components: femoral stem, femoral head, acetabular liner, and acetabular cup. In a THA, the surgeon removes diseased bone and cartilage from the hip joint and implants the femoral and acetabular components into the prepared bone (Figure 6).

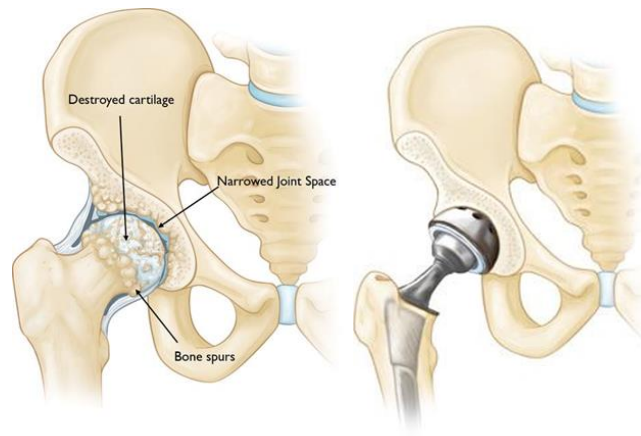


Figure 6: Visualization of THA (image from: <https://orthoinfo.aaos.org/en/treatment/total-joint-replacement/>)

To ensure good clinical outcomes, implant survivorship, and mitigate surgical complications, there has been an increase in the available geometry and modularity of implanted components to accommodate patient-specific needs. THA components are generally separated into two categories based on the initial fixation of implanted components. These categories are referred to as cemented or cementless implants. Cemented implantation uses polymethylmethacrylate to fill the prepared acetabular and/or femoral sites to fix the femoral stem and/or acetabular components [48]. Cementless implants utilize a press-fit, hydroxyapatite-coated component, to promote bone on-growth to be the primary method of fixation in the prepared acetabular and femoral sites [49] (Figure 7). However, components can be entirely cemented, cementless, or a hybrid combination of cemented/cementless hardware. Regardless of the fixation technique, these implants aim to ensure lasting fixation while avoiding surgical and post-operation complications that could lead to revision surgery.



Figure 7: Left) X-ray of the hip in lateral view showing a partial cemented hip replacement Right) X-ray showing an uncemented total hip replacement. (Images from: <https://www.cortho.org/hip/uncemented-vs-cemented-hip-replacement/>)

Fixation techniques have been shown to vary internationally; this has led to debate around specific patient demographics that could benefit from a specific fixation technique over others [7], [13]. Due to the variability in patient anatomy, fixation technique, and surgical technique used, the causes for revision surgery are multifactorial. In literature, some of the reasons for revision have been shown to vary with fixation technique, surgical procedure, implant type, implant geometry, patient proximal bone morphology, and patient bone quality [6]–[10]. Some of the most common reasons for revision include aseptic loosening of one or more components, osteolysis, infection, dislocation, malposition, instability, fracture of a prosthesis, periprosthetic bone fracture, and excessive wear [6]–[8], [13].

Conflicting information exists in the literature. One study found that for patients under 55, there was no difference in revision rates when age or gender was compared to the fixation technique [6]. However, at the 10-year mark of the same study, there was a much higher risk of revision for cementless components than cemented implants regardless of demographics [6]. It is unclear if the differences in revision rate reflect the range of implants used for these procedures rather than the mode of fixation and patient variability [6], [8]. Additionally, the discrepancy in activity level and pain tolerance in younger and older patients was also thought to influence the survivorship of implants and the need for revision surgery [6], [50]. This rationale could explain the higher revision rate found in younger populations when compared to older populations [6].

Alignment of the femoral stem is related to several of the leading reasons for revision and is crucial for long-term THA component survivorship. Misalignment of the femoral stem has been identified as a primary early- to -medium-term mechanical failure mode of a THA [16]. Improper alignment of the femoral stems could imply poor prosthesis seating and decreased likelihood of initial fixation [27]. Femoral stems are also at their greatest risk of subsidence or early fracture following primary placement and impaction into the bone but before secondary osteointegration, [26]. The two most prominent mechanical failure modes caused by misalignment are aseptic loosening of the femoral stem and periprosthetic fracture of the proximal femur. The alignment of the femoral stem is also thought to impact the severity of stress shielding which can lead to periprosthetic fracture [16], [17]. Stress shielding is a stiffness mismatch between implanted components and native bone. This discrepancy can cause the mechanical load

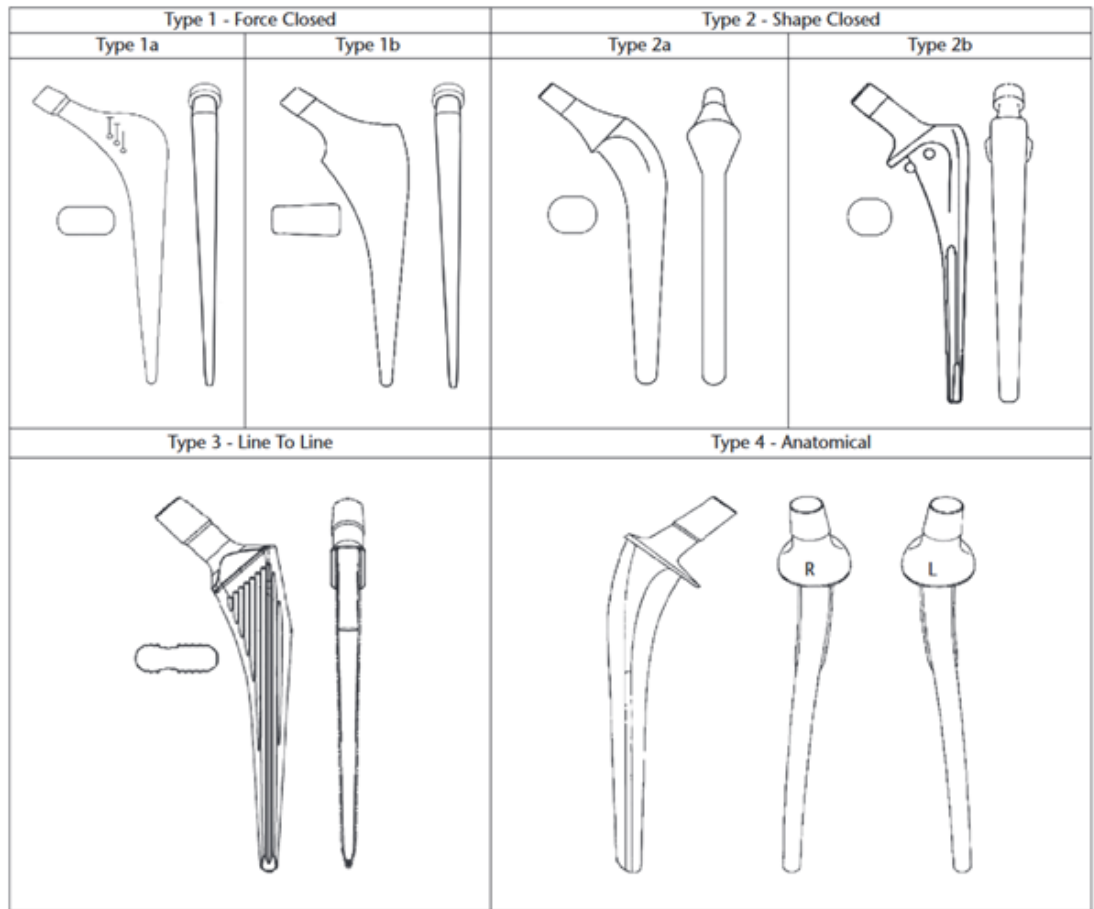
to transfer away from the adjacent bone leading to cortical and cancellous osteolysis [51]. Additionally, the success of a THA is strongly influenced by the initial stability and stress shielding effect [52], which are both affected by femoral stem alignment.

Challenges in reproducing native alignment with an implanted femoral stem are multifactorial. The femoral offset is often used to describe the variation in femoral stem placement relative to the native femoral head center often being described by a series of anatomically meaningful measurements in three dimensions. Surgical approach and stem geometry can be potential factors that impact stem alignment and femoral head restoration [27]. The same study found that stems were more often placed in varus when the direct anterior approach was used, while the stem geometry was held constant between subjects [27]. Thigh pain is often a symptom associated with loosening and stress shielding, however, there were no indications of thigh pain while using cementless femoral stems observed any time after five years [17], [27], [53]. Furthermore, femoral stem misalignment was found to alter patient gait and joint kinematics [30], [54], [55]. It was found that decreasing the femoral offset generated a slower swing speed of the knee and reduced knee range of motion throughout gait [55]. This highlights the influence of femoral stem alignment and the impact of femoral head center restoration in producing good clinical outcomes.

2.2.2 Femoral Stem Design

Many things separate different models of femoral stems. The differences are mostly geometry-based, but fixation techniques can also impact femoral stem design. As mentioned before cemented femoral stem geometry is designed to provide a stable

interface between the implant-cement and cement-bone interface while cementless femoral stem geometry prioritizes osseous bone contact due to the press-fit contact of the implant and bone. [5], [48], [49]. Most cemented femoral stems are made of cobalt-chromium or stainless-steel alloys which are corrosion-resistant [48]. Cementless femoral stems are mostly made of similar cobalt-chromium-molybdenum alloys and/or titanium aluminum-vanadium alloys [49]. For cementless implants, a hydroxyapatite coating is applied to a porous coating located on the shaft where the desired osteointegration occurs [5], [49]. The porous coating can either be plasma sprayed onto the entire implant or a porous portion of the stem; typically, the porous regions are created by either grit blasting or plasma spraying the regions of interest [49]. Figure 8 shows one classification method for both cemented and cementless femoral stem designs.



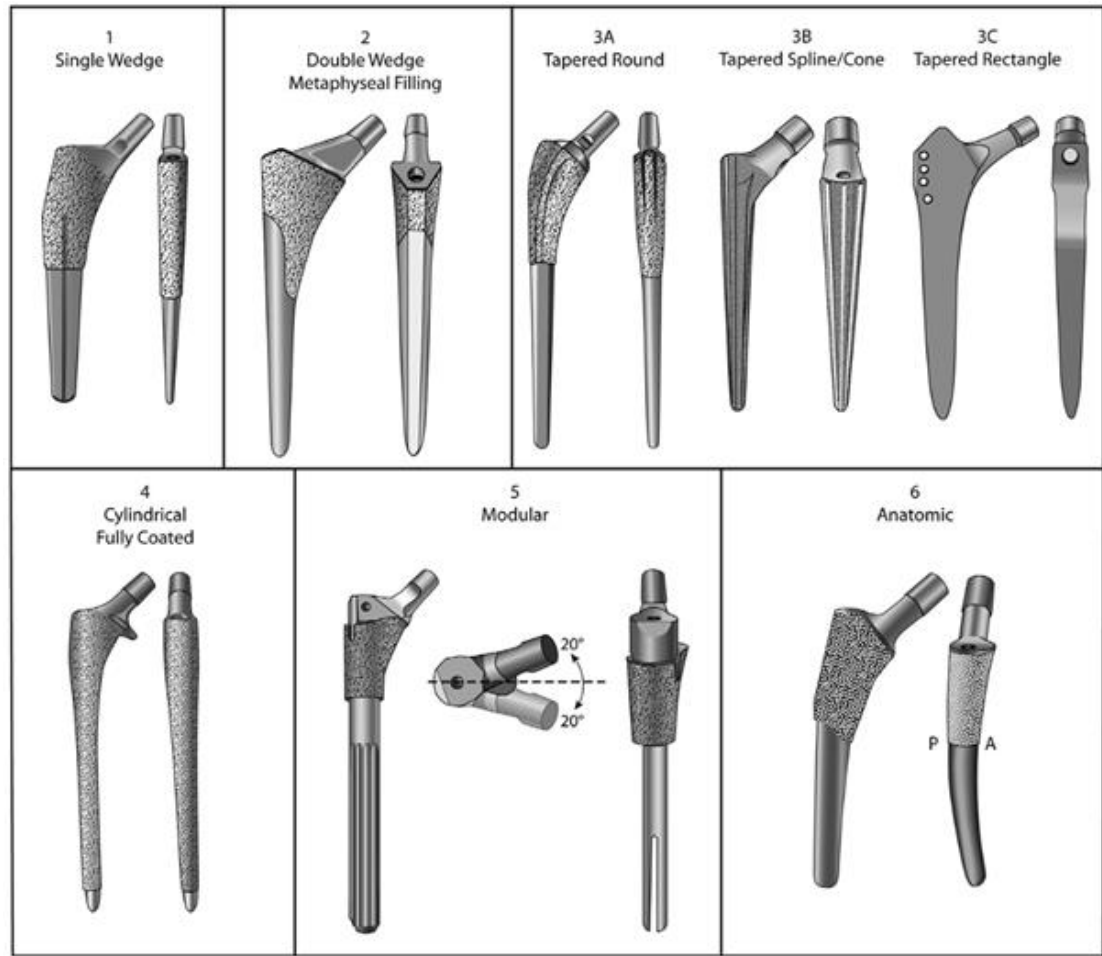


Figure 8: Upper) Examples of cemented femoral stem design (image from Khanuja et al 2019). Lower) Classifications of cementless femoral stems (image from Cassar-Gheiti et al 2020)

Looking closer at cementless implants, various femoral stem geometry-based classification systems exist. Following the classification system used by Khanuja et al., the design of cementless stems falls into one of six general categories. However, the evolution of modern femoral stem design can fall under multiple categories [49]. Referring to Figure 8 lower, types 1-3 of cementless femoral stem are tapered and designed to obtain more proximal fixation while being by the diaphyseal canal [49], [56].

Type 1 implants are designed to engage the intramedullary canal in only the medial-lateral plane with initial stability occurring because of metaphyseal engagement of the

stem or three-point contact [49]. Type 2 or metaphyseal-filling, stems are designed to have metaphyseal contact in two directions: medial-lateral and anterior-posterior. Type 3 implants have a long anterior-posterior and medial-lateral taper that is consistent over its length; unlike Types 1 and 2, there is no abrupt change in geometry when the taper ends. The fixation location of Type 3 implants is in the metaphyseal-diaphyseal junction rather than the metaphysis [49]. Modern femoral stem geometry is a mix of the classification systems used in literature and the classification systems that have been developed to serve as a way of identifying general implant shapes.

Modifications can be made to the Type 1-3 geometries for specific bone morphologies and anthropometric demographics. One modification is the use of a collar, designed to improve initial stability, reduce subsidence, and lower the calcar fracture risk [26], [57]. Another modification is a shorter stem length, which was believed to aid proximal femur loading and used in minimally invasive surgical approaches (such as the direct anterior approach) to preserve implantable bone [5], [57]. In a 34-study meta-analysis, Panichkul et al. [57]. found that the general occurrence rate of complications (e.g. loosening) was lower in collared femoral stems compared to collarless stems which was further consistent with larger retrospective studies. In the same study, the long femoral stems were found to have lower risks of complications when compared to short femoral stems [57]. Though, there was no significant difference in the rate of revision between both collared and collarless stems and short and long femoral stems [26], [57]. When comparing stem geometry there were also significant findings to revision, specifically femoral component loosening, which could further validate the claims by

Ruben et al. [26]. While there are mixed reasons in the literature surrounding the causes of aseptic loosening and periprosthetic fracture, fixation techniques alone are likely not the only cause of these early-to-mid-term failures of cementless stem geometries [6], [8]. Variability in intercortical bone morphology and stem geometry were thought to potentially promote poor implant seating which can cause aseptic loosening due to lack of osseointegration [27], [49], [56], [57]. Bone morphology, specifically the canal-flare index, was shown to influence the incidence of calcar region fractures [53]. The canal-flare index is defined as the ratio of the diameters of the intramedullary canal at the isthmus and 20mm proximal to the lesser trochanter in the frontal plane. Furthermore, it was found that the lack of initial stability in tapered wedge stems is also thought to impact the incidence of subsidence and early- to -mid-term axial migration of the implant in the femoral canal, which could be a predictor for aseptic loosening [58]. Stem alignment was also found to not impact the occurrence of stem subsidence [53]. Clinical symptoms such as thigh pain were not observed in the studies that calculated stem alignment, furthermore, there was no correlation between stem alignment and symptomatic thigh pain [17], [53]. However, symptomatic thigh pain is an example of one patient reported metric, and the interplay between stem geometry and implant alignment could be present in other clinically observed metrics of success (e.g. patient gait, limping, etc.). Further indicating that the relationship between femoral stem geometry, alignment and patient outcome and complication rates using one patient reported metric of success may not capture this relationship.

Expanding more on the relationship between bone morphology and cementless femoral stem survivorship, the classification system described by Dorr et al. [59] has been used to classify the proximal intercortical morphology into one of three types to aid the fixation decisions made by surgeons but has been adapted to be used as a metric of comparison between studies in recent times. Dorr et al. [59] defined two critical metrics related to bone morphology: the cortical index and the calcar-to-isthmus ratio. The cortical index was defined as the difference between the femoral diaphyseal diameter and intramedullary canal diameter over the femoral diaphyseal diameter at a point 10 cm distal to the mid-lesser trochanter as a reflection of the cortical thickness [59], [60]. From the cortical index three distinct proximal morphologies of bone were found ranging from the largest cortical thickness to the smallest: Type A, Type B, and Type C (Figure 9).

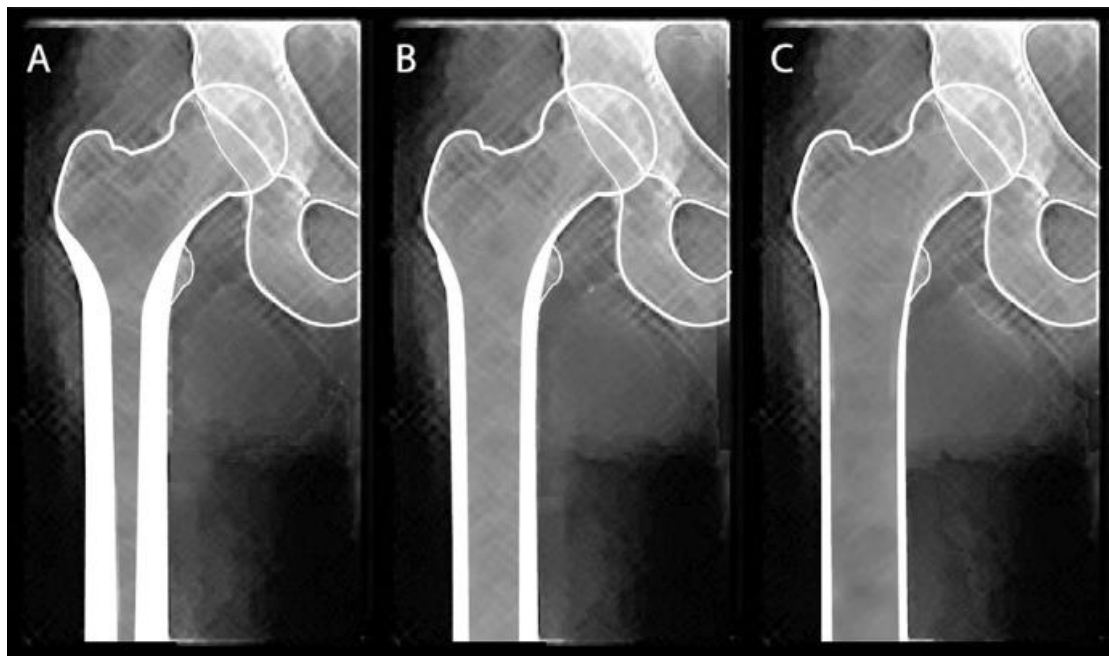


Figure 9: Examples of the three Dorr classifications (image from: Khanuja et al 2019)

The calcar-to-isthmus ratio is defined as the fraction of the intramedullary canal isthmus over the calcar isthmus dimension, with both calcar and intramedullary isthmus being measured 3cm and 10cm respectively distal from the midpoint of the lesser trochanter [59]. When looking specifically at tapered wedge femoral stems for each type of Dorr classification, Type-C femurs showed the highest survivorship when comparing geometries [49]. Additionally, Type-A bones were thought to benefit from wedge stems similar to the Corail [26]. Additionally, Watanabe et al. observed that excessive femoral stem anteversion increases is a risk factor for the different types of Dorr classifications with cementless stems, however, there was no further examination of each Dorr type due to biasing of Type B Dorr femurs [61]. This could explain why Type-C bones saw the highest success as this morphology allows for the largest amount of flexibility in femoral stem placement allowing surgeons to achieve the desired position with primary fixation.

Beyond the Dorr classifications for each bone, other femoral morphological changes have driven implant design. The variance of femoral bowing and femoral neck version has driven the use of anatomic stems, which can utilize both cemented and cementless methods of fixation. When high degrees of femoral bowing or femoral neck anteversion exist, traditional straight, rectangular, tapered implants may not fully restore the head center, so to solve this problem anatomic stems were created to produce a better match between native version and bowing of the femur [48], [49]. In particular, the design of cementless anatomic stems consists of a wider proximal portion of the stem in the lateral and posterior directions, this is to better match the flare that can exist within the metaphysis (Figure 10). Fixation of cementless anatomic stems is primarily achieved

through the metaphyseal fill and the distal curve and broaching can be less exact due to the curve of the implant matching the intermedullary geometry [49]. Native femoral anteversion can also be achieved through modular implants which can accommodate a specific degree of intermedullary fill while allowing for unique cases of femoral anteversion or retroversion to be solved independent of the stem behavior. Though in practice success of modular and anatomic cementless stems is design-dependent, making their use scarce compared to the tapered stem [49].

Test Bone and Stem Types

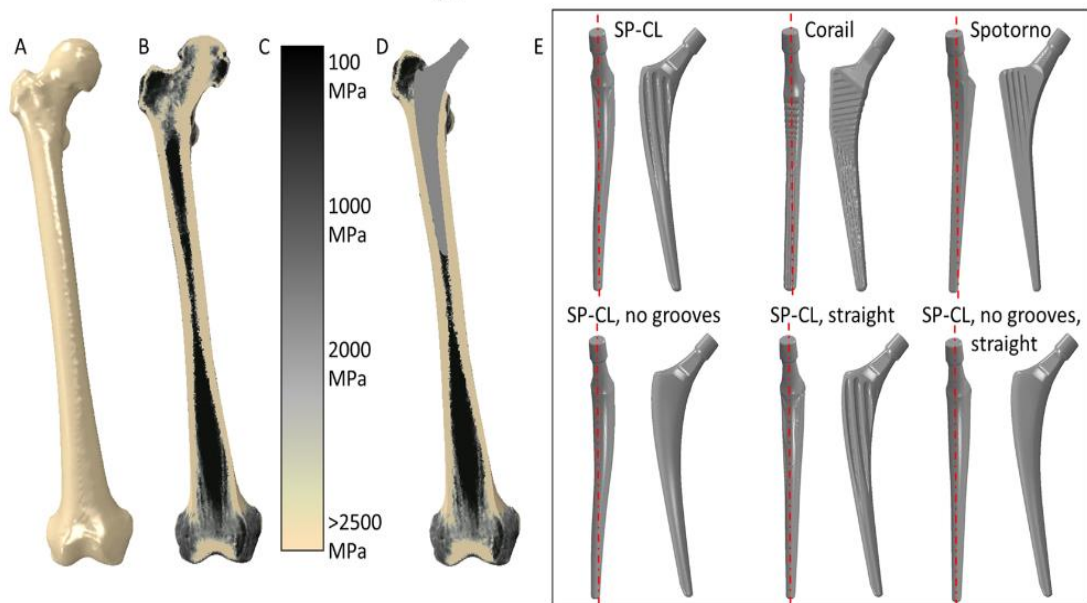


Figure 10: Examples of anatomic stems in comparison to tapered wedge femoral stems (image from Heyland et al. 2019)

Another source of proximal femoral deformity is the differences that can exist in the femoral neck-shaft angle. Individuals that have a femoral neck shaft angle that is less than 120° are normally considered coxa vara and individuals with a neck shaft angle greater than 135° are considered coxa valga (Figure 11). If a standard 130° or 135°

femoral neck offset stem is implanted in these individuals the femoral offset may not fully be restored or may produce a leg length discrepancy. Both of these are undesirable and could impact the functional outcomes of the patient or drive failure of the bone-implant interface. If there is a produced discrepancy in the global femoral offset, components (acetabular liners and femoral heads) have offset built into each liner and femoral head that can be used in combination with the implanted femoral stem and acetabular cup to restore the native femoral offset. Another method of solution involves implant designs carrying modular offsets to accommodate this neck-shaft discrepancy [21]. Which modern implants have started to incorporate a variety of neck-shaft angles to increase effective femoral head restoration (e.g., Corail).

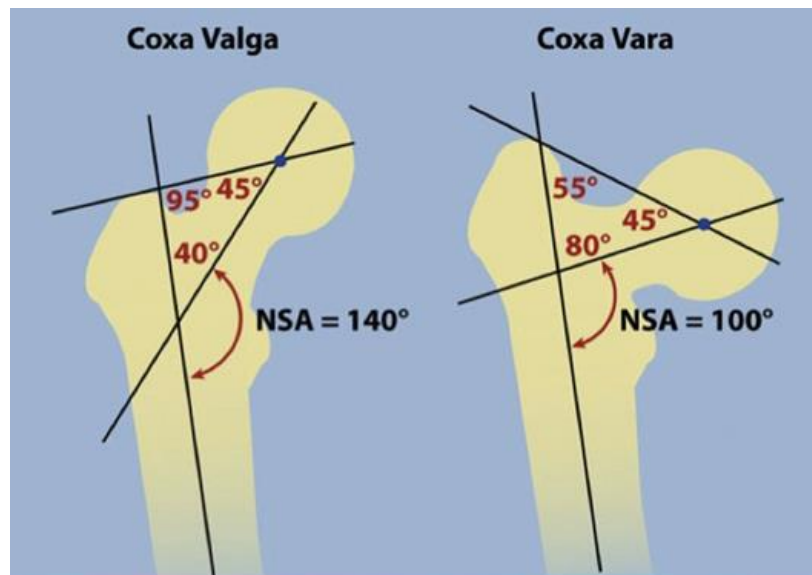


Figure 11: Examples of coxa vara and coxa valga femoral neck-shaft angle deformities. (Image from: <https://paleyinstitute.org/centers-of-excellence/joint-health/hip-preservation/#/>)

2.3 General Preoperative Software Information

This section aims to introduce the concepts of templating to the reader. By the end of this section, the reader should understand the clinical use of templating, its purpose, and

its capabilities and limitations concerning femoral stem placement. The reader should also understand what clinically is done to evaluate a THA and what can potentially influence the clinical success of a femoral stem.

2.3.1 Overview of Pre-operative Software

In orthopedics, preoperative software is generally defined as any software that a surgeon can use before an operation to aid their understanding of patient anatomic geometry and implant geometry. This software also supports the surgeon by allowing changing of medical devices placement, orientation, and size, as well as the surgical approach in a virtual environment. Currently, this process is done using some combination of digitized patient imaging data consisting of 2D radiographic imaging, 3D Computed Tomography (CT), weight-bearing radiographic imaging, and Magnetic Resonance Imaging (MRI) imaging [19] (Figure 12). The type of procedure and surgeon typically dictate both the imaging and the software used. For THA, templating using 2D frontal and lateral radiographs is the most common preoperative approach that surgeons use.

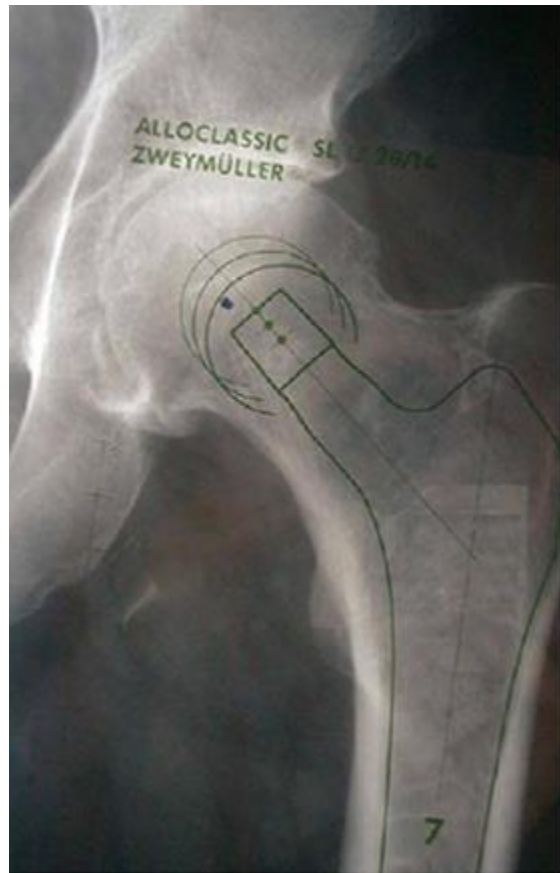


Figure 12. Examples of Upper) 2D radiographic templating (image from Bachour et al. 2010). Lower) 3D biplanar radiographic templating using the EOS system (image from Knafo et al. 2019)

While preoperative templating has become an established part of the surgical workflow today, it is vital to understand the impact of templating on the restoration of the

hip joint postoperatively. As mentioned before, templating is performed to give the surgeon a better understanding of the relationship between the patient anatomic and geometric constraints of the implanted device. Much like the definition above, templating uses patient imaging data to allow the surgeon to identify patient anatomic landmarks, size, and position implants before the procedure. One of the most common methods across THR procedures are the use of 2D digitized patient imaging [34], though, other methods of 3D imaging are becoming more reliable, feasible, and safer for patients (e.g., OPS Corin, HIPOP-PLAN, hipEOS, HIP-PLAN, etc.).

2.3.2 Differences Between Pre-Operative Software

Generally, one of the largest advantages of using templating is the increased efficiency and accuracy of THA procedures [62]. Virtual sizing and positioning of components have allowed surgeons to effectively reduce total time in surgery by reducing the inventory of implanted components and improving the overall accuracy of component placement [19]. While these improvements are beneficial on multiple levels, it is also important to consider the underlying clinical metrics that impact the success of THA surgeries. These include restoration of the femoral center of rotation and the anatomic leg length [23]–[25]. The use of templating to restore these clinical variables has been shown to improve patient outcomes causing it to become a standard approach in THA procedures. Currently, the three most common methods for templating are acetate overlays on digital images, 2D, and 3D digital templating [19]. However, this literature review will primarily consider 2D and 3D digital templating as they are most practiced clinically.

While preoperative templating has been shown to improve both clinical and functional outcomes, using 2D and 3D digital imaging to accomplish this has been a matter of debate historically [19]. The only difference between 2D and 3D imaging is that 2D primarily consists of a single coronal (Frontal) plane image of the pelvis while 3D imaging utilizes a volumetric scan of the same region but uses 3D imaging methods (Figure 13). The advantage of using 2D imaging is a lower dose of radiation than 3D methods as well as a quicker time to complete the templating before the start of an operation [19]. Additionally, 2D imaging methods are more cost-effective compared to 3D imaging as the physical apparatuses needed to capture 3D images require more equipment than 2D. The practicality of 2D imaging does come at a cost; by only using one plane, patient anatomy, and anthropometrics are not fully considered and this can lead to issues with the accuracy of implant orientation and sizing [11], [12], [19], [21], [22]. However, for most surgeons, the potential inaccuracies with 2D imaging do not outweigh the risk, monetary cost, and access of using 3D imaging and thus it has remained the standard imaging method used by templating for surgeons.

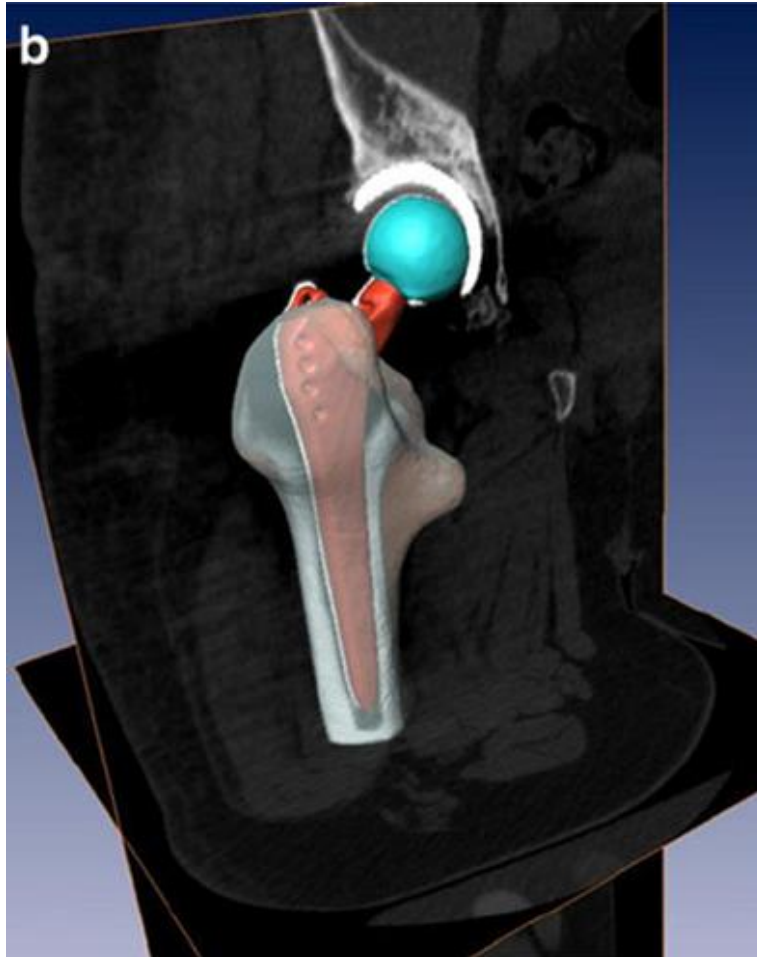
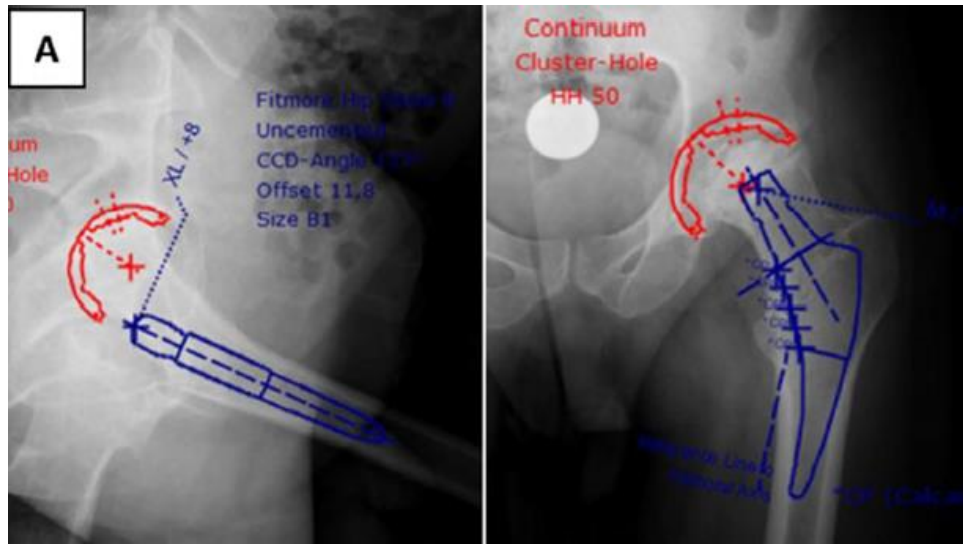


Figure 13: Upper) 2D frontal plane image (image from: Mirghaderi et al. 2022). Lower) 3D Computed Tomography scan with bony and implant geometry segmented (image from Müller et al 2011)

Beyond imaging, comparisons have been made between the accuracy and efficiency of implant positioning using 2D and 3D imaging-based measurements. Most literature in this space agrees that 3D templating is more accurate than its 2D counterpart [19]–[22], [34]. Furthermore, the use of 3D radiographs can also provide patient-specific information for improved accuracy of osteotomy surrounding the affected area, which can improve the accuracy of the placement of the implant.

However, it is observed that the accuracy of both 2D and 3D templating is multifactorial and extends past the limitations of the software itself. Patient intramedullary canal geometry has been observed to potentially be one dictating factor that can influence implanted component placement accuracy [20], [21]. Additionally, it has been observed that implant geometry can dictate the final placement of the implant, as well as the accuracy of the templating software [34]. Patient anthropometric factors such as body mass index (BMI) have also been noted to potentially influence the overall accuracy of 2D and 3D imaging, potentially impacting the overall accuracy of templating software [19]. Exposure to templating software was also thought to impact the overall accuracy of placement [19]. Excluding imaging-related factors, the impact of these variables exists regardless of the type of software used. Patient radiation exposure is known to be high for 3D methods of imaging, specifically CT scans [19], [20]. Despite this cost to patients, recently there has been a shift from 2D to 3D templating coupled with a progressive improvement in the overall accuracy of THA procedures because of this [19].

2.3.3 Evaluation of Templating in THA

The success of THA and templating are influenced by many factors, however, one common metric used to assess the success of both are: restoration of anatomic leg length and restoration of the femoral head center [23]–[25]. Within the context of THA, the leg length measurement refers to the distance from the anterior superior iliac spine (ASIS) to the medial malleolus. This measurement is made for both the implanted and contralateral hip and is later compared between each leg to provide an understanding of if leg length has been restored.

Restoration of the center of rotation occurs at the center of the femoral head. Implanted hardware is aligned such that the implanted center of rotation coincides with the anatomic center of rotation as best as possible. The orientation of the implant occurs in three dimensions with the potential for translations and rotations to occur about each axis of the bone. Restoration of the femoral head center and leg length areas of concern have needed to be solved using a combination of all four components of a THA: femoral stem, femoral head, acetabular cup, and acetabular liner. For example, within one model of the femoral stem, there can be multiple neck angles and neck-length implants that a surgeon can implant. Previous reviews in this space have been divided into two main methods of analysis: implant size and implant orientation. The subsequent review will focus primarily on the femoral stem, though acetabular cup and liner implanted decisions are very relevant.

Predicting accurate implant size and orientation is only one part of correctly reproducing the anatomic function of the joint and may not produce positive clinical

outcomes [63]. Due to this, collecting patient-reported outcomes paired with implantation data can provide insight into surgical outcomes. Questionnaires and studies have been developed to understand joint function, pain, and range of motion concerning the restoration of the femoral offset [24], [25], [29], [31]. The femoral offset being insufficiently restored has led to greater incidences of trochanteric pain, increased abductor moment arm, and altered gait kinematics [25], [29], [31], [54], [55], [64]. This is likely due to the influence that the femoral offset has over the abductor forces required to initiate and maintain gait [64]. Increasing the femoral offset offers a reduction in the abductor energy requirement for gait and the overall reactive force at the articulating surface [64]. A reduction in the femoral offset reduces abductor strength and can increase the incidence of limping among other gait alterations [64]. Within the effects listed above, the type of implant could potentially influence the degree to which the symptoms listed above occur or exist [18]. Implant orientation has been shown to have a range of values that if the femoral center of rotation is inside, can maximize patient-reported outcomes as well as preserve the function of the joint [24]. In a study by Cassidy et al., 249 patients had femoral offset and patient-reported pain scores measured both preoperatively and postoperatively [24]. Finding that increasing the implanted femoral offset to 5mm of the native femoral head or greater did not affect patient-reported functionality, pain, and range of motion scores while decreasing the femoral offset decreased the reported hip joint function [24]. It is important to recognize that these findings were made using 2D templating, and the femoral offset described by Cassidy et al. was defined as the perpendicular line from the center of the femoral head on both the

replaced hip and the contralateral side to a line representing the anatomic axis of the femur. Additionally, restoration of the femoral offset less than 5mm produced poor functional outcomes concerning the range of motion and pain scores [24], [25], [29].

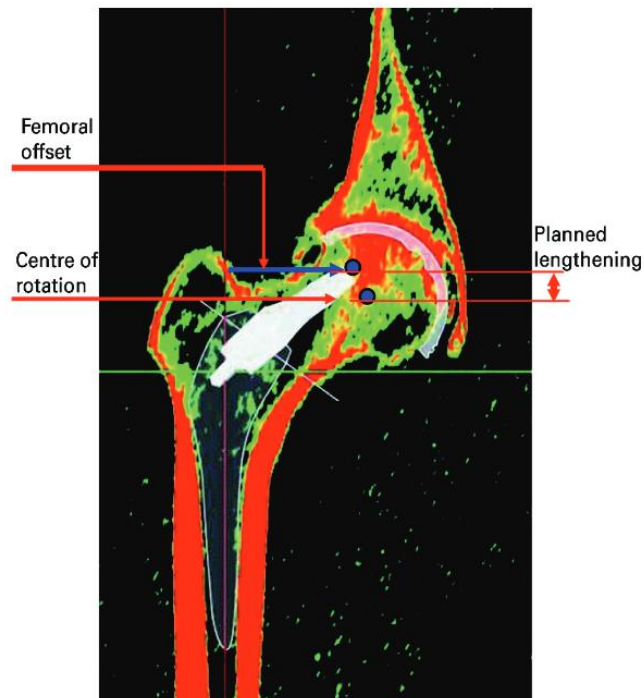
By using templating, the relationship between patient geometry and implant sizing can be better understood. This relationship does make sense, given that the fit of the stem is directly related to the contact with the intercortical canal [11], [12], [35], [36]. Successful implant sizing and templating can also be influenced by the type of procedure, modality of imaging, patient anthropometrics such as BMI, gender differences in anatomy exposure level to surgical software, fixation type, and implant shape, among other factors [34], [35], [63]. Unsuccessful stem sizing can lead to periprosthetic fracturing of the femur if an implant is too large [35]. Additionally, implant sizing that is too small can induce subsidence of the implant and lead to impingement via a cam effect [35]. Regardless of the methodology of imaging used, successful stem sizing must incorporate all of the previously discussed factors to reduce the incidence of adverse effects.

When quantitatively evaluating the restoration of the femoral offset and leg length, correct sizing of all implanted hardware is crucial. An obvious advantage of using templating for implanted component sizing is that intraoperative factors are largely independent of implant sizing making it easy to measure [65]. In a paper by Brenneis et al., the effect of templating on implant size was evaluated using 2D and 3D templating in 51 patients. The findings were that for femoral stems, regardless of 2D or 3D templating, the prediction of the final implant size within one size was roughly the same for both 2D

and 3D (85.7% and 91.3%, respectively) [34]. Additionally, multiple stem geometries were tested in this study, and it was found that in a cohort of 26 patients, shorter geometry stems in 2D had worse predictions than 3D (76.9% and 92.3%) [34]. Other studies reported that 3D templating was more accurate for stems within one size of the final implanted size (94%, 94%, 84%) than 2D templating (83.6%, 68%, 68%) [19], [35], [63], [65], though the same femoral stems were not tested so any interstudy comparison is not possible. For femoral stem sizing, 3D templating could better predict correct component sizing, i.e., sizing that was the same as the final implanted femoral stem. This highlights the importance of using accurate patient imaging for stem size predictions.

Femoral stem orientation refers to the orientation of the femoral stem after it has been implanted in the intercortical canal. Orientation aims to evaluate the restoration of the anatomic femoral offset by measuring the perpendicular line from the center of the femoral head on both the replaced hip and the contralateral side to a line representing the anatomic axis of the femur [24]. The femoral offset in 2D is most commonly represented as a line in the frontal plane of the local anatomic coordinate system, which can be broken into two components: vertical femoral offset (VFO) (translation along SI direction) and horizontal femoral offset (HFO) (translation along ML direction) (Figure 4 Upper). The VFO most commonly drives the leg length discrepancy, while the HFO often drives the abductor muscle moment arms [12], [28]. Further examining the 2D orientation of the implant, 3 degrees of freedom are considered when placing an implant. Referring to Figure 14 Lower, those are Medial-Lateral translation (X), Superior-Inferior translation (Y), and Varus-Valgus rotation. It is also worth noting that the Varus-Valgus

(VrVI) rotation can ultimately influence both SI and ML translations [37]. Expanding from 2D to 3D templating, the degrees of freedom for stem placement expand from 3 to 6 with rotations and translations now occurring about each anatomic axis. While the 3D orientation of the implant expands to now have Anterior-Posterior translation (Z), Adduction- Abduction rotation (Version), and Flexion-Extension rotation (Anterior-Posterior). 2D templating methods traditionally only evaluate a frontal plane rotation of the implant in the intramedullary canal. Figure 3 illustrates the different translations and rotations possible for a femoral stem with the center of rotation being the femoral head. These degrees of freedom combined with patient anatomy can limit the femoral stem and, consequently, the success of THA.



Discrepancy in Six Degrees of Freedom

1) Varus-Valgus 2) Version 3) Anterior-Posterior

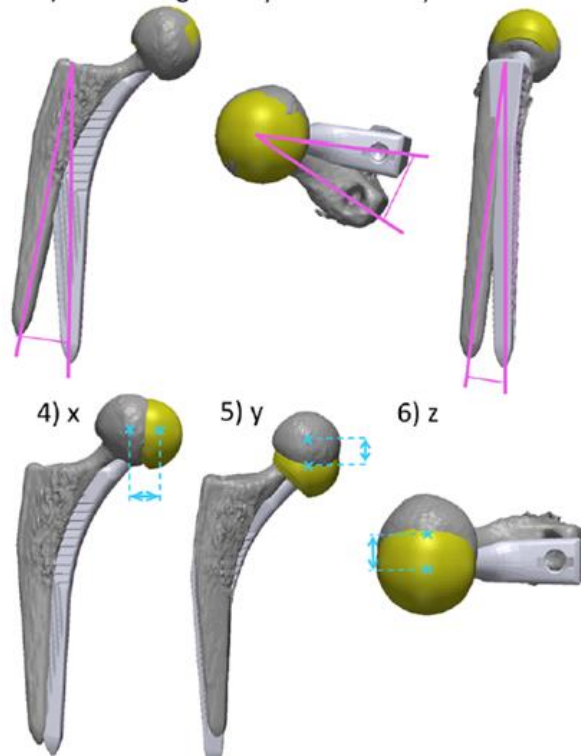


Figure 14: Upper) Image detailing the calculation of the femoral offset (image from Sarili et al 2009). Lower) Visualization of classification of femoral stem degrees of freedom described in the anatomic coordinate system (image from Belzunce et al. 2020).

In literature, the femoral offset and its components are reported in numerous ways. These values have been reported in the literature as either a comparison to the contralateral unoperated leg or as a difference between the pre-operative planned and post-operative implanted values and marked as such in Table 1. The impact of HFO and VFO modification postoperatively is fairly consistent when evaluating the global femoral offset restoration (Table 1).

Interstudy comparisons revealed no consistent trends of lengthening or shortening the femoral offset and there were mixed rationales reported for the measured differences in HFO and VFO. It was reported across multiple studies that there were trends of greater

variability in HFO comparisons than in VFO (Table 1). There was also a trend toward increasing the femoral offset, though the observed interstudy variability was lower than HFO, VFO, and Leg Length. However, the effect on leg length was more variable, with studies that reported leg length showing both an increase and decrease compared to preplanned values or contralateral leg. The variability in leg length post-operation also had a high degree of variability between studies. The increase in leg length is potentially caused in part by surgeons wanting a better fill of the implant to improve diaphyseal and metaphyseal contact [35]. The discrepancy in the variability of all reported values could also be caused by various factors such as the type of implant geometry, surgical approach, and preoperative planning software used that were not held constant between studies [21], [35]–[37], [66].

*Table 1: Horizontal Femoral Offset and Vertical Femoral Offset Literature Comparison. *Results were compared to the contralateral side of the hip instead of the planned approach with no reported standard deviations. † Results were reported as a median number with no standard deviation. Surgical approaches used are reported as follows: PA-Posterior approach, PLA – Posterior lateral approach, AA-anterior approach, ALA-anterolateral approach*

Author(s)	Surgical Approach Used	Number of Patients	Variability in Horizontal Femoral Offset [mm]	Variability in Vertical Femoral Offset [mm]	Variability in Femoral Offset [mm]	Variability in Leg Length [mm]
<i>Amenabar et al. 2015*</i>	PA	44	-1.4	0.4	0.6	0.36
<i>Belzunce et al. 2020</i>	PA	29	2.2 ± 2.8	0.1 ± 2.2	2.2 ± 2.8	NA
<i>Knafo et al. 2019</i>	NA	33	NA	NA	0.3 ± 5.0	-1.9 ± 5.9
<i>Saralli et al. 2009</i>	PA, ALA	223	1.2 ± 2.0	0.73 ± 3.5	0.8 ± 3.1	0.3 ± 3.3
<i>Saralli et al. 2012</i>	A	60	-0.92 ± 5.7 (2D) 1.3 ± 2.6 (3D)	0.33 ± 5.7 (2D) -0.07 ± 2.7 (3D)	NA	1.37 ± 6.4 (2D) -1.8 ± 3.6 (3D)
<i>Hassani et al 2014</i>	A	50	0.4 ± 2.6	-0.2 ± 1.9	1.4 ± 3.1	0.3 ± 2.3
<i>Hu et al. 2023†</i>	PLA	79	NA	NA	5.2	4.7

When evaluating the 3D positioning of the stem, there are noticeably fewer studies examining the variability in stem positioning. Table 2 shows an evaluation of the rotational alignment of the stem. The effects of sagittal rotation and femoral anteversion were not considered or measured in other studies despite how much these rotations can

affect the position of the implanted femoral head [12], [36], [66]. The variance in femoral stem anteversion was captured in Table 2. A similar trend exists with the number of patients recruited, femoral stem geometry used, and surgical techniques. In general, the values of femoral anteversion were reported with a high variance, suggesting that this may be something that surgeons can control for or that many factors may influence the restoration of the native femoral version.

*Table 2: Femoral Stem Anteversion. * Results were reported as a comparison to the contralateral hip Surgical approaches used are as follows: PA-Posterior approach, PLA – Posterior lateral approach, AA-anterior approach, ALA-anterolateral approach, DLA – Direct Lateral Approach*

Author(s)	Surgical Approach Used	Number of Patients	Variability in Femoral Stem Anteversion [°]	Variability in Femoral Stem Sagittal Tilt [°]
Sarali 2009	PA, AL	223	5.8	NA
Belzunce et al 2020	PA	29	-1.5 ± 7.8	0.1 ± 1.5
Rajputra 2022*	PLA	40	12.9 ± 9.8	2.3 ± 1.8
Müller 2011	DLA,ALA	30	10	5.2
Lazennec 2015	NA	110	13.45 ± 13.46	NA
Imai 2016*	NA	65	-3.4 ± 7.1	NA
Inoue 2015	PLA	65	4.0 ± 3.6	NA

The recreation of native version and native alignment as a whole is influenced by implant size, implant orientation, and patient anatomy. As implants grow in size, there is a reduction in the amount of available space the implant has in the intramedullary canal due to diaphyseal and metaphyseal contact, conversely, as the implants decrease in size there is more allowed variability in the alignment [12], [18], [35]–[37]. This leads to a delicate balancing of diaphyseal and/or metaphyseal contact and correct sizing of implants, which influences bone strains post-operation [17], [18] and the aforementioned early- to -medium-term outcomes and surgical complications [16], [17], [27]. This furthers the argument that patient anthropometric information, intercortical geometry, and implant geometry can potentially impact the final alignment of the stem. Furthermore, if surgeons do not consider the effects of 3D translational and rotational alignment of the

femoral stem there could be consequences of misalignment leading to postoperative complications.

The methodology used by Al-Dirini et al. (2018) provides evidence of how stable implant positions can be created by varying the femoral stem alignment. Their use of CT scans to extract external cortical and internal cortical geometry addresses the previous concerns regarding the lack of anatomic intramedullary information and its impact on accurate implant placement. This methodology allows for the determination of the nominal stem size based on a mesh overclosure tolerance, compensation of segmentation error, and following the surgical technique for the implant system. By using Latin Hypercube sampling as the design of experiment, there was an efficient sampling of the perturbations made to the stem alignment from the nominal position, which allowed for multiple implant positions to be evaluated. However, by only analyzing one healthy specimen that is meant to represent the average case for THA procedures, this study does not account for the anatomic variability that exists within a larger population. Moreover, the finding that variation in stem size and position resulted in up to 50% differences in micromotion highlights the need to account for anatomic variability [18].

Despite the current body of evidence on femoral stem alignment, the broad impact of intermedullary morphological variability is understudied and underreported. This is in part due to the limitations of the accuracy of 2D radiographic image-based templating, which is one of the inherent limitations of using planar radiographic imaging. Furthermore, depending on the technique employed, interobserver reproducibility for femoral intramedullary measurements can be low [67]. Presently, there exists no

methodology to quantify the large-scale effect of intermedullary anatomical variance on femoral stem size and femoral stem orientation. This study aims to utilize a large cohort of high-resolution cadaveric CT data to assess and evaluate the effects of bony morphology on stem placement.

CHAPTER THREE: A Virtual Method for Establishing Stem Placement in Total Hip Arthroplasty

3.1 Introduction

Total Hip Arthroplasty (THA) is one of the most common arthroplasty procedures; with more than a million occurring annually [47]. There is no expected decline in the rate of THA procedures, with a projected increase in THA procedures of 71% over the next 10 to 20 years (635,000 cases per year) [4]. The incidence of surgical complications of implanted hardware needs to accommodate the expected rise and improve the survivability of THA components in both the short-and-long-term. Many factors influence the femoral stem's survivability; however, one notable factor has been the alignment of the femoral stem relative to the patient's native anatomy. Stem misalignment has been identified as an early-to-mid-stage postoperative complication [16] causing decreased initial fixation, which can lead to changes in gait, periprosthetic fracture, and aseptic loosening among other post-surgical complications [26], [27], [30]. Stem alignment is thought to be influenced by the implant geometry, patient intercortical anatomy, and femoral stem size [12], [18], [35]–[37].

Recent literature has not explored the relationship between implant geometry, patient intercortical anatomy, and femoral stem size. One reason for this is the current standard of care for THA only requires pre-operative templating using 2D radiographic imaging to

establish component positioning. The use of 2D radiographic images has introduced errors when considering the 3D morphology of a patient's intramedullary anatomy [21], [22]. By not fully considering the intramedullary anatomy, malalignment of femoral and acetabular components, particularly in the sagittal plane (i.e., femoral version and anterior femoral offset), has produced inaccurate femoral head restoration [12], [36]. Other methodologies of capturing 3D patient anatomy often present undesirable radiation risks to the patients (e.g. CT scans) or are too time-consuming (e.g. MR imaging), leading to the use of 2D radiographic imaging out of convenience [19]. Since the final alignment of the femoral stem is thought to be influenced by the combination of patient anatomy and implant geometry, the influence of the shape of the intercortical anatomy on the envelope of potential stem solutions cannot be fully considered using 2D imaging data which leads to inaccuracies in implant orientation and sizing [11], [12], [19], [21], [22].

One of the main limitations of previous studies in this space is the lack of consideration of the variability in femoral intercortical anatomy when restoring the femoral head center. The variability in the anatomy of the bone includes factors such as cortical thickness, femoral version, and femoral bowing. When clinicians use 2D radiographs to reproduce anatomic alignment there is an inherent limitation of the information provided by single-plane radiographs and the accuracy of quantifying key femoral anatomical factors. In particular, altering implant size as a means to restore the head center with variation in intercortical anatomy has not been thoroughly investigated and issues surrounding 2D templating may add additional confounding variables [21],

[22]. As femoral stem alignment and geometry have been shown to impact the primary and secondary fixation of cementless stems, the desire to create a stable femoral stem position may not always restore the head center [26], [27], [49], [56], [57].

Intramedullary anatomy and the effect on potential stem placements have not been reported sufficiently in the literature. The motivation for this study is driven by this lack of information surrounding the effect of intercortical anatomy on the accuracy of restoration of the femoral head center and femoral stem size.

The purpose of this study was threefold. The first goal was to create an automated workflow capable of accurately segmenting the intercortical geometry from existing manual segmentation of the external cortex surface and to extract relevant bony parameters such as the Dorr classification, canal-to-calcar-ratio (CTCR), cortical-thickness (CT), global coronal bow angle (GCBA), global sagittal bow angle (GSBA), local sagittal bow angle (LSBA), femoral neck-shaft angle (FNSA), and femoral neck version (FNV). The second goal was to evaluate the effect of stem sizing and orientation on restoring the femoral head by performing virtual implantations in a large cohort of femurs. The third goal of this study was to validate the algorithm using existing implanted geometry in an experimental cadaveric cohort. We hypothesized that increasing the femoral stem size would reduce the accuracy of the ability to restore the femoral head center.

3.2 Methodology

CT scans from 75 bilateral postmortem L5-toe tip specimens were obtained (Age: 73 \pm 9.7 years, height: 1.72 \pm 0.08 m, mass: 65.94 \pm 15.05 kg, BMI: 22.46 \pm 5.53 kg/m²,

Sex: 48M | 27F). Slice thickness, slice spacing, and image resolution were extracted for each series of DICOM images. Each CT scan included a Mindways QTC Pro phantom (Mindways Inc, Austin, TX), to obtain densitometric information. External cortical geometry was manually segmented for the pelvis and bilateral femurs (Simpleware ScanIP Synopsys, Sunnyvale, CA).

The following methodology describes the process used for automated segmentation of the intramedullary geometry, quantification of femur morphological parameters, and automated algorithm for femoral stem implantation and validation (Figure 15). All relevant functions to both automatic intramedullary segmentation and automated femoral stem implantation are recorded in Appendix A and B.

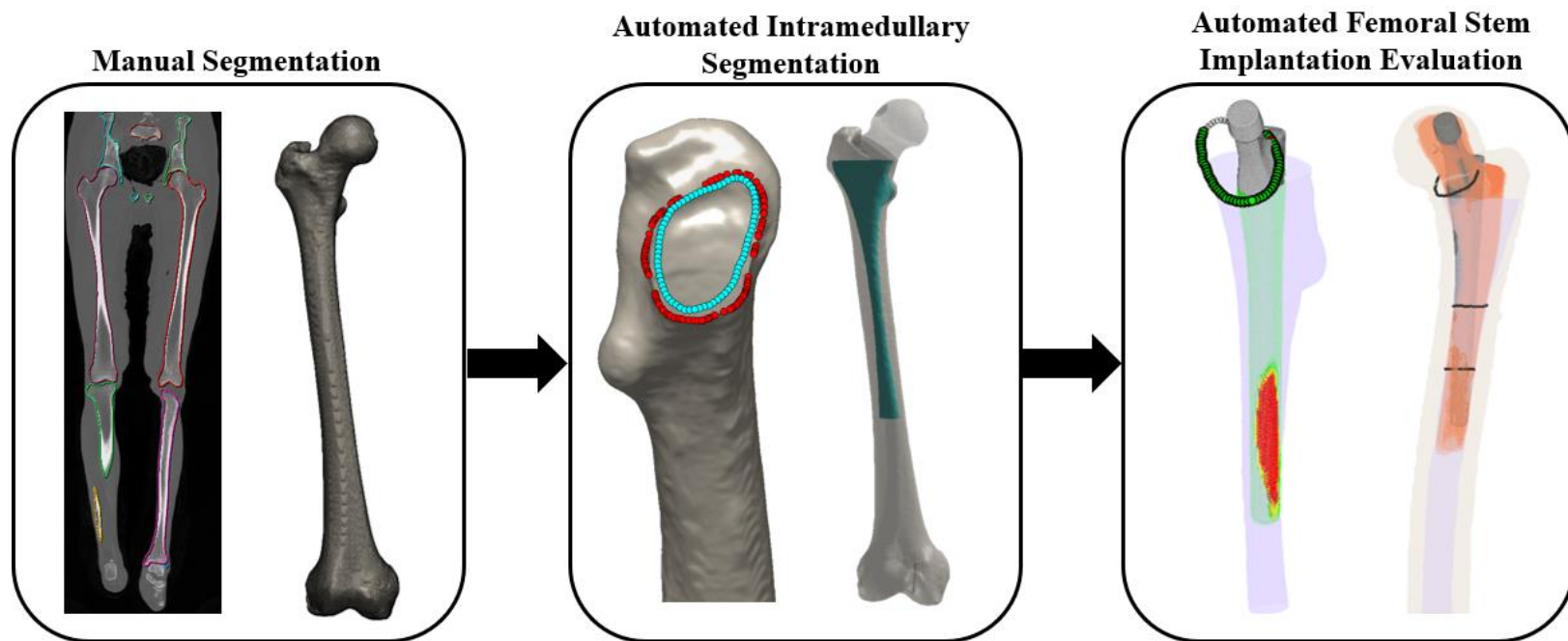


Figure 15: Proposed workflow to accomplish both automatic segmentation of the intercoral femoral anatomy and automated femoral stem orientation and size evaluation.

Intercortical Calcar Wall Segmentation

Intercortical calcar wall segmentation was automated using a custom MATLAB algorithm (Mathworks, Narwick, MA), the DICOM images, and the manually segmented external cortex surfaces. To orient the calcar resection, the sulcus between the femoral head and the greater trochanter was identified from the femur's external geometry, and the frontal plane resection angle was set to 45° passing through the sulcus. A 45° cut plane was chosen based on the surgical guide for the chosen stem geometry. The resection plane was then rotationally aligned such that the normal of the plane was colinear with the normal vector of the proximal femoral neck axis. Cortical nodes within $\pm 0.3\text{mm}$ of the resection plane were identified. The sulcus cut coordinate system was defined with the positive z-axis along the proximal femoral neck axis. The y-axis was perpendicular to the femur's anterior-posterior plane, and the x-axis was the cross-product of the z- and y-axes. A convex hull was defined which bounded the nodes of the calcar resection plane and linearly interpolated into 400 equally spaced peripheral nodes (Figure 16).

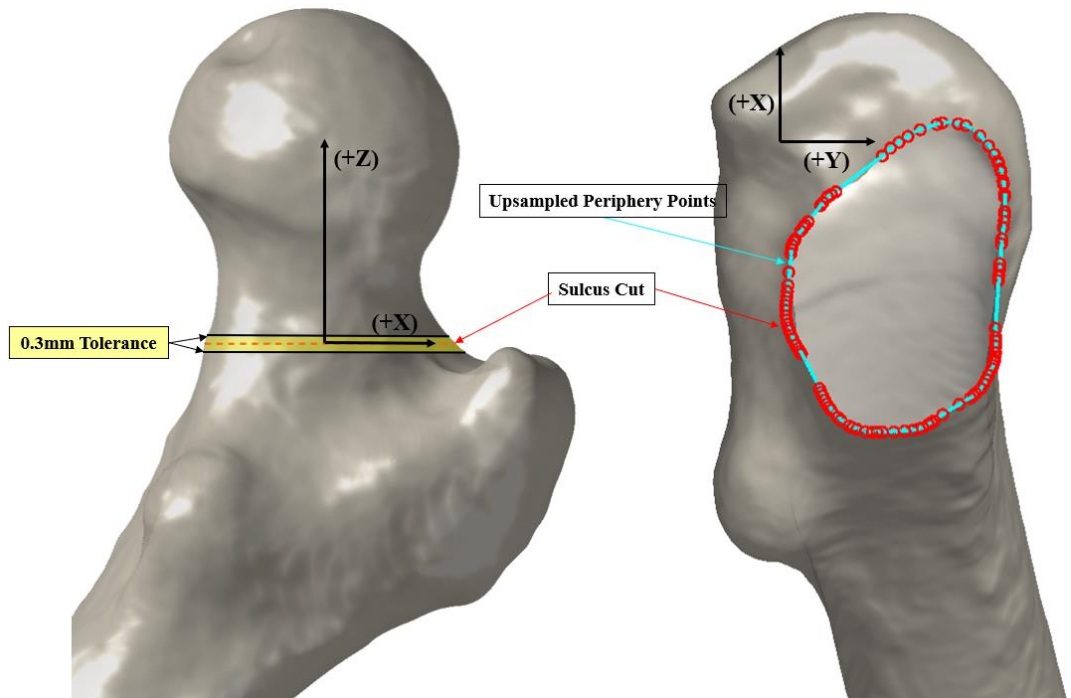


Figure 16: Sulcus cut and example of resampled external surface nodes on a right femur

A radial profile line (RPL) was drawn from the centroid of the convex hull to each node on the sulcus periphery. An additional 400 nodes were equally spaced along each RPL and the associated Hounsfield unit of the bone was extracted from the corresponding voxel in the DICOM image (Figure 17, Upper). A threshold of 75% of the maximum density observed on the RPL was used to identify potential nodes on the intercortical wall of the calcar. The intercortical wall node was defined as the node with the minimum distance to the sulcus periphery. In this fashion, the profile of the inner-cortical wall was calculated circumferentially for each sulcus periphery node. Subsequently, the intercortical nodes were resampled to 100 uniformly spaced nodes using linear interpolation (Figure 17, Lower).

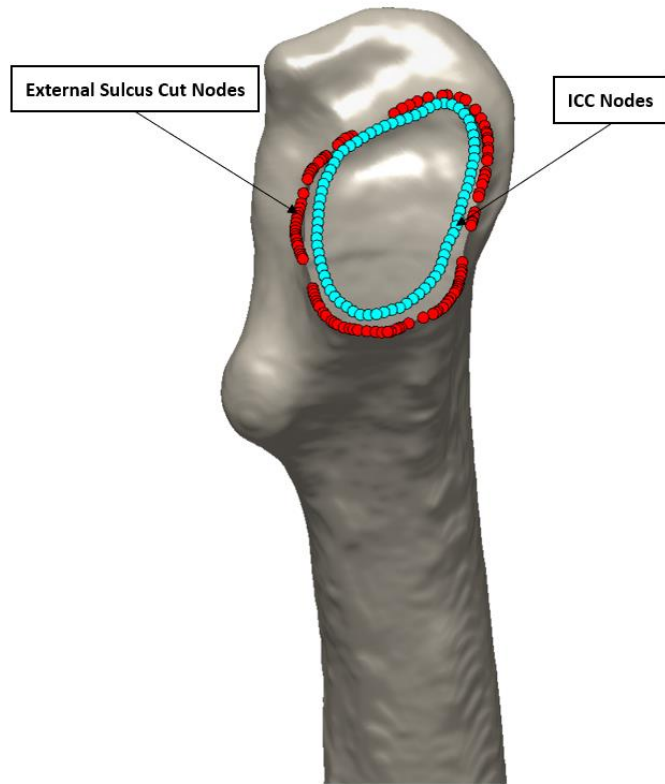
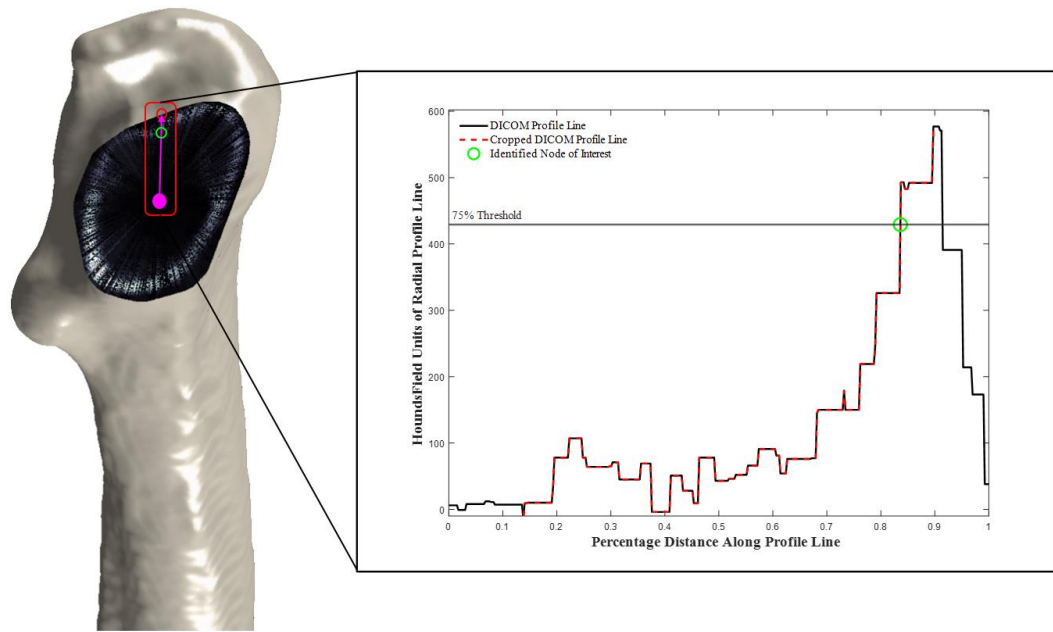


Figure 17: Upper) Visualization of intercostal node segmentation Lower) The final output of the intercostal node segmentation algorithm

Intramedullary Canal Segmentation

Intercortical canal segmentation was performed using a similar custom MATLAB script. The metaphyseal region was defined between the lowest identified intercortical calcar wall node from the calcar resection to a plane located 10mm distal to the lesser trochanter. The diaphyseal region was defined from the distal plane of the metaphysis to a plane at 30% of the overall length of the femur (Figure 18).

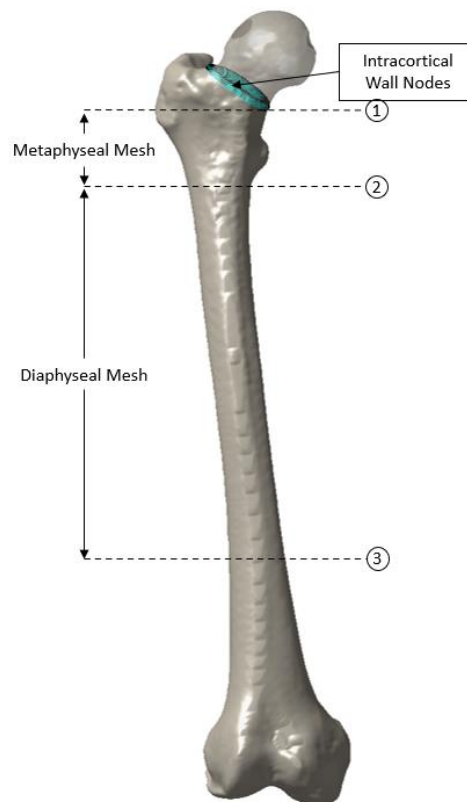


Figure 18: Visualization of the height range used for diaphyseal and metaphyseal classifications. 1) Lowest node on the calcar resection, 2) intersection of the diaphysis and metaphysis 10-mm distal to the lesser trochanter, 3) distal end of diaphyseal region segmentation.

Axial slices were extracted for both the metaphyseal and diaphyseal regions of the femur corresponding to the slice thickness used in the CT scan (typically 0.6mm). For each slice, the external bony surface nodes within half the slice thickness from the

identified axial slice were isolated to form the external cortical periphery. Subsequently, the slice periphery was resampled to 300 equally spaced points using a cubic smoothing spline.

For all diaphyseal slices, which generally had a convex cross-section and high contrast between the cortical wall and intramedullary canal, an RPL was drawn from the centroid of the resampled periphery to each of the 300 periphery nodes. Hounsfield units were extracted along the RPL at 400 equally spaced nodes. A threshold of 40% of the maximum observed value on the RPL was used to identify all potential node solutions corresponding to the inner cortical wall (Figure 19). This threshold was chosen due to the high contrast that exists between the cortical bone of the femur and the cancellous bone. Visual confirmation that the 40% threshold value correctly captured the location of the intramedullary node was observed and confirmed in a smaller sample of femurs. The chosen node was defined as having the minimum distance to the resampled periphery node.

A similar process was performed for all metaphyseal slices. As the metaphyseal cross-sections were often concave, three different points were used as the origin of the RPL to reach each point on the cortical periphery (the centroid of the periphery and points shifted 10mm medially and 10mm laterally of the centroid). A threshold value of 50% was used to account for the lower contrast between the cortical wall of the metaphyseal region and the cancellous bone.

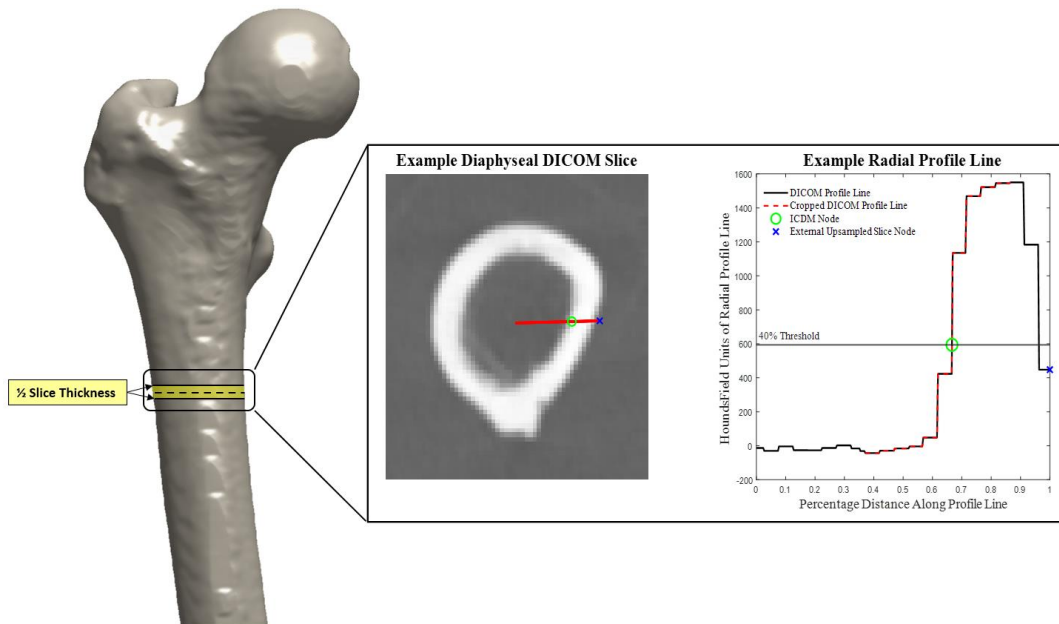


Figure 19: Example radial profile from a slice in the diaphysis. Hounsfield units were mapped along radial lines extending from the centroid of the bone. Inner wall of the cortex was identified using a 40% threshold between the maximum and minimum HU on the line.

The resulting inner cortical wall nodes were meshed and smoothed to remove noise between slices resulting from the segmentation process. The original set of 400 nodes on each slice was resampled to 100 equally spaced nodes, renumbered, and indexed based on the most posterior node on the slice. Connectivity between adjacent slices was established using triangular elements starting with the index node of the first slice connecting to the index node of the adjacent slice, the third node of the triangular element was the closest node identified by a nearest neighbor search on the first slice, processing clockwise around the periphery through all slices. Initial mesh smoothing was performed by fitting cubic smoothing splines (smoothing factor= 0.001) to each S-I column of nodes in the mesh and resampling the A-P and M-L position of each node at each slice height. After initial smoothing using the cubic splines, the mesh was smoothed using a low-pass

Fourier smoothing filter with a cutoff frequency of 500 (cycles/m). Figure 20 shows the mesh through each step of the smoothing algorithm.

Raw Canal Nodes



Slice Smoothed Canal Mesh



S-I Cubic Spline Smoothed Canal Mesh



Fourier Smoothed Canal Mesh



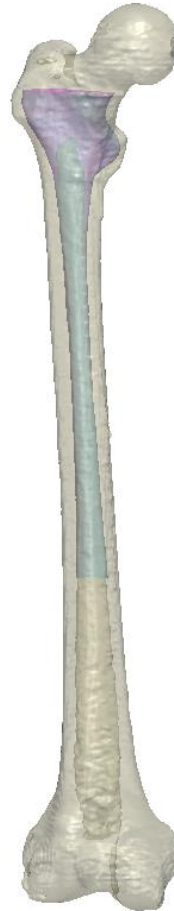


Figure 20: Upper) Refinement of the diaphyseal and metaphyseal medullary canal mesh from identifying raw nodes from the DICOM images, to establishing an initial mesh, to applying smoothing splines along the S-I axis of the mesh, and finally using a Fourier smoothing filter. Lower) Example of femoral segmentation with intercortical segmentation.

Femoral Anatomic Measurements

Bony geometry for each femur in the population was characterized using the following anatomic measurements:

- Calcar-to-Canal-Ratio (CTCR) – the quotient of the canal diameter in the diaphysis and the canal diameter in the lesser trochanter.
- Cortical Index (CI) – the quotient of the inter canal M-L width and the external cortex M-L width measured below the lesser trochanter.
- Global Coronal Bowing Angle (GCBA) – the acute angle between the proximal femoral axis and distal femur axis in the frontal plane.
- Global Sagittal Bowing Angle (GSBA) – acute angle between the proximal femoral axis and distal femoral axis in the sagittal plane.
- Local Sagittal Bowing Angle (LBSA) – acute angle between S-I plumb at native femoral head and proximal femoral axis.
- Femoral-Neck-Shaft Angle (FNSEA) – the angle between the proximal femoral shaft axis and the axis of the femoral neck in the frontal plane.
- Femoral Neck Version (FNV) – the angle between the femoral neck axis and the posterior condylar axis in the transverse plane.

Many of these metrics were based on published techniques for analyzing planar radiographs taken for THA preoperative planning, but, adapted to 3D femoral geometries segmented from CT scans.

The CTCR and CI were calculated following the methodology described by Dorr. et al (1963). CTCR was calculated as the M-L width of the intramedullary canal mesh

measured 3cm distal to the lesser trochanter divided by the mesh M-L width 10cm distal to the lesser trochanter (Figure 21, Left). CTCR was used to classify femora as Dorr Type A ($CTCR < 0.5$), Type B ($0.5 \leq CTCR \leq 0.75$), or Type C ($CTCR > 0.75$). CI was defined as, the ratio of M-L width of the intramedullary canal mesh measured 10cm distal to the lesser trochanter divided by the external cortex width measured at the same height (Figure 21, Right).

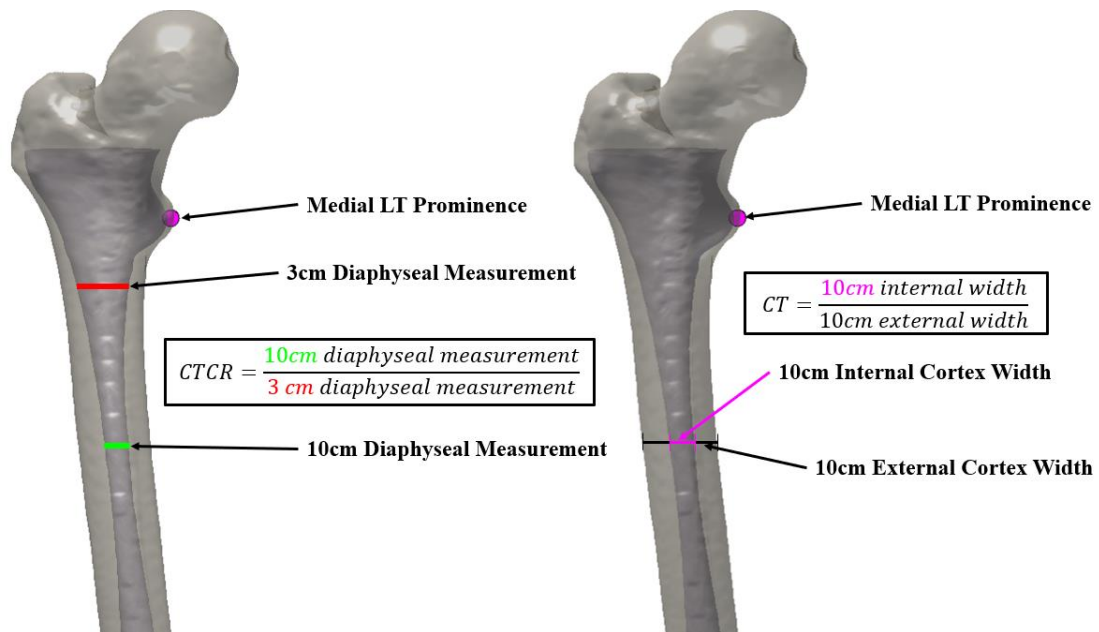


Figure 21: Left) CTCR calculation (LT-Lesser Trochanter) Right) CI calculation (LT-Lesser Trochanter)

The GCBA and GSBA were calculated based on the methodology described by Lee et al (2022). The proximal femoral axis was defined as a vector between the centroids of the outer femoral cortex taken 5cm and 10cm distal to the lesser trochanter (Figure 22). The distal femoral axis was defined as a vector between the centroid of the outer femoral cortex at 7.5cm and 12.5cm proximal from the midpoint of the most distal points

identified on the medial and lateral condyle. These points were chosen not to avoid artifacts caused by flaring of the metaphysis proximal to the femoral condyles. GCBA was defined as the acute angle formed by the intersection of the two vectors in the frontal plane, where positive angles indicated lateral bowing of the femur. Similarly, GSBA was defined as the acute angle formed by the two vectors in the sagittal plane, where positive angles indicated anterior bowing.

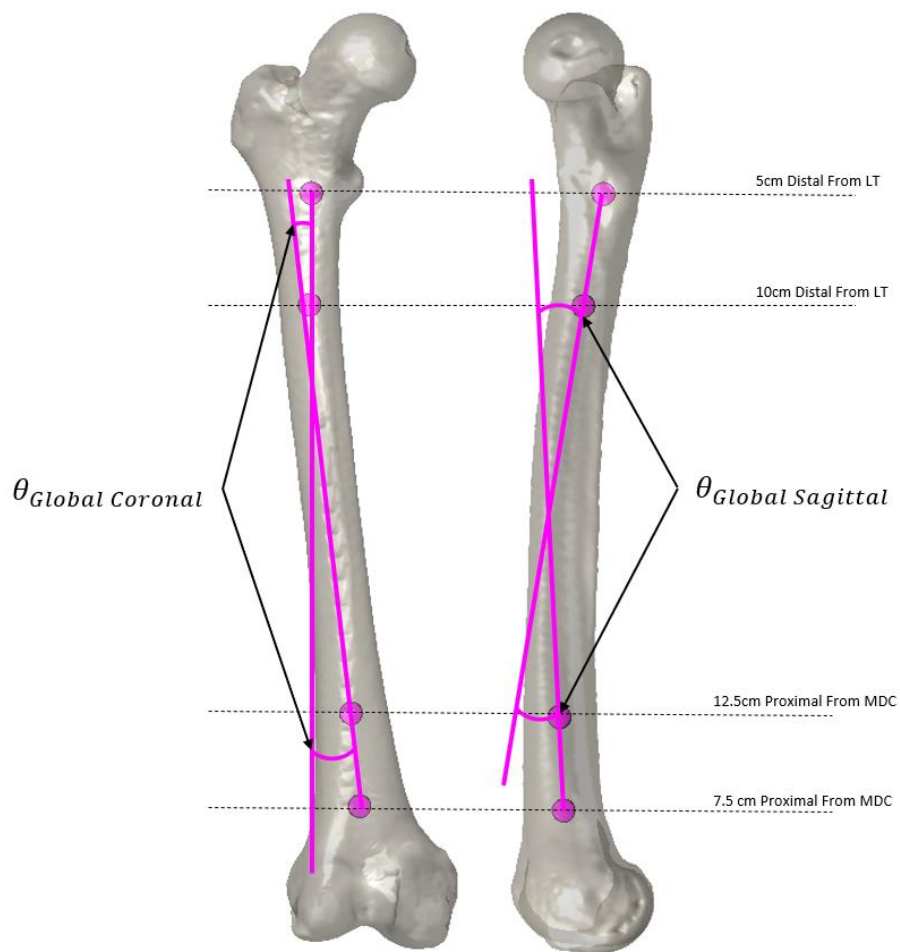


Figure 22: Example femur with lateral bowing GCBA and anterior bowing GSBA. (LT – Lesser Trochanter, MDC- Midpoint of Distal Condyles)

While GSBA quantified bowing over the entire length of the femur bone, localized proximal curvature through the diaphyseal region that engages with femoral stems may

also affect stem alignment. The Localized Sagittal Bowing angle (LSBA) and Femoral-Neck-Shaft-Angle (FNSA) were calculated to quantify the amount of localized curvature of the proximal femur. The same proximal shaft axis determined from the GCBA/GSBA calculations was used in the LSGA and FNSA calculations. LSBA was defined as the acute angle between the proximal shaft axis and the mechanical axis of the femur in the sagittal plane, where positive angles indicated anterior bowing (Figure 23, Left). To calculate FNSA, the femoral neck axis was first defined as the vector from the midpoint of the calcar resection to the center of the native femoral head. The FNSA was defined as the interior angle formed by the intersection of the proximal shaft axis and the femoral neck axis in the frontal plane (Figure 23, Right). Finally, FNV was defined as the angle between the femoral neck axis and a vector connecting the most posterior points on the femoral condyles in the axial plane (Figure 24).

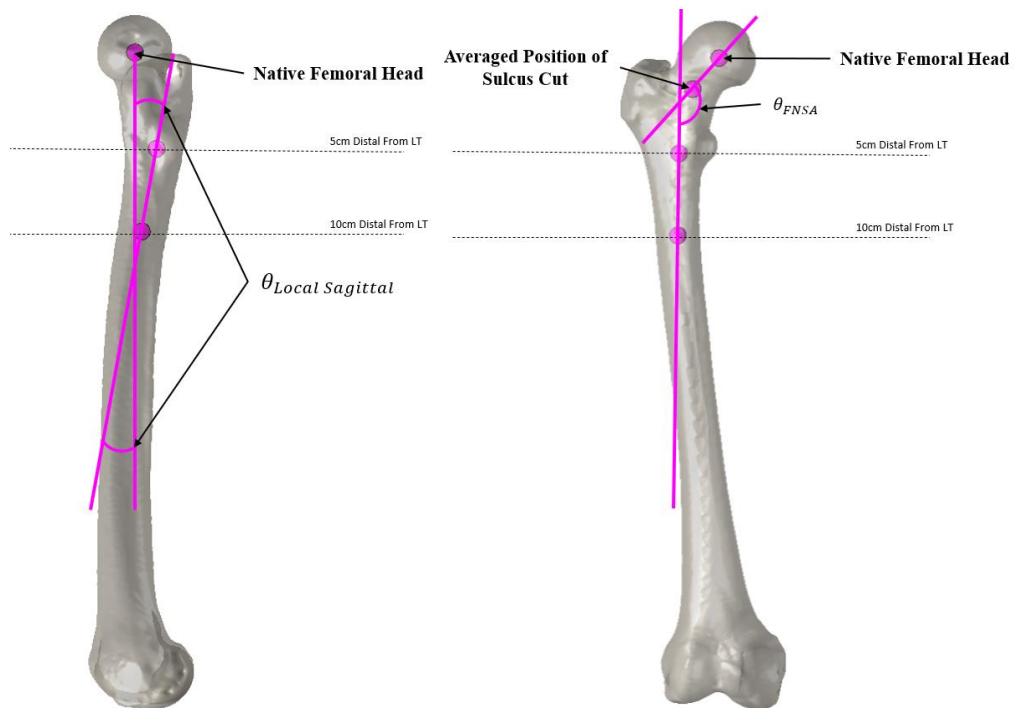


Figure 23: Example of LBSA (left) and FNSA (right) calculations.

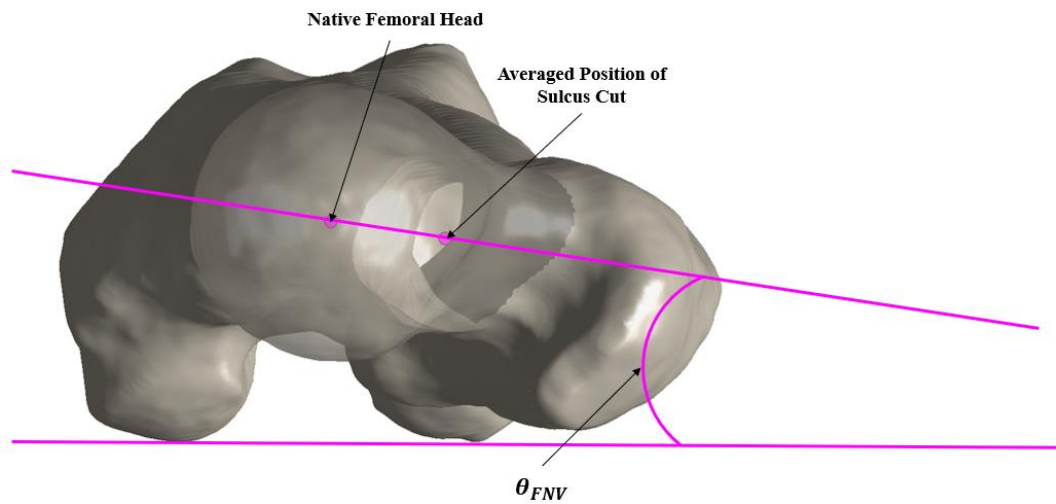


Figure 24: Example of FNV calculation on a right femur

Automated Femoral Stem Implantation

Defeatured collarless cementless 135° standard neck offset Corail® CAD implant geometries (Sizes 8-20) were used in this analysis. To facilitate the automated implantation algorithm, several implant landmarks were manually identified using Hypermesh (Altair Engineering, Troy, MI) (Figure 25). A triangular surface mesh (1mm element size) was created for each size of the implant along with the following landmarks.

1. The center of the proximal face of the trunnion
2. The medial aspect of the neck at the junction with the stem
3. The distal tip of the femoral stem
4. The unit vector normal to the face of the proximal trunnion

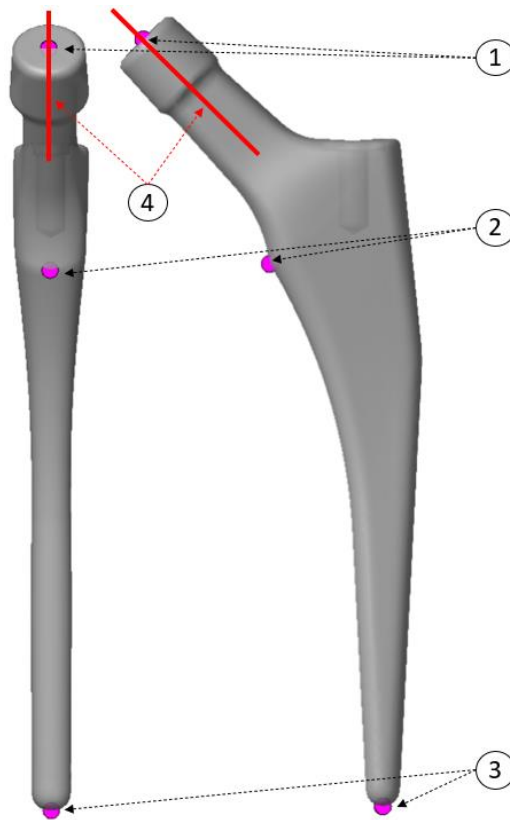


Figure 25: Size 8 Corail® 135° standard offset with standardized landmark locations labeled: 1. center of the proximal face of the trunnion, 2. the medial aspect of the neck and the junction with the stem, 3. the distal tip of femoral stem and 4. the trunnion normal vector (red).

Nominal stem implantations were performed based on the unique anatomy of each femur. First, the local femoral shaft axis was calculated by fitting a line to the centroid of 20 transverse slices taken through the intramedullary canal mesh, starting at the height of the most inferior point on the sulcus resection plane (Figure 26). The axial slices were equally spaced over the length of the longest femoral stem. The stem was then aligned colinear to the local shaft axis constraining the stem's F-E and Ad-Ab rotations and A-P and M-L translations. The stem was rotated about its long axis so that the trunnion axis was pointing towards the native femoral head center defining its I-E rotation. Finally, the stem was translated along the S-I axis until the medial aspect of the neck at the junction

with the stem was at the same height as the most inferior point on the calcar resection. This alignment was performed for every stem size, independent of overclosure with the intramedullary canal mesh of each femur.

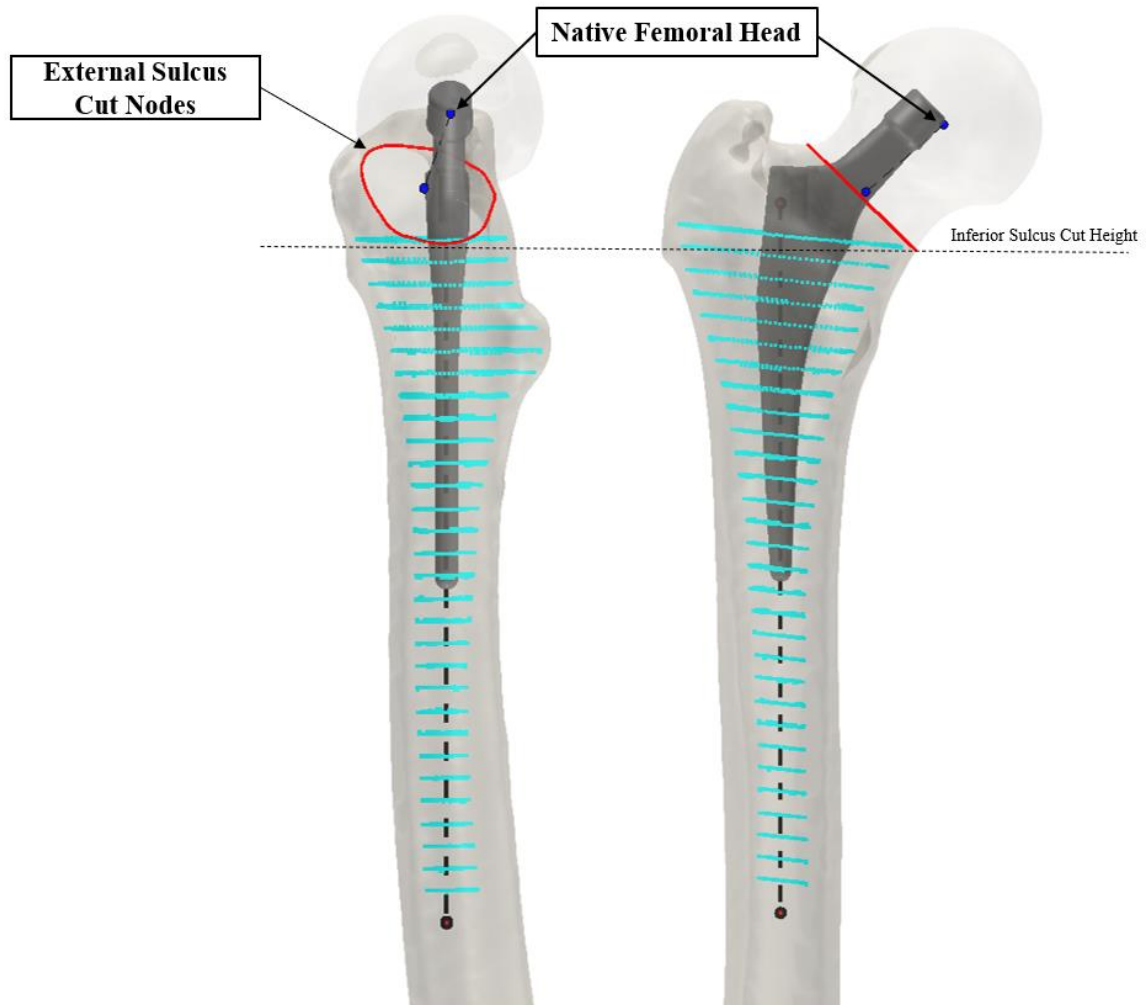


Figure 26: Example of femoral stem geometry aligned with proximal femur axis (shown as black dashed line), rotationally aligned to the native femoral head, and at the inferior sulcus cut height.

Overclosure between the nominal stem position and the intramedullary canal mesh was evaluated by calculating the overclosure distance from each node on the implant mesh to the intramedullary canal mesh [68]. Femoral stem alignment was considered valid when no more than 5% of the nodes were overclosed with the intramedullary canal

and if the maximum observed overclosure was less than 0.5mm. This value was chosen based on a smaller sample of femurs that there was an observed lack of valid implantations for sizes of stems that would have otherwise fit, so to allow the algorithm to consider larger nominal stem sizes than normally wouldn't be implanted.

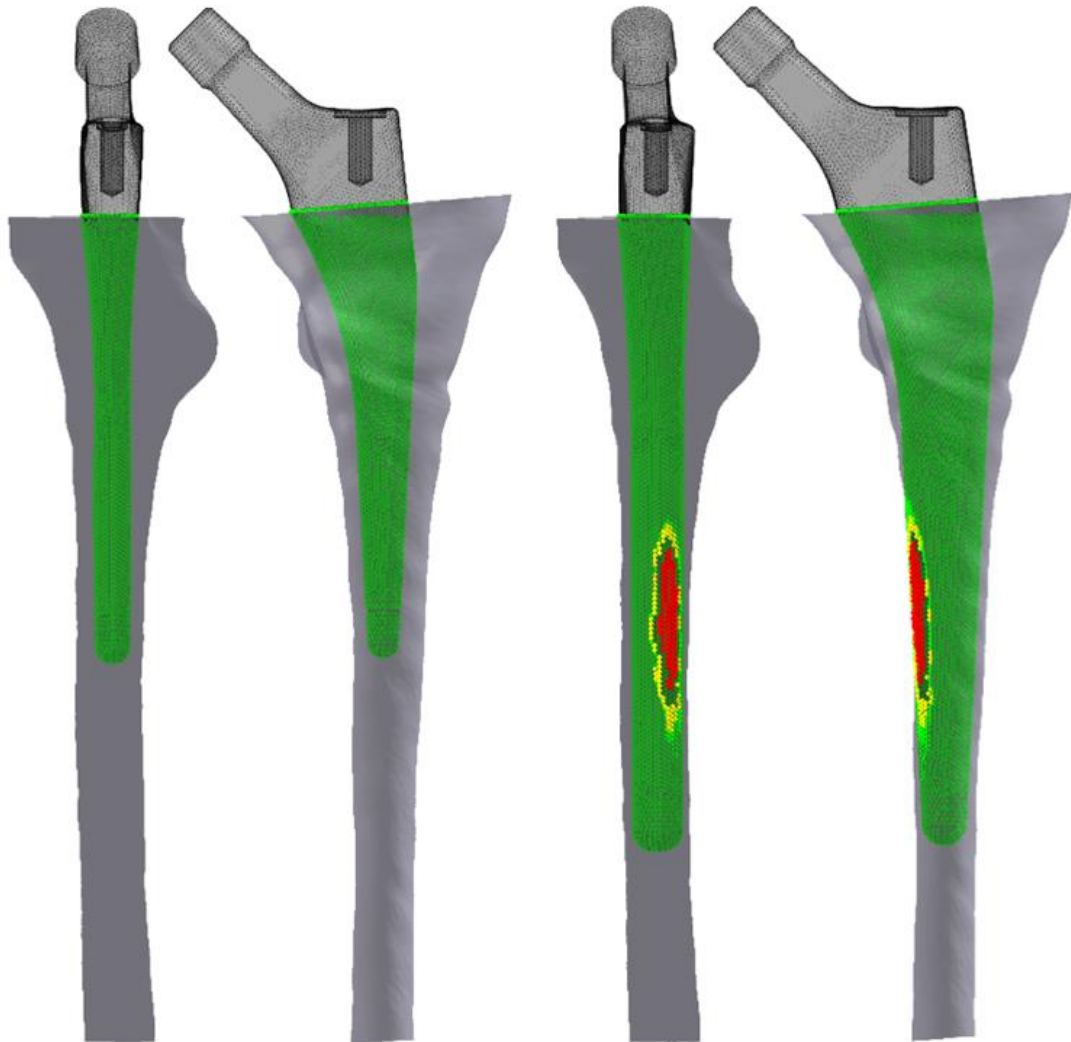
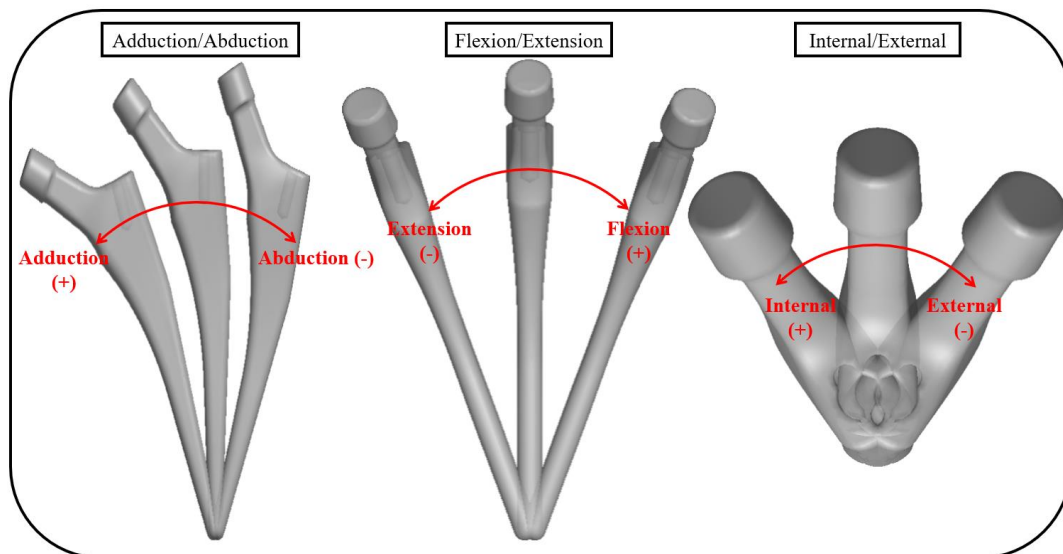


Figure 27: Examples of valid Size 8 Corail® 135° standard offset (left) and invalid Size 13 Corail® 135° standard offset (right) in the nominal implant positions. Green indicates no overclosure, yellow indicates overclosure less than 0.5mm, and red indicates overclosure greater than 0.5mm.

To evaluate the envelope of potential stem positions within the intermedullary canal, a series of alignment perturbations were applied to valid nominal stem sizes. 2,500

perturbed alignments were calculated for each stem size, with perturbations simultaneously applied in all six degrees of freedom about the distal tip of the stem (Figure 27). Perturbations to each degree of freedom were sampled from uniform distributions with ranges based on published literature via Latin Hypercube Sampling (LHS). Ranges for A-P and M-L translations were ± 2.5 mm (Shi et al. 2014) (Table 3). Ranges of F-E, I-E, and Ad-Ab stem perturbations were based on a similar study of stem alignment by Al-Dirini et al (2018) (Table 3). The allowable range of S-I translation was based on the vertical difference between the implanted femoral head center and the native anatomic head for the nominal implant position. S-I translations were sampled from a uniform distribution ranging from 0% to 100% of this difference.



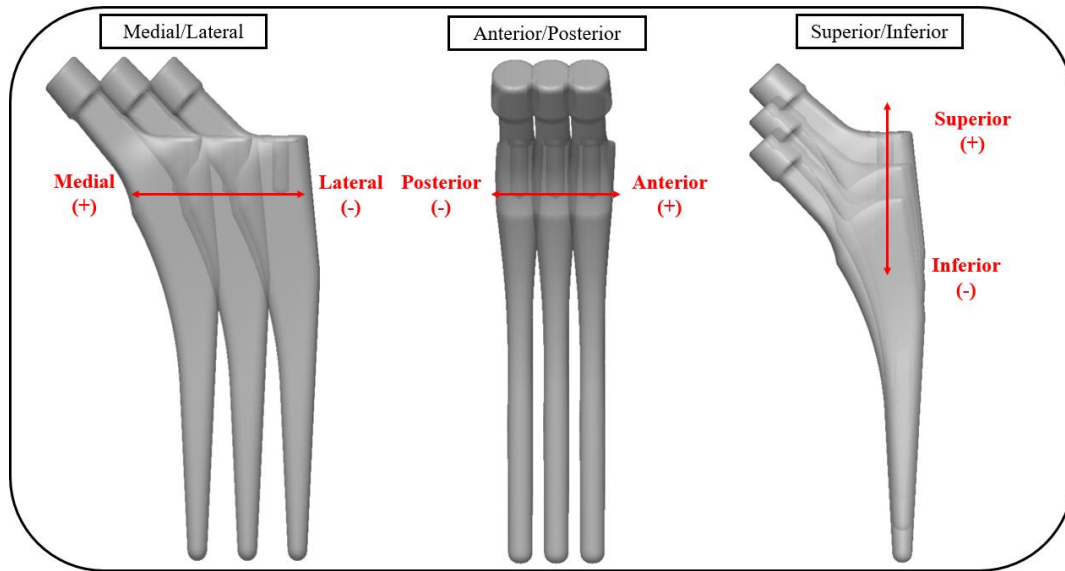


Figure 28: Examples of size 8 Corail® 135° standard offset rotational (top) and translational (bottom) directions of perturbation about the distal tip

Table 3: Range of LHC perturbation values

Degree of Freedom	Range of allowed LHC Perturbation (Minimum Value to Maximum Value)
Internal-External rotation	-20 to 10 (°)
Flexion-Extension rotation	-5 to 5 (°)
Adduction-Abduction rotation	-7 to 10 (°)
Medial-Lateral translation	-2.5 to 2.5 (mm)
Anterior-Posterior translation	-2.5 to 2.5 (mm)
Superior-Inferior translation	0 to 100 (%)

Overclosure between the stem and both the intermedullary canal mesh and inner calcar boundary were evaluated for each perturbed alignment. Alternate alignments were considered valid when overclosure with the intermedullary mesh was less than 0.25 mm and no overclosure was present with the calcar resection. To identify overclosure

between the femoral stem and calcar nodes, the calcar boundary nodes were cropped to the lower 85% of the overall height of the resection. Removing nodes along the top boundary of the calcar resection is consistent with the surgical technique where a reamer is used to remove this bone when overclosure is present. A convex hull was used to isolate the cross-section of femoral stem nodes on the plane of the calcar resection. If the boundary of the femoral stem intersected the intercortical calcar nodes, the implant position was considered invalid (Figure 29).

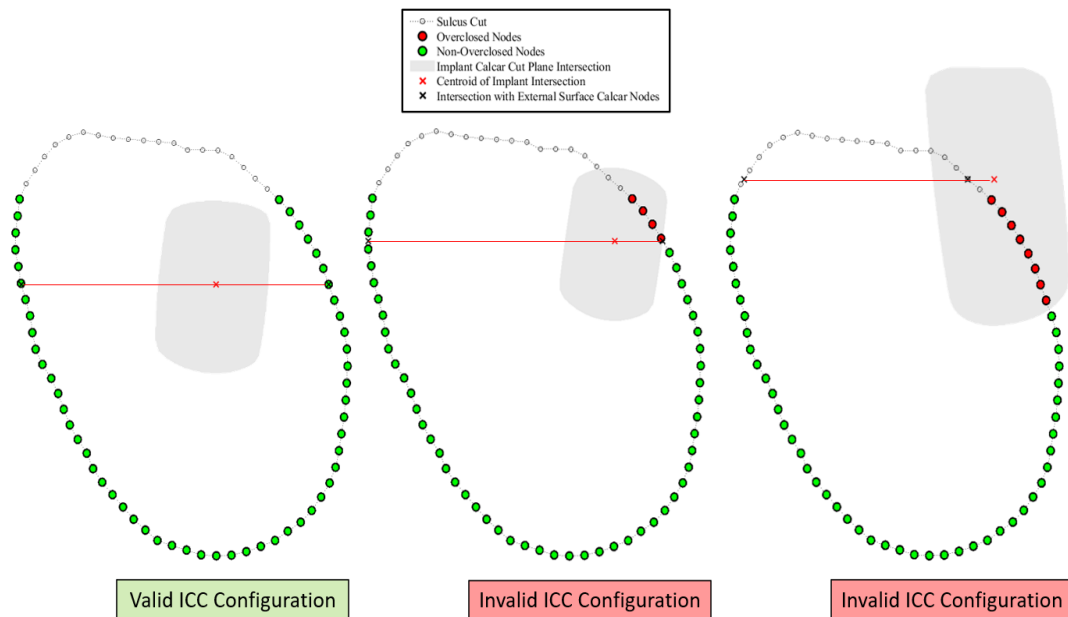


Figure 29: Examples of a valid femoral stem cross-section within the canal's cortical wall (left) and invalid femoral stem cross-sections caused by overclosure of the femoral stem (middle, right). Green nodes represent 85% of the range of external sulcus cut nodes.

The largest size of femoral stem with a valid perturbed alignment for each femur was recorded as the Terminal implant size (T), with smaller sizes referred to as T-1 to T-n for implants 1 to n-sizes smaller, respectively. For valid stem alignments, cross-sections of the stem and intramedullary canal mesh were taken at 20% and 40% of the S-I length of the stem. The canal fill percentage at each height was defined as the ratio of the femoral stem cross-sectional area and the canal cross-sectional area. Additionally, distances were calculated from the most medial, lateral, anterior, and posterior aspects of the stem to the closest points identified along the respective axis on the intramedullary canal mesh in each cross-section. A negative distance indicated overclosure (Figure 30).

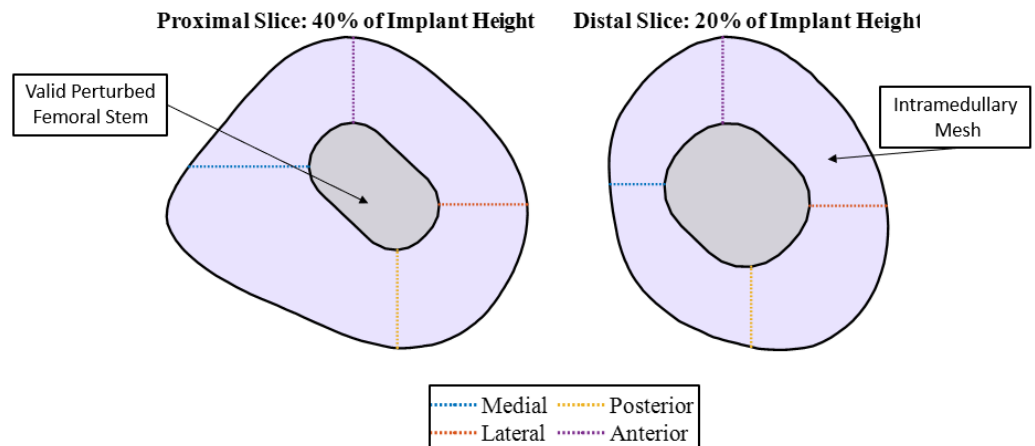


Figure 30: Example of 20% and 40% fit and fill identified metrics, with no measured overclosures. Positive ML and AP distances are notated by their respective colors.

To quantify restoration of the native head center, the unit vector normal to the neck trunnion axis was used to calculate the distance from the implanted head center to the native head center (Figure 31). By accounting for the distance in this way we accounted for the ability to augment the head center along the trunnion axis by using a femoral head or acetabular liner with an increased/decreased offset to restore the femoral

head center. The distance was quantified by first translating the calcar resection plane to the native femoral head center along the femoral neck axis. Then, calculating the intersection of the trunnion normal vector with the translated plane. The vector between the native head center and the intersection point of the trunnion axis was decomposed into components using a local coordinate system assigned to the calcar resection plane, where the x-axis extended from the lowest point on the calcar resection to the highest point, the z-axis was perpendicular to the resection plane, and the y-axis was mutually perpendicular to both the x- and z-axis.

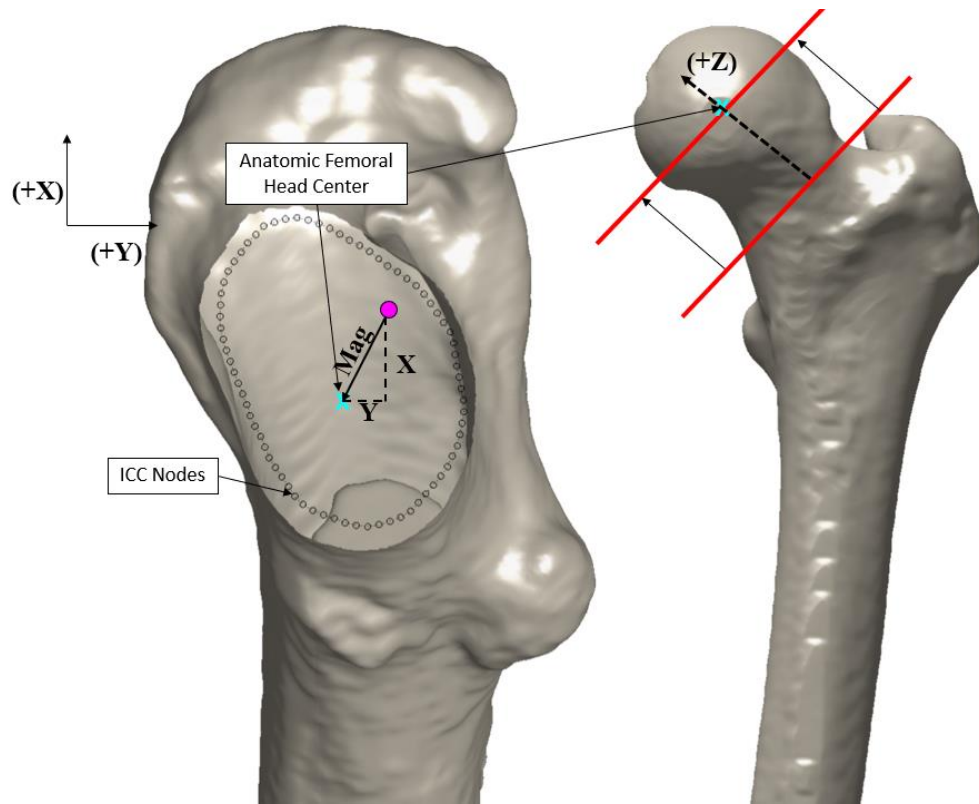


Figure 31: Illustration showing the process for creating the projected calcar resection coordinate system. An example femoral head center intersection with the projected sulcus cut plane (magenta) is shown with the distances in the coordinate system to the anatomic femoral head center.

The *best fit* femoral stem was defined as the valid stem orientation with the minimum distance between the native and implanted femoral head centers. The *best-fit* stem was identified for each valid stem size. The transformation matrix from the hip implant coordinate system to the femur anatomic coordinate system for the *best-fit* stem was used to calculate the anatomic orientation of the stem from the nominal position using the Grood and Suntay notation [69].

Implantation Algorithm Validation

To verify the automated implantation algorithm, the envelope of acceptable implant positions was evaluated in five femurs that were also physically implanted with the same stem geometry as part of a separate cadaveric experiment. All stems were implanted by the same surgeon using the anterolateral approach and a traction table (Table 4). Following implantation, the femora were extracted from the cadavers and denuded soft tissue. 3D optical scans of the implanted femurs were recorded (Artec Space Spider, Artec 3D, Luxembourg, DE). Manual segmentation of the external cortex and CAD of the femoral stem was aligned to the optical scan using an iterative closest point algorithm in the same anatomic femoral coordinate system employed in the automated implantation algorithm.

*Table 4: Implant sizes used for each specimen. * Denotes a collared stem, for purposes of this study collared stems were treated as collarless. For S221141 the left femur was excluded due to poor segmentation.*

Specimen ID	Implant Size (R/L)
S221141	11/-
S221160	8*/8
S221172	12/12

To increase the fidelity of the envelope of potential stem alignments, the number of alignment perturbations was increased from 2,500 to 20,000 using the automated implantation workflow detailed above. The intramedullary nodal overclosure tolerance was expanded to 0.8mm and the S-I bounds of translation were changed to accommodate motion in both directions of the femoral head discrepancy. Two alignment cases were created to evaluate the algorithm:

- Alignment 1: represents the most accurate recreation of the anatomic head center.
- Alignment 2: represents the most accurate recreation of the experimentally placed stem.

Both Alignment 1 and 2 were described as the difference between the experimental stem position and were calculated using the Grood and Suntay description from the nominal position of the stem [69]. For Alignment 2 specifically, each valid stem position had the distance from the nominal position calculated using the Grood and Suntay description and then subtracted from the experimental stems difference from the nominal position [69]. The stem that was defined as being the most accurate recreation of the experimentally placed stem had the minimum cumulative error in all six degrees of freedom (M-L, A-P, S-I, Ad-Ab, F-E, I-E). The experimental calcar cut plane angle and the maximum penetration of the experimental stems with the internally segmented geometry were also calculated.

Statistical Analysis

Pearson's correlation coefficients were calculated between each of the anatomic variables and distances between native and implanted head centers for each most accurate restoration of the head center size of stem. Correlation coefficients were also calculated between the Grood and Suntay description of the stem's most accurate head recreation and the anatomic variables. The classification scheme proposed by Landis and Koch was used to group the observed correlations [70], with correlations between 0.61 and 0.80 representing significant agreements and correlations exceeding 0.81 being classified as strong agreement. One and two sample Kolmogorov-Smirnov tests were also used to test if the distributions of output implantation data were normal with a 5% significance level. The Anderson-Darling test was also computed to verify that the output distributions were normal within 5% significance. Coefficients of variation (CoV) were calculated between the anatomic measurements and implantation to measure the variation present in each distribution.

3.3 Results

A total of 146 bones were successfully analyzed in this study. Two specimens were discarded due to excessive overclosure for the smallest size of femoral stem. The runtime was approximately 25 hours in total using 6 cores of an Intel(R) Core (TM) i7-8700 CPU and 32 GB RAM workstation. On the same workstation, it took on average 1120 seconds to generate the intercortical geometry and 1230 seconds to run the implantation algorithm for one specimen.

Anatomic Measurement

The full cohort of implanted femurs (n=146) had Dorr Classifications that were distributed as follows: 3 femurs Type A (2.1%), 131 Type B (89.7%), and 12 Type C (8.2%). The mean, standard deviation, median, and interquartile range (IQR) were calculated for each anatomic measurement and are shown in Table 5.

Table 5: Boney Parameters for Full Cohort. (CTCR- Canal-to-Calcar ratio, CI – Cortical Index, GCBA – Global Coronal Bowing Angle, GSBA – Global Sagittal Bowing Angle, LSBA – Local Sagittal Bowing Angle, FNSA – Femoral Neck Shaft Angle, FNV – Femoral Neck Version)

Anatomic Measurement	Measurement (AVG±STD)[Median:IQR]
CTCR	0.65 ± 0.07 [0.66;0.08]
CI	0.58 ± 0.05 [0.59;0.07]
GCBA	6.17° ± 2.5° [6.06°;3.6°]
GSBA	13.21° ± 2.4° [13.26°;2.96°]
LSBA	10.34° ± 1.6° [10.33°;2.02°]
FNSA	136.7° ± 7.0° [137.9°;8.3°]
FNV	7.8° ± 8.1° [7.6°;10.9°]
Dorr Classification (A/B/C)	3/131/12

No correlations were observed between the majority of anatomic measurements (Table 6).

Data was tabulated below in Table 6. A statistically significant correlation was observed between the femur’s global sagittal bow angle and the femur’s local sagittal bowing angle (r=0.78). A statistically moderate correlation existed between the cortical index of the femur and the canal-to-calcar-ratio of the femur (r=-0.43).

Table 6: Upper diagonal of symmetric correlation coefficient. Red text indicates potential association and correlation. (CTCR- Canal-to-Calcar ratio, CI – Cortical Index, GCBA – Global Coronal Bowing Angle, GSBA – Global Sagittal Bowing Angle, LSBA – Local Sagittal Bowing Angle, FNSA – Femoral Neck Shaft Angle, FNV – Femoral Neck Version)

	Boney Parameter						
	GCBA	GSBA	LSBA	FNSA	FNV	CI	CTCR
GCBA	1.0000	-0.0130	0.0365	0.0447	-0.0330	-0.0345	0.0506
GSBA	-	1.0000	0.7821	-0.1703	-0.3295	-0.0262	0.2255
LSBA	-	-	1.0000	-0.1334	0.0580	0.1478	0.2593
FNSA	-	-	-	1.0000	0.2453	0.0239	-0.0433
FNV	-	-	-	-	1.0000	0.1402	0.0669
CI	-	-	-	-	-	1.0000	-0.4178
CTCR	-	-	-	-	-	-	1.0000

Implanted Restoration of Native Head Center

For each femur, the number of valid stem alignment perturbations was reduced with increasing implant sizes indicating a tighter fit within the intramedullary canal (Figure 32, Upper).

The mean and standard deviation for the number of valid alignments for the Terminal (T), Terminal minus 1 (T-1), and Terminal minus two (T-2) stem sizes were 7 ± 9 (median=5, IQR=7), 32 ± 27 (median=25, IQR=27), and 83 ± 57 (median=69, IQR=51) respectively (Figure 30, Lower). Figure 32 Lower shows the LHC box plot for the T, T-1, and T-2 femoral stems.

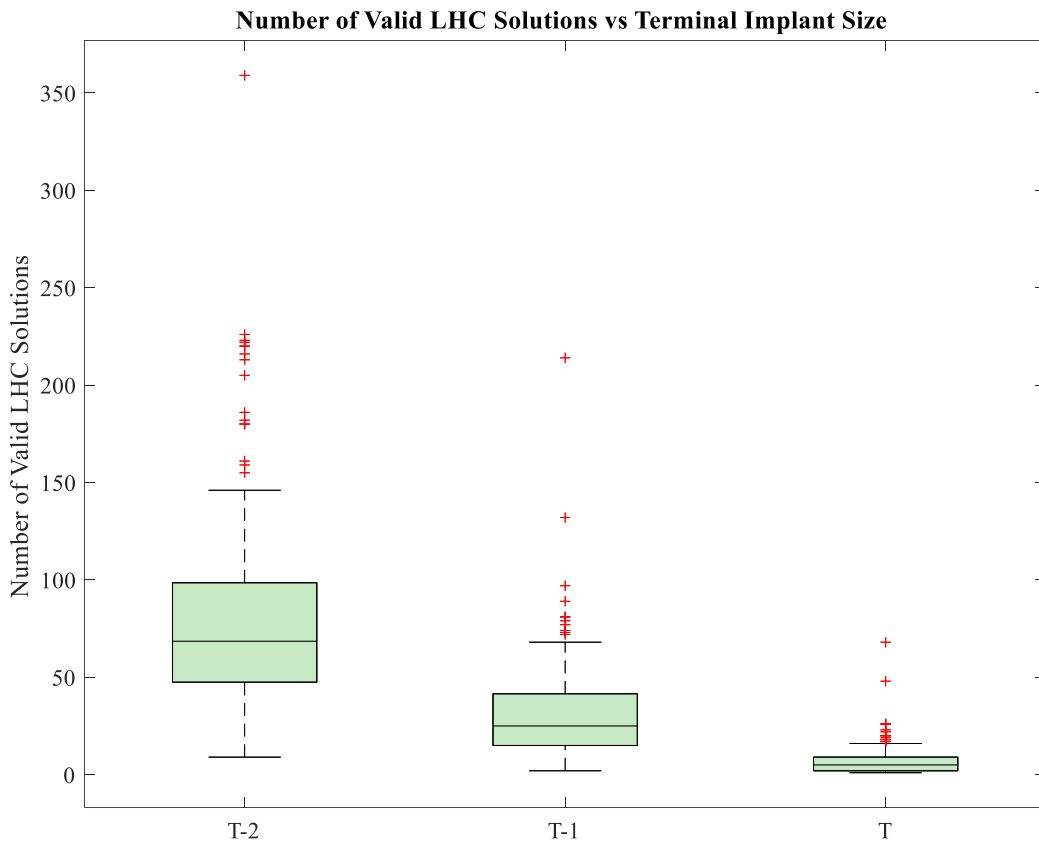
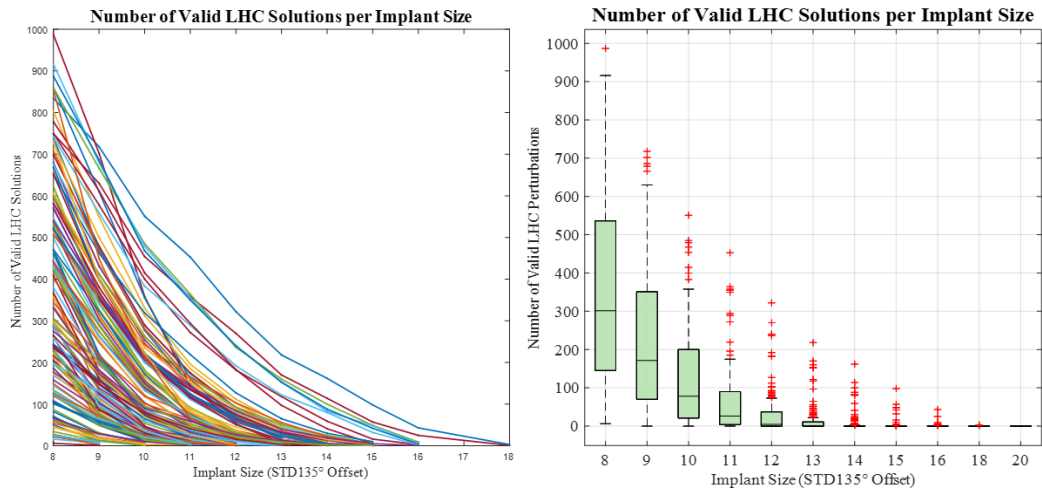


Figure 32: Upper) Variance in the number of valid implant alignments vs size of the implant. Lower) The number of valid alignments for the terminal stem sizes. Statistical outliers are labeled with red crosses.

The mean terminal T size (n=146) of femoral stem implanted was 12 ± 2 (median=12, IQR=3). The mean T-1 size (n=144) of femoral stem implanted was 11 ± 2 (median=11,

IQR=3). The mean T-2 size (n=132) of femoral stem implanted was 10 ± 2 and the median T-2 implant size was 10 with an IQR of 3. An Anderson-Darling test was performed on all three distributions to confirm that the size distributions were normal.

Table 7 shows the implant size distributions for T, T-1, and T-2.

Table 7: Tabulated femoral stem terminal size results. T indicates the terminal stem size, T-1 indicates the next largest fitting stem, and T-2 indicates the next largest fitting stem underneath the T-1 stem size.

Stem Size	Number of Femurs		
	T	T-1	T-2
Size 8	2	12	13
Size 9	12	13	31
Size 10	13	31	25
Size 11	31	25	26
Size 12	25	26	17
Size 13	26	17	12
Size 14	17	12	6
Size 15	12	6	2
Size 16	6	2	0
Size 18	2	0	0
Size 20	0	0	0
AVG \pm STD	12 ± 2.01	11 ± 1.92	10 ± 1.73
[Median,IQR]	[12,3]	[11,3]	[10,3]

The best-case alignment that most accurately restored the native center was identified for all valid stem sizes for each femur (Table 8, Figure 33). An example composite plot showing all valid femoral head sizes as stem size increases is shown below (Figure 34).

The average and standard deviation of the resulting change in the head center for the terminal implant size was $8.2 \pm 4.7\text{mm}$. Relative to the calcar resection coordinate system, the mean implanted head center was $7.4 \pm 4.9\text{mm}$ proximal and $1.3 \pm 3.0\text{mm}$ anterior to the native head center. While the mean difference was smaller for T-1 and T-2 stem sizes, differences of $6.6 \pm 4.2\text{ mm}$ and $6.7 \pm 4.4\text{ mm}$ were still present in the best-fit implants, respectively.

Table 8: Best-case implantation differences in the implanted femoral head center and anatomic in projected sulcus cut coordinate system for T, T-1, and T-2 femoral stems.

	T	T-1	T-2
X Distance [mm]	-7.4 ± 4.9	-6.5 ± 4.4	-6.6 ± 4.5
Y Distance [mm]	-1.3 ± 2.9	-0.06 ± 0.6	0.05 ± 0.3
Magnitude [mm]	8.2 ± 4.7	6.6 ± 4.2	6.69 ± 4.4

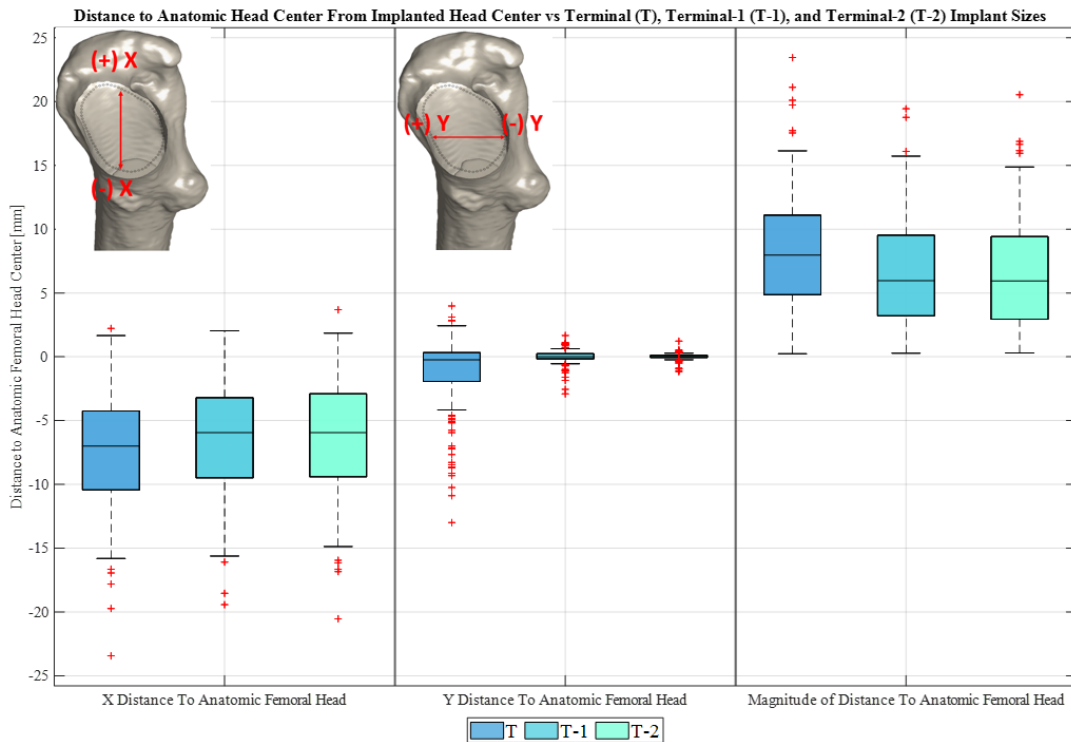


Figure 33: Distance to anatomic head center from the implanted head center for T, T-1, and T-2 femoral stems. Red crosses indicate outliers.

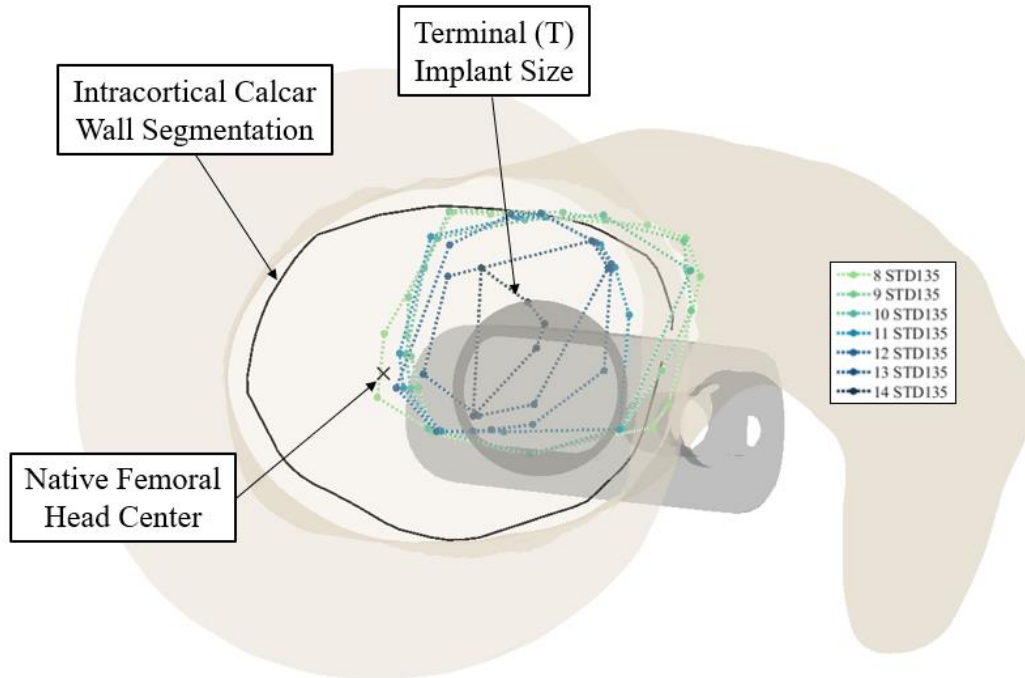


Figure 34: Example composite plot showing the differences in the implanted femoral head center as femoral stem size increases. An example is within the sulcus cut coordinate system.

The differences were calculated from T to T-1 and T-1 to T-2 for the Y direction and the absolute discrepancy in accuracy for femoral head restoration. The probability of the improvement in the accuracy of femoral head restoration was reported below based on the two resulting distributions (Table 9). A two-sample Kolmogrov-Smirnov test was used to distinguish between the two resulting distributions and the null hypothesis was rejected within 5% accuracy.

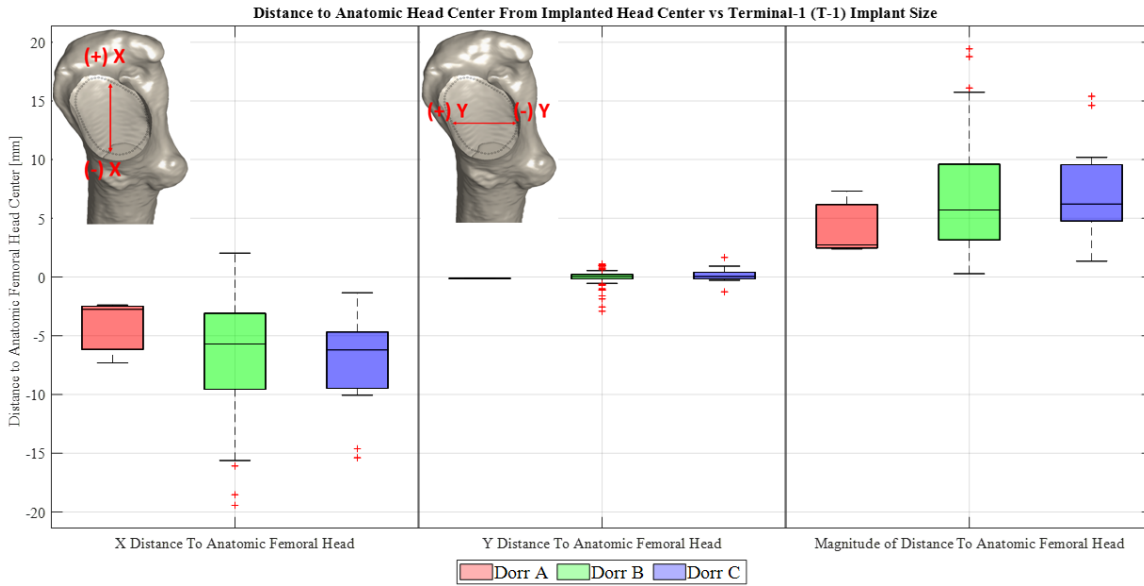
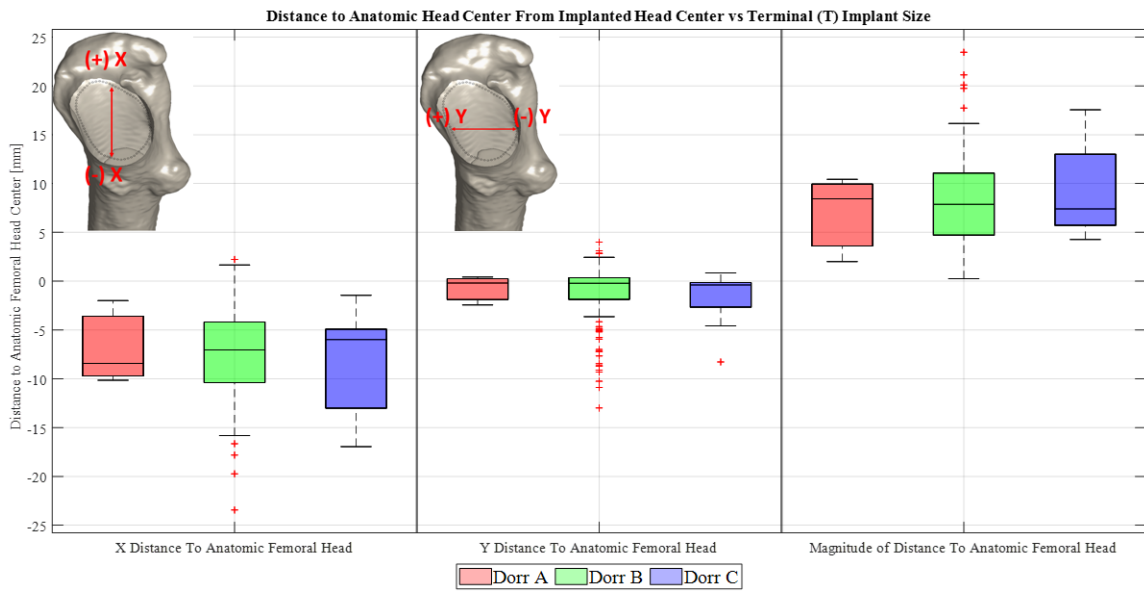
Table 9: Discrepancy in implant size and Y direction restoration of the femoral head

	Incidence	
	T-(T-1)	(T-1)-(T-2)
Y Discrepancy ≤ 0.5 mm	48.23%	87.05%
Y Discrepancy ≤ 1.0 mm	63.97%	100%
Y Discrepancy ≤ 2.0 mm	73.63%	100%

When stratified based on Dorr classification, the Dorr A cohort had the smallest change in femoral head center (7.0 ± 4.4 mm), followed by the Dorr B (8.1 ± 4.8 mm), and Dorr C (9.1 ± 4.5 mm) cohorts (Table 10, Figure 35). No tests for significant differences were performed due to the small sample sizes of the Dorr A and Dorr C cohorts.

Table 10: Distance between implanted and native femoral head center for T, T-1, and T-2 size stems separated by Dorr classification. The numbers under each Dorr Classification are the size of each cohort.

Dorr Type		T	T-1	T-2
(#T, #T-1, #T-2)				
Dorr Type A (3/3/3)	X Distance [mm]	-6.9 ± 4.3	-4.2 ± 2.7	-4.81 ± 2.3
	Y Distance [mm]	-0.8 ± 1.5	-0.1 ± 0.01	-0.03 ± 0.01
	Magnitude [mm]	6.9 ± 4.4	4.1 ± 2.7	4.8 ± 2.3
Dorr Type B (131/129/118)	X Distance [mm]	-7.3 ± 4.9	$-6.4 \pm 4.$	-6.5 ± 4.6
	Y Distance [mm]	-1.3 ± 3.0	-0.02 ± 0.6	0.01 ± 0.3
	Magnitude [mm]	8.1 ± 4.7	6.6 ± 4.2	6.6 ± 4.4
Dorr Type C (12/12/11)	X Distance [mm]	-8.3 ± 4.9	-7.6 ± 4.2	-8.2 ± 4.3
	Y Distance [mm]	-1.7 ± 2.6	0.2 ± 0.7	-0.09 ± 0.4
	Magnitude [mm]	9.0 ± 4.5	7.6 ± 4.2	8.2 ± 4.3



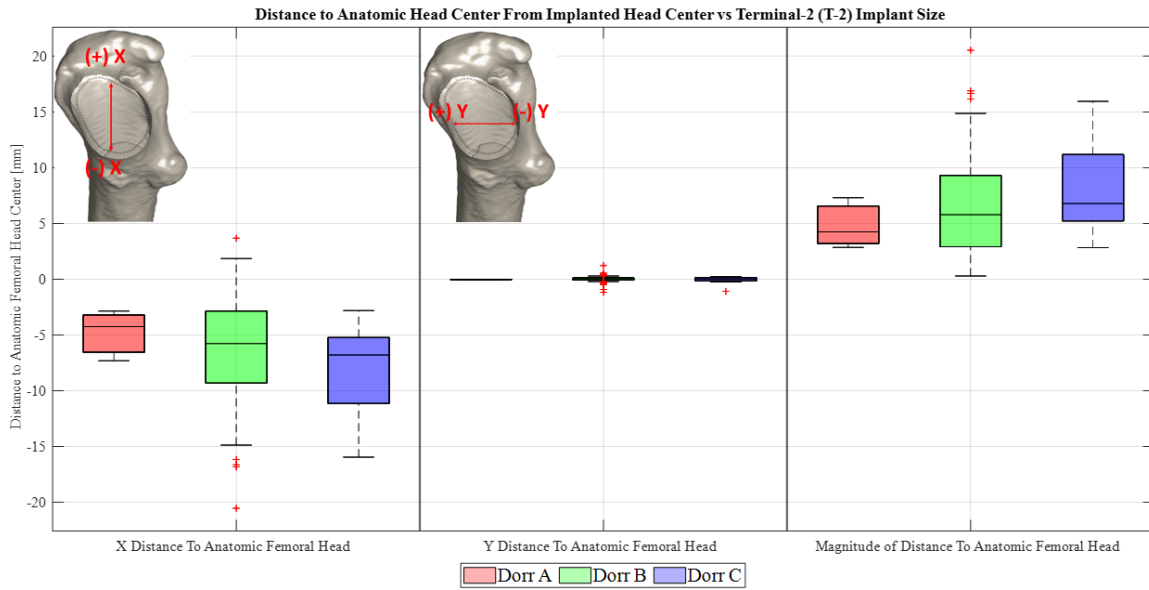


Figure 35: Distance from the implanted head to the native head center for the terminal stem size (Upper), Terminal-1 stem size (Middle), and Terminal-2 stem size (Lower)

No correlations existed between any of the anatomic measurements and the ability of the stem to restore the native femoral head center (Table 11). No correlations also existed between the anatomic measurements and the anatomic location of the stem from the nominal position using the Grood and Suntay description (Table 12).

Table 11: Correlation coefficients between anatomic measurements and changes in head center for the T, T-1, and T-2 stem sizes. Correlations of moderate or higher statistical significance are shown in red.

Terminal (T) Size of Implant			
	X Distance	Y Distance	Magnitude
GCBA	-0.02	-0.01	0.01
GSBA	0.02	-0.04	0.01
LSBA	-0.02	-0.07	0.05
FNSA	0.18	-0.02	-0.16
FNV	0.00	0.02	0.01
CI	0.03	0.00	-0.03
CTCR	-0.11	-0.12	0.14
Next Terminal (T-1) Size of Implant			
	X Distance	Y Distance	Magnitude
GCBA	-0.01	-0.01	0.00
GSBA	0.06	-0.04	-0.05
LSBA	0.06	-0.02	-0.05
FNSA	0.18	0.04	-0.16
FNV	0.01	-0.07	-0.01
CI	0.04	0.01	-0.04
CTCR	-0.14	0.07	0.14
Next Terminal (T-2) Size of Implant			
	X Distance	Y Distance	Magnitude
GCBA	0.05	0.07	-0.06
GSBA	-0.01	0.09	0.01
LSBA	0.03	-0.04	-0.03
FNSA	0.15	-0.15	-0.13
FNV	0.08	-0.18	-0.08
CI	0.05	0.05	-0.04
CTCR	-0.11	-0.07	0.10

Table 12: Correlation coefficients between the anatomic description of the location of the stem from the nominal position and the anatomic measurements for the T, T-1, and T-2 stem sizes. Correlations of moderate or higher statistical significance are shown in red.

Terminal (T) Size of Implant						
	ADAB	FE	IE	ML	AP	SI
GCBA	0.02	0.41	-0.04	-0.13	0.06	0.06
GSBA	-0.25	-0.10	-0.09	0.13	0.08	-0.08
LSBA	-0.41	-0.05	0.11	-0.04	-0.09	-0.03
FNSA	0.01	0.11	0.06	-0.18	-0.03	-0.13
FNV	-0.27	0.15	0.44	-0.34	-0.44	0.08
CI	-0.08	0.00	0.01	0.01	-0.03	-0.20
CTCR	-0.17	0.03	0.12	-0.10	-0.03	0.25
Next Terminal (T-1) Size of Implant						
	ADAB	FE	IE	ML	AP	SI
GCBA	-0.03	0.41	-0.03	-0.15	0.01	0.02
GSBA	-0.27	-0.13	-0.11	0.15	0.08	-0.10
LSBA	-0.46	-0.09	0.12	-0.01	-0.13	-0.07
FNSA	0.00	0.10	0.05	-0.17	-0.02	-0.13
FNV	-0.27	0.12	0.49	-0.32	-0.45	0.03
CI	-0.15	0.06	0.03	-0.04	-0.06	-0.17
CTCR	-0.06	0.01	0.00	-0.09	0.04	0.28
Next Terminal (T-2) Size of Implant						
	ADAB	FE	IE	ML	AP	SI
GCBA	-0.03	0.43	-0.02	-0.19	0.03	0.04
GSBA	-0.20	-0.06	-0.11	0.06	0.11	-0.04
LSBA	-0.39	-0.03	0.09	-0.06	-0.09	-0.04
FNSA	-0.02	0.18	0.05	-0.19	-0.03	-0.08
FNV	-0.28	0.14	0.46	-0.28	-0.45	0.03
CI	-0.15	-0.01	0.02	0.01	-0.06	-0.15
CTCR	-0.03	-0.01	-0.03	-0.04	0.05	0.26

Implantation Algorithm Validation

A total of five femurs had 135° standard offset Corail Femoral stems implanted using the revised methodology. The runtime for all specimens was approximately 2021.97 seconds in total using 6 cores of an Intel(R) Core (TM) i7-8700 CPU and 32 GB RAM workstation. The Grood and Suntay anatomic discrepancy between the experimental and Alignment 1 and Alignment 2 stems are shown below (Table 13, Table 14). The average values of the discrepancy were recorded below for all six degrees of freedom (Table 15).

The experimental changes in cut plane angle and maximum overclosure amount were recorded below in Table 16. Figure 37 and 38 shows the Alignment 1 and Alignment 2 stem positions for each femur.

Table 13: Alignment 1 for right and left legs for terminal (T) size of implants

Specimen ID	Valid Alignments	Discrepancy in Between Alignment 1 and Experimental Stem Position					
		ADAB	FE	IE	ML	AP	SI
S221141R	361	4.37	4.10	-20.87	-10.55	-0.22	14.73
S221160R	710	5.52	2.54	-6.15	-12.36	2.86	15.60
S221160L	903	3.32	2.50	-12.13	-9.23	1.01	15.30
S221172R	33	1.88	4.40	-2.27	1.68	9.54	15.74
S221172L	46	3.47	1.81	-22.96	2.24	0.78	10.62

Table 14: Alignment 2 for the terminal (T) size of implants

Specimen ID	Valid Alignments	Discrepancy in Between Alignment 2 and Experimental Stem Position					
		ADAB	FE	IE	ML	AP	SI
S221141R	361	2.49	2.27	-12.13	-0.35	-7.57	1.98
S221160R	710	1.25	1.61	-1.60	1.11	-2.62	-1.88
S221160L	903	2.31	1.66	0.04	-0.97	-3.76	0.01
S221172R	33	1.60	3.16	-3.04	1.44	5.24	15.99
S221172L	46	3.81	0.29	-13.45	1.46	-6.10	6.44

Table 15: Differences in experimental cut plane angle and the maximum observed overclosure of the implanted experimental femoral stem

Specimen	Valid Alignments	Difference in Calcar Cut Plane Angle (°)	Maximum Observed Overclosure (mm)
S221141R	361	-2.68	1.3163
S221160R	710	1.84	1.2249
S221160L	903	4.73	3.4509
S221172R	33	14.10	1.7691
S221172L	46	12.30	1.3748

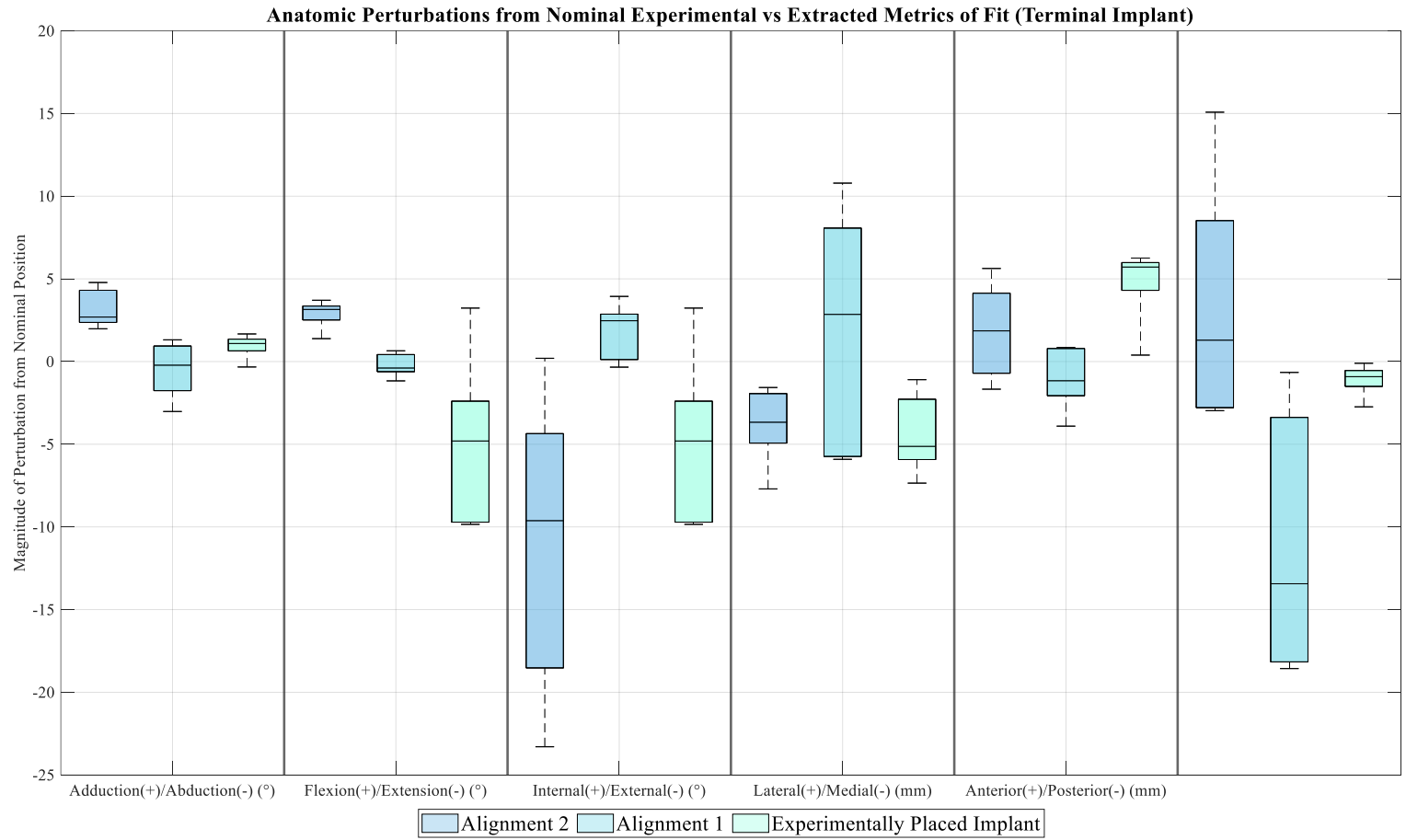


Figure 36: Anatomic description of Alignment 1, Alignment 2, and the experimentally placed femoral stem.

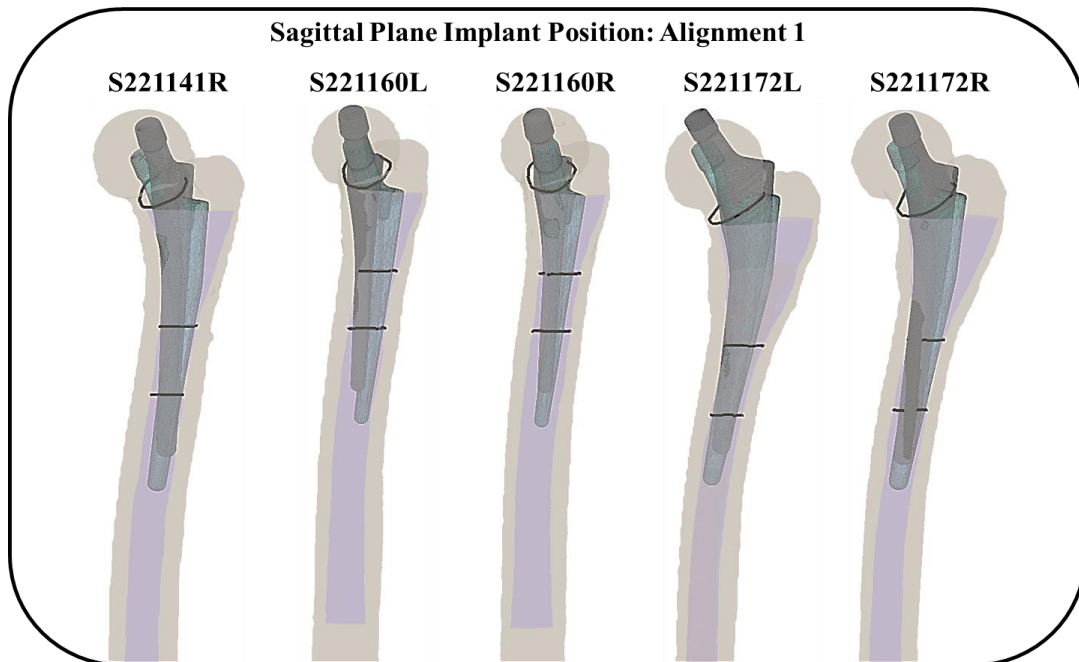
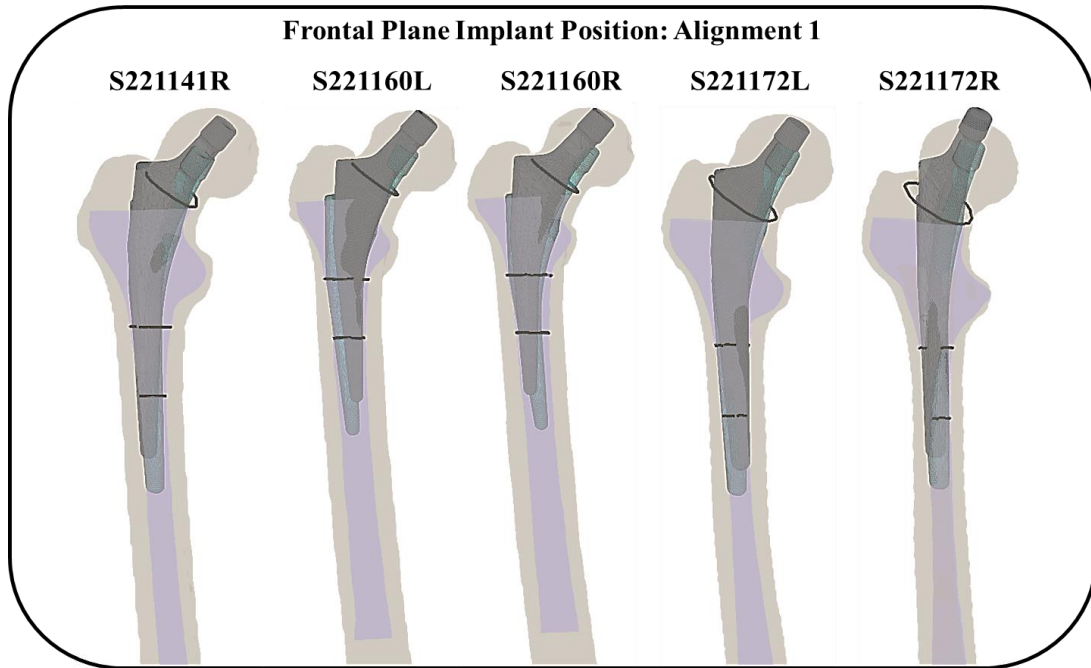


Figure 37: Alignment 1 stem position in the frontal (Upper) and sagittal (Lower) planes. Gray represents the experimentally placed stems, cyan represents the color of the Alignment 1 stem for each specimen, and light blue represents the intramedullary canal mesh.

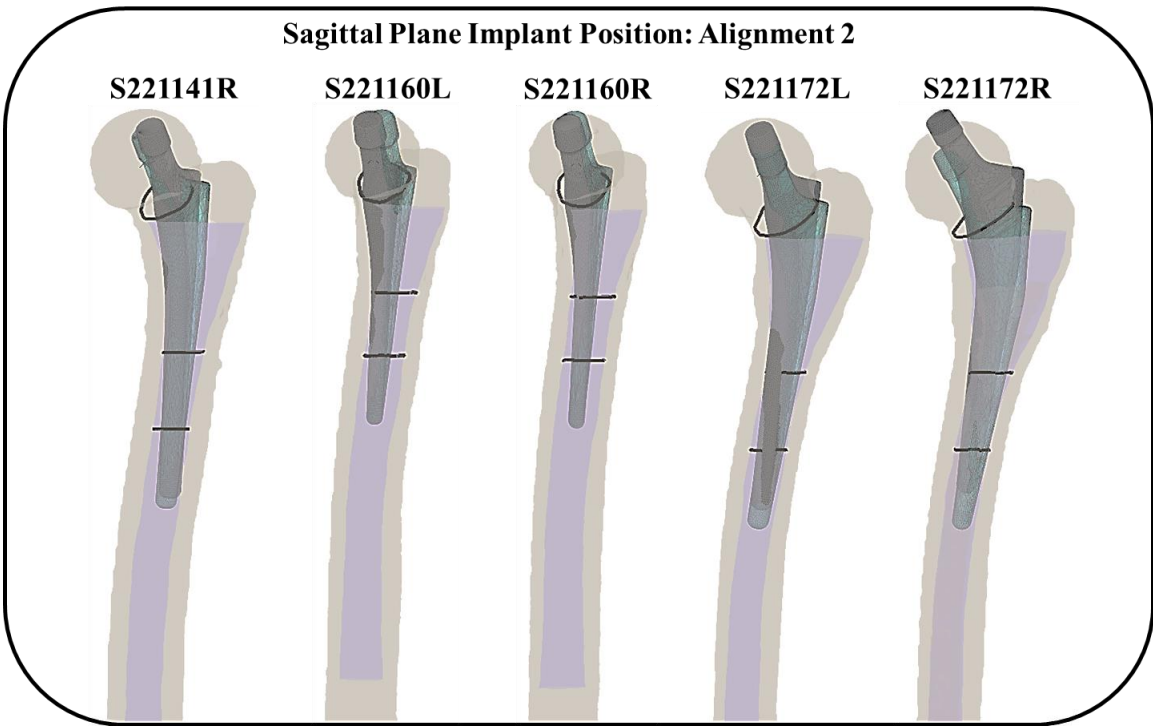
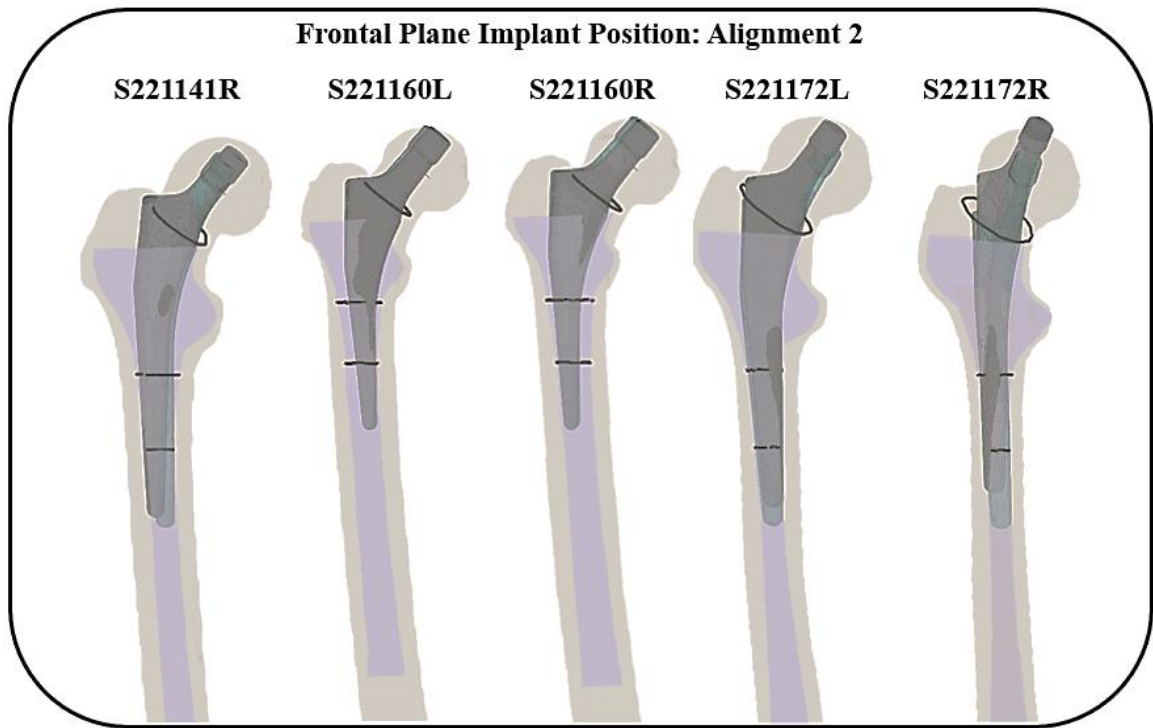


Figure 38: Alignment 2 stem position in the frontal (Upper) and sagittal (Lower) planes. Gray represents the experimentally placed stems, cyan represents the color of the Alignment 2 stem for each specimen, and light blue represents the intramedullary canal mesh.

3.4 Discussion

This study investigated the influence of femur anatomic shape on the ability of a commercially available femoral stem geometry (Corail, Depuy Synthes) to restore the native head center. As the size of the stem approached the terminal implant size there was a reduction in the number of valid alignments. As the diaphyseal and metaphyseal fill of the femoral stem was increased there was less room for the implant to be perturbed, eventually leading to a convergence of femoral stem size to a small family of terminal stem alignments.

These findings were consistent with other literature in this space [12], [18], [35]–[37]. In the literature however, the relationship between the size of the implant and convergence on a smaller family of solutions is not directly reported regardless of the design of the study (e.g., virtual implantations, pre-to-post-operative comparison, clinical outcomes) rather it is mentioned as a potential explanation for the lack of accuracy as femoral stem size was increased. Even as this was reported, the influence of stem size on generating a family of positions was not consistently reported. In a similar study, Xu et al. (2022) virtually evaluated the impact of stem size on the 3D alignment of the stem in eight patients. They observed that increasing stem size resulted in increasing the global femoral offset and the varus position of the stem relative to the femoral head center [37]. Similarly, Al-Dirini et al (2018) virtually perturbed the alignment and size of femoral stems in a single subject while modeling the impacts on stem fixation. They observed that as stem size was increased there was increased positional variation in the I-E rotation of the stem and decreased variation in the F-E and Ad-Ab ranges. Though in each of these

studies, the sample size was very small, and quantifying the family of implantable positions was not the goal, thus the changes in perturbation were not reported. The smaller sample size may not represent the behavior of a larger cohort.

We observed significant differences between the native and implanted femoral head centers, even for the best stem alignments. As femoral stem size approached the terminal size, the majority of variance in allowable positions was in the S-I of the calcar resection plane. Though for the terminal implant size, there was also an A-P direction variance introduced. These results suggest that by under-sizing the femoral stem the head center could be restored more accurately. The restoration of the femoral head center A-P distance discrepancy being (48.23% and 87.05% for T-(T-1) / (T-1)-(T-2), respectively) reduced as the discrepancy in implant size increases suggests that there is potentially a limit to the benefit of decreasing femoral stem size to restore femoral head center. Furthermore, the A-P discrepancy between native and implanted head centers was 0.0 ± 3.6 mm, $-0.1 \text{ mm} \pm 3.7$ mm, -0.1 ± 3.6 mm for T, T-1, and T-2 femoral stem sizes, respectively, which suggests that A-P alignment of the stem is being constrained by contact with the cortical wall of the intramedullary canal, which could ultimately limit the accuracy of femoral head restoration and influence the S-I variability as the size of implant approaches terminal. The correlation coefficients between the measured femoral neck version and A-P ($p=-0.44$, $p=-0.45$, $p=-0.45$ for T, T-1, T-2 respectively) and I-E ($p=0.44$, $p=0.49$, $p=0.46$ for T, T-1, T-2 respectively) anatomic description of the best fit from the nominal position stem did show a moderate correlation which suggests that stem placement from the nominal position to best restore head center is dictated by the

morphology of the intramedullary canal and that contact within the intramedullary canal should be further evaluated.

Another trend was observed in the virtual femoral stem implantations by Xu et al (2022), who identified that increases in femoral stem size introduced increases in global offset and varus position of the stem. Additionally, when compared with the experimental data, the current studies algorithmically aligned stems were on average more adducted which aligns with the findings of Xu et al for the largest size of implant [37]. Though how the internal geometry of the femur was created and used by the preoperative software used by Xu et al could be in a different way than was used by the methodology we used. Although implant size was not perturbed from the experimental femoral stem size in our study, interpretation of these results should be cautioned.

In the clinical environment, restoration of the femoral head centered is limited by ensuring primary stability of the stem. The use of undersized cementless stems to restore the native head center could lead to increased rates of implant subsidence and aseptic loosening [26], [35], [58]. However, if a larger cementless stem is used there is potential for stress shielding and periprosthetic fracture of the calcar during impaction [16]–[18], [35]. In a computational study, under-sizing stems were found to induce large variations in both micromotion and interfacial strains which could result in post-surgical complications [18]. While exploring the impact of femoral stem alignment on bony stress, micromotion, and implant stability was outside the scope of current analysis, this indicates the need for accurate alignment of the femoral stem in a clinical environment.

Suggesting the use of a smaller femoral stem to restore the native head center more accurately should be cautioned. In clinical practice, surgeons could choose implant configurations with different neck offsets and angles to potentially restore the head center. Saraili et. al (2009) mention in a clinical analysis on templating modalities, that modular stems can better accommodate restoration of the femoral offset in cases of disproportion between stem size and femoral offset. Additionally, cases of femoral bone morphologies such as Coxa Vara and Coxa Valga have been thought to influence the accuracy of restoring the femoral head [71]. Modular necks are commonly used to accommodate the variability of femoral neck-shaft angles, which was found to be $136.7^{\circ} \pm 7.0^{\circ}$ for our cohort. One reason for the lack of femoral head restoration could be the lack of modularity in the femoral stem neck employed in our analysis. Use of the standard 135° femoral neck likely created cases where the native femoral neck-shaft angle was not being recreated by the femoral stem increasing the inaccuracy of best-case implant placement, potentially leading to a large discrepancy in the S-I direction.

When compared to the experimentally determined femoral stem positions, from Alignment 1 to produce the experimental stem position there needed to be more adduction, flexion, and external rotation and additionally more medial, anterior, and superior stem translation. When compared to Alignment 2, the discrepancy in alignments improved, and on average the discrepancy in each degree of freedom improved across this cohort (Table 13, Table 14). This suggests that using a larger overclosure tolerance and allowing S-I translation in both degrees of freedom improved the ability of the algorithm to restore the head center in this smaller population of femurs.

During a THA procedure, a surgeon can remove as much cancellous bone as desired when broaching, leading to a canal that can be more conforming to the implant. When impacting the femoral stem, there can be additional cortical bone that is deformed which could further influence the stem's final alignment in the intramedullary canal. It was observed that broaching loads between different types of implant geometries that the force used to broach was on average lower for non-modular stems compared to modular stems [72]. This suggests that the removal of cancellous bone could be an indicator of stem fit. The consideration for the removal of cancellous bone was estimated by the thresholds used during intramedullary canal segmentation. By estimating the level of cancellous bone removal there is potential for inaccuracies of final implant placement and could lead to limitation and exclusion of potentially valid placements based on this threshold in both regions of the intramedullary mesh. These differences could be present in the significant amount of observed overclosure of each experimental stem, which was ultimately higher than the imposed 0.8mm value for the validation cohort (Table 15). There is an indication that the overclosure tolerance could be driving the sensitivity of the number of valid alignments present. This is apparent when results between the larger cohort and the smaller experimental cohort are compared which indicates that there is a sensitivity to reproducing a clinical stem position which is directly related to the S-I direction of translation and the overclosure tolerance. Another impact of the removal of cancellous material is the influence on the path of the stem during implantation, which was not evaluated as part of this study. To further improve the algorithm and to increase the envelope of valid implantation results, the overclosure tolerance should be determined

using a larger database of clinically determined implant positions to optimize both the canal segmentation and the overclosure tolerance.

Despite revising the S-I direction of the implantation algorithm to be in both directions of the discrepancy for the experimental cohort the placement of the experimental stems differed from the most accurate recreation of the head center. Interestingly, there was a large discrepancy between the Alignment 1 and Alignment 2 stems in the S-I direction. Though the clinical success of any of the positions proposed by the algorithm or used experimentally is ultimately unknown. For virtual stem predictions, the lack of consensus on why Alignment 1 differed from Alignment 2 highlights the potential differences adopted when clinically trying to place a stem. In the methodology used by Al-Dirini et al., the S-I offset was adjusted as part of an optimization to reduce the distance from the implanted head center to the overall head center for the nominal implant alignment, before applying alignment perturbations. The production of valid and stable implant positions from this methodology suggests that optimizing the initial position could produce clinically relevant variation in alignments. Furthermore, the diaphyseal and metaphyseal contact of the stem is likely what drives stem orientation [5], [49], so in the observed cases of poor head restoration there was likely insufficient diaphyseal and metaphyseal contact that could have produced improper restoration of the femoral head. The current stem positions predicted may not be clinically meaningful (e.g., prioritizing stability while restoring head center) there was also no reported sensitivity to the overclosure tolerance that affects this. Again, this was not an immediate goal of this study and should be studied in future research.

The calcar resection angles for stems implanted in the experimental cohort deviated from the standard 45° cut plane used in the algorithm for some specimens. Similar to other studies, the virtual calcar cut angle was standardized in other previous studies [18] based on the surgical technique for the femoral stem. The experimentally used cut planes were steepened by 10 degrees for some femurs (2/5). Automatic internal geometry segmentation and the virtual implantations use the cut plane angle to determine nominal implant position, which could explain the lack of agreement with the discrepancy between these variables specifically in the S-I direction where for S221172R/L this was observed to be the highest. For the other specimens the calcar cut plane was within 5 degrees of variation which gave closer agreement in the S-I discrepancy and were altogether closer to the experimentally placed stem. Future work should include exploring the sensitivity of implantation alignment solutions concerning the calcar cut plane.

There were no correlations between the best-case restoration of the head center and the anatomic measurements. This finding suggests that it is difficult to pre-operatively determine whether head center restoration is possible. This finding was counter to our expectations. However, in retrospective reviews, similar findings of no correlation between femoral offset and canal measurements were reported [21], [71]. Additionally, the geometry of the femoral stem used was not consistent between studies in this space, indicating that varying geometry of the femoral stem used could impact the restoration of the femoral head with respect to canal measurements [12], [21], [71]. No information was provided about the variance or type of canal measurements performed making a direct

comparison between studies difficult. However, the majority of our cohort was Dorr Type B (82.9%), which could have influenced the implantation results.

When evaluating femoral bowing, there was no correlation between any of the bowing angles (GCBA, GSBA, and LSBA) and restoration of the femoral head center in the sulcus coordinate system. This was inconsistent with previous findings, which suggested that the patient-specific bone morphology could be an influential factor in restoring the head center, particularly when femoral anteversion was not restored [12], [36]. Exploring the coefficients of variation, no trends towards variability existed when comparing anatomic perturbations from nominal in the rotational directions; furthermore, when expanding to evaluate the Pearson correlation coefficients for the anatomic measurements, there were no trends. This further strengthens the argument that preoperatively determining the femoral center restoration by considering anatomic bowing angles with one fixed femoral neck offset angle may not be possible.

The final parameters that were evaluated were the FNSA and FNV, which similar to other anatomic factors, did not influence head center restoration of the best-case implant alignment. This finding was also unexpected, especially considering the CoV calculated for the FNV data of the entire cohort (1.0305 CoV). A reason for the lack of statistical significance between FNSA/FNV and restoration of the femoral head center could be the number of valid stem alignments that exist within the cohort for the T, T-1, and T-2 sizes of the femoral stem (7 ± 9 , 32 ± 27 , and 83 ± 57 , for T, T-1, and T-2 respectively). Some implantations saw a significant reduction in the family of valid solutions as the stem size

approached terminal, which could further lead to the production of outliers for this dataset, despite the lack of correlation between the FNSA and FNV.

In two retrospective studies evaluating virtual component placement, the relationship between implanted FNV and anatomic version was observed [37], [66]. Despite differences in the geometry of femoral stem used between studies, the results suggest that stem geometry could be an influential factor that leads to variance in the accuracy of head center restoration. The calcar resection angle was controlled differently in each study as well. This could have led to better restoration of the femoral head by accommodating changes in the FNSA or FNV which were ultimately unreported by both studies. In our study, the calcar resection angle was set to 45° and the femoral stem neck angle was fixed at 135° , which could have produced unrealistic femoral head restorations given the FNSA for each specimen, which was not evaluated. This contrasts with the findings of other retrospective studies that suggest accounting for FNSA will have a high correlation between predicted FNSA and the restored FNSA [21]. Within the experimental cohort as well there was an observed difference in the calcar cut plane angle used experimentally and the one simulated virtually, which could have influenced the final location of the femoral stem. Despite these findings not agreeing with the implanted FNSA, the lack of correlation implies that changing stem geometry or calcar resection cut could facilitate better restoration of the femoral head center, this sensitivity was ultimately not evaluated within this study. By accommodating different femoral neck offsets or controlling native anteversion through modular implants the anatomic measurements could be predictors for

restoring head center though the current evaluation only considers one type of femoral stem geometry but allows for these factors to be augmented to observe the impact.

There were several limitations of this study. The findings of this study were based on threshold values from the Hounsfield unit values recorded in each DICOM image. This methodology neglects the bone mineral density, which ultimately influences the ability of the broach to remove intercortical material and the force the surgeon uses to impact the stem and broach. The stem overclosure tolerance was further based on the threshold values and did not match values recorded in the literature due to differences in obtaining the intramedullary canal geometry. However, the existing variability in the implantation data for T, T-1, and T-2 sizes of femoral stem highlights how crucial having a tolerance for evaluating stem overclosures is in determining valid implant alignments under the current criterion. Additionally, the algorithm is currently based on the automatic segmentation of the femur rather than being independent of it. The time to process one subject also presents room to improve the performance of the algorithm and by combining the segmentation and implantation algorithm the user interaction can further be removed improving the efficiency of the algorithm.

The implantation algorithm only predicts allowable implant alignments; however, there are no metrics that quantify stability in the implant. Further, no analysis was performed to quantify the amount of overclosure present in each experimental stem. An adapted methodology used to evaluate the wear of the femoral stem described by Gruen et al could be used in future work to better understand the regions of overclosure as it relates to stem size and alignment [73]. The findings by Al-Dirini et al. 2018 suggest that

the implant's stability depends on position and size, which to run a stability analysis using finite element analysis is costly taking 394 hours [18] the methodology described in this thesis provides a potential surrogate to assess femoral stem sizing and alignment. The existing algorithm does not evaluate the path of the implant during implantation, which could be influential (Xu et al 2022). The implantation path is dependent on surgical preparation of the canal via broaching, which is subject to variability in surgeon impaction force, implant geometry (e.g., modular, or non-modular), and fixation technique (e.g., cementless or cemented). These factors can influence the final location of the stem and, if inadequate material is removed, can influence the location of the stem. In the present study, despite these factors not being accounted for, the differences in best-case femoral stem alignment compared to the experimental alignment could be evidence of just how crucial the removal of intramedullary bony material is. Moreover, how crucial the surgical preparation of the bony canal is to produce a valid virtual stem position.

Other limitations include using a single implant neck offset, which has been observed to influence the restoration of the femoral offset [21], [22], [71]. Expanding the algorithm to accommodate different femoral stem neck offsets could produce a better restoration of the femoral head and potentially show trends in the overall accuracy of restoration of the femoral head with respect to some femoral deformities [71]. The femoral head offset was not investigated in this study; exploring and expanding the algorithm to include a variety of femoral head geometries could potentially allow for an expansion of the results and further accuracy in the restoration of the femoral center of rotation.

The lack of correlations between the femur anatomy and restoration of the head center could be influenced by the configuration of the algorithm. We used conservative thresholding during automatic segmentation of the intramedullary canal that likely reduced the envelope of space in which the stem could be placed. Further, we used conservative limits on the allowable overclosure between the stem and inner wall of the canal, which were considerably smaller than the overclosures observed in the verification experiment. Finally, we limited the range of perturbations applied to the stem's alignment along the S-I axis in the virtual implantation algorithm. Together, these conservative assumptions may have limited the ability of the algorithm to identify viable implant alignments that would have more accurately restored the femoral head center. Post hoc evaluation of a set of exemplary stems found that manual manipulation of the stem could identify stem alignments that more accurately restored the head center and did not overclosure with the intramedullary canal or calcar bone. These alignments were not identified by the original algorithm but were found when the bounds of the S-I alignment and the number of perturbations were increased. An immediate future step should be to evaluate the sensitivity of the model to each of these conservative assumptions and then reevaluate the subject population using fewer conservative parameters.

One of the final limitations of this study is that there were only cementless femoral stems evaluated, the methodology of cemented implantation is entirely different and to simulate the stresses at the implant-cement and bone-cement interface is entirely different. The methodology proposed in this thesis may not provide an accurate understanding of the cement mantle that the implant sits in, despite the success of

cemented THA surgeries being dependent on some of the same factors that dictate the performance of cementless THA femoral stems. Valid locations of the implant using a cemented mantle could be one that induces no cortical contact which was the purpose of this algorithm, though the proposed methodology could be adapted to prioritize restoration within a cemented mantle. The correlation between the anatomic measurements and the restoration of the head center now includes the sensitivity to the thickness of the mantle used to seat the implant. Though analysis performed by Shi et al. [74] utilizes the Gruen zones [73] to evaluate the regions of overclosure which could easily be adapted to observe overclosure within this methodology.

CHAPTER FOUR: Concluding Remarks

4.1 Final Remarks

The work presented in this thesis focused on quantifying accurate femoral stem component placement for Total Hip Arthroplasty (THA) by developing an intercortical anatomy segmentation algorithm. The segmentation methodology developed herein can be expanded to obtain the intramedullary geometry of the femur and understand better what factors influence the accuracy of femoral head restoration more efficiently.

From the work outlined in Chapter Three, we concluded three things. First, as implant size increases and approaches the terminal size of the implant there is a family of valid solutions that the implant converges on. Second, implant sizing approaches terminal greater deviation from the anatomic head center as implant size increases. Third, anatomic measurements were shown to not be impactful when evaluating the restoration of the head center.

The review of current literature in Chapter Two provides insight into the research that has been performed and to better understand the purpose of this thesis and the issues it aims to address. In Chapter Three, a thorough investigation into the effects of various patient metrics on possible implant stem sizing was found, and from this, three significant takeaways were found. First, as implant size increases and approaches the terminal size

of the implant, there is a family of valid solutions that the implant converges on. Second, implant sizing approaches greater deviation from the anatomic head center as implant size increases. Third, anatomic measurements were shown to not be impactful when evaluating the restoration of the head center.

4.2 Future Work

Continuing the evaluation of the methodologies described in Chapter Three would be the most impactful way to expand the research done for this thesis. Running a convergence study and sensitivity analysis on the internal geometry segmentation algorithm to explore the sensitivity of the final meshes. In addition, the algorithms created could be further expanded to extract bone mineral density and produce finite element (FE) ready meshes. These FE meshes could be used as part of a full FEA to test the valid or invalid LHC combinations and to extract the stability and micromotion of the stem throughout the implantation process and activities of daily living.

A future study could expand the validation data set and calculate a clinically informed allowable amount of overclosure to further refine the overclosure tolerance used in this model. Within this study, the sensitivities to each variable in both the automatic segmentation and implantation algorithm could be calculated to better inform the users of these workflows. Separation of the femoral stem into zones similar to the ones proposed by Gruen et al. could also provide a benefit in quantifying implantation trends based on specific locations of stem overclosure [73]. Using statistical shape modeling or morphing to create a family of stem sizes to implant could also provide nodal correspondence allowing for regions of overclosure to be identified regardless of

stem size. The use of principle component analysis (PCA) on the internal calcar and intramedullary geometries could also provide valuable insights into what shape parameters are driving the variation in this data set and could provide further insight into understanding the behavior of the implantation algorithm.

4.3 Key Takeaways

This thesis provides evidence that accurate restoration of the femoral center of rotation using one type of implant geometry is not possible. Furthermore, bone morphology did not impact the accuracy of femoral head restoration, suggesting that the use of anatomic parameters as a metric to add to pre-operative templating may not provide additional benefit. Though this statement is overshadowed by a lack of sensitivity analysis performed and immediate steps should be taken to evaluate which parameters are driving the success of both workflows. With these findings presented, we have hopes to enhance the existing body of evidence and to further improve the quality and mitigate the inaccuracies present within current preoperative THA methodologies.

References

- [1] S. Bahadori, S. Collard, J. M. Williams, and I. Swain, “Why Do People Undergo THR and What Do They Expect to Gain—A Comparison of the Views of Patients and Health Care Professionals,” *Journal of Patient Experience*, vol. 7, no. 6, pp. 1778–1787, Dec. 2020, doi: 10.1177/2374373520956735.
- [2] C. A. Jones, L. A. Beaupre, D. W. C. Johnston, and M. E. Suarez-Almazor, “Total Joint Arthroplasties: Current Concepts of Patient Outcomes after Surgery,” *Rheumatic Disease Clinics of North America*, vol. 33, no. 1, pp. 71–86, Feb. 2007, doi: 10.1016/j.rdc.2006.12.008.
- [3] J. Foran et al., “Total Joint Replacement - OrthoInfo - AAOS,” *Total Joint Replacement*. <https://www.orthoinfo.org/en/treatment/total-joint-replacement/> (accessed Jul. 07, 2023).
- [4] M. Sloan, A. Premkumar, and N. P. Sheth, “Projected Volume of Primary Total Joint Arthroplasty in the U.S., 2014 to 2030,” *J Bone Joint Surg Am*, vol. 100, no. 17, pp. 1455–1460, Sep. 2018, doi: 10.2106/JBJS.17.01617.
- [5] I. D. Learmonth, C. Young, and C. Rorabeck, “The operation of the century: total hip replacement,” *The Lancet*, vol. 370, no. 9597, pp. 1508–1519, Oct. 2007, doi: 10.1016/S0140-6736(07)60457-7.
- [6] K. T. Mäkelä, A. Eskelinen, P. Pulkkinen, P. Paavolainen, and V. Remes, “Results of 3,668 primary total hip replacements for primary osteoarthritis in patients under the age of 55 years: A follow-up of a previous report from the Finnish Arthroplasty Register,” *Acta Orthopaedica*, vol. 82, no. 5, pp. 521–529, Oct. 2011, doi: 10.3109/17453674.2011.618908.
- [7] M. P. Kelly, P. H. Chan, H. A. Prentice, E. W. Paxton, A. D. Hinman, and M. Khatod, “Cause-Specific Stem Revision Risk in Primary Total Hip Arthroplasty Using Cemented vs Cementless Femoral Stem Fixation in a US Cohort,” *The Journal of Arthroplasty*, vol. 37, no. 1, pp. 89–96.e1, Jan. 2022, doi: 10.1016/j.arth.2021.09.020.
- [8] M. Tanzer, S. E. Graves, A. Peng, and A. J. Shimmin, “Is Cemented or Cementless Femoral Stem Fixation More Durable in Patients Older Than 75 Years of Age? A Comparison of the Best-performing Stems,” *Clin Orthop Relat Res*, vol. 476, no. 7, pp. 1428–1437, Jul. 2018, doi: 10.1097/01.blo.0000533621.57561.a4.
- [9] R. B. Bourne and K. Corten, “Cemented Versus Cementless Stems: A Verdict Is In,” *Orthopedics (Online)*, vol. 33, no. 9, Sep. 2010, doi: 10.3928/01477447-20100722-24.
- [10] N. Wirries, M. Örgel, M. Schwarze, S. Budde, H. Windhagen, and M. Skutek, “Cementless total hip arthroplasty with anatomic-shaped implants. Does the minimal invasive anterolateral technique influence the stem position or subsidence in contrast to the standard lateral approach?,” *Arch Orthop Trauma Surg*, vol. 142, no. 9, pp. 2389–2395, Aug. 2021, doi: 10.1007/s00402-021-04122-5.
- [11] S. G. J. Eckrich, P. C. Noble, and H. S. Tullos, “Effect of rotation on the radiographic appearance of the femoral canal,” *The Journal of Arthroplasty*, vol. 9, no. 4, pp. 419–426, Aug. 1994, doi: 10.1016/0883-5403(94)90053-1.

- [12] A. Rajpura, S. G. Asle, T. Ait Si Selmi, and T. Board, “The accuracy of restoration of femoral head centre of rotation in the anteroposterior plane after uncemented total hip arthroplasty: a CT-based study,” *Bone & Joint Research*, vol. 11, no. 3, pp. 180–188, Mar. 2022, doi: 10.1302/2046-3758.113.BJR-2021-0378.R2.
- [13] B. D. Springer, C. D. Etkin, P. B. Shores, T. J. Gioe, D. G. Lewallen, and K. J. Bozic, “Perioperative Periprosthetic Femur Fractures are Strongly Correlated With Fixation Method: an Analysis From the American Joint Replacement Registry,” *The Journal of Arthroplasty*, vol. 34, no. 7, pp. S352–S354, Jul. 2019, doi: 10.1016/j.arth.2019.02.004.
- [14] G. Kelmer, A. H. Stone, J. Turcotte, and P. J. King, “Reasons for Revision: Primary Total Hip Arthroplasty Mechanisms of Failure,” *JAAOS - Journal of the American Academy of Orthopaedic Surgeons*, vol. 29, no. 2, p. 78, Jan. 2021, doi: 10.5435/JAAOS-D-19-00860.
- [15] K. J. Bozic, S. M. Kurtz, E. Lau, K. Ong, T. P. Vail, and D. J. Berry, “The Epidemiology of Revision Total Hip Arthroplasty in the United States;,” *The Journal of Bone and Joint Surgery-American Volume*, vol. 91, no. 1, pp. 128–133, Jan. 2009, doi: 10.2106/JBJS.H.00155.
- [16] A. S. Dickinson, A. C. Taylor, J. R. T. Jeffers, and M. Browne, “Performance of the resurfaced hip. Part 2: The influence of prosthesis stem design on remodelling and fracture of the femoral neck,” *Proc Inst Mech Eng H*, vol. 224, no. 7, pp. 841–851, Jul. 2010, doi: 10.1243/09544119JEIM680.
- [17] T. Shishido, T. Tateiwa, Y. Takahashi, T. Masaoka, T. Ishida, and K. Yamamoto, “Effect of stem alignment on long-term outcomes of total hip arthroplasty with cementless Bi-Metric femoral components,” *Journal of Orthopaedics*, vol. 15, no. 1, pp. 134–137, Mar. 2018, doi: 10.1016/j.jor.2018.01.008.
- [18] R. M. A. Al-Dirini, D. O’Rourke, D. Huff, S. Martelli, and M. Taylor, “Biomechanical Robustness of a Contemporary Cementless Stem to Surgical Variation in Stem Size and Position,” *Journal of Biomechanical Engineering*, vol. 140, no. 9, p. 091007, Sep. 2018, doi: 10.1115/1.4039824.
- [19] S. P. Mirghaderi *et al.*, “Determining the accuracy of preoperative total hip replacement 2D templating using the mediCAD® software,” *J Orthop Surg Res*, vol. 17, no. 1, p. 222, Dec. 2022, doi: 10.1186/s13018-022-03086-5.
- [20] F. Bachour, E. Marchetti, D. Bocquet, L. Vasseur, H. Migaud, and J. Girard, “Radiographic preoperative templating of extra-offset cemented THA implants: How reliable is it and how does it affect survival?,” *Orthopaedics & Traumatology: Surgery & Research*, vol. 96, no. 7, pp. 760–768, Nov. 2010, doi: 10.1016/j.otsr.2010.05.004.
- [21] E. Sariali, A. Mouttet, G. Pasquier, E. Durante, and Y. Catone, “Accuracy of reconstruction of the hip using computerised three-dimensional pre-operative planning and a cementless modular neck,” *THE JOURNAL OF BONE AND JOINT SURGERY*, vol. 91, no. 3, 2009.
- [22] E. Sariali, R. Mauprivez, F. Khiami, H. Pascal-Mousselard, and Y. Catonné, “Accuracy of the preoperative planning for cementless total hip arthroplasty. A randomised comparison between three-dimensional computerised planning and

- conventional templating,” *Orthopaedics & Traumatology: Surgery & Research*, vol. 98, no. 2, pp. 151–158, Apr. 2012, doi: 10.1016/j.otsr.2011.09.023.
- [23] S. Sabharwal and A. Kumar, “Methods for Assessing Leg Length Discrepancy,” *Clinical Orthopaedics & Related Research*, vol. 466, no. 12, pp. 2910–2922, Dec. 2008, doi: 10.1007/s11999-008-0524-9.
- [24] K. A. Cassidy, M. S. Noticewala, W. Macaulay, J. H. Lee, and J. A. Geller, “Effect of Femoral Offset on Pain and Function After Total Hip Arthroplasty,” *The Journal of Arthroplasty*, vol. 27, no. 10, pp. 1863–1869, Dec. 2012, doi: 10.1016/j.arth.2012.05.001.
- [25] M. Worlicek, B. Messmer, J. Grifka, T. Renkawitz, and M. Weber, “Restoration of leg length and offset correlates with trochanteric pain syndrome in total hip arthroplasty,” *Sci Rep*, vol. 10, no. 1, p. 7107, Apr. 2020, doi: 10.1038/s41598-020-62531-9.
- [26] K. R. Cidambi, S. L. Barnett, P. R. Mallette, J. J. Patel, N. A. Nassif, and R. S. Gorab, “Impact of Femoral Stem Design on Failure After Anterior Approach Total Hip Arthroplasty,” *The Journal of Arthroplasty*, vol. 33, no. 3, pp. 800–804, Mar. 2018, doi: 10.1016/j.arth.2017.10.023.
- [27] B.-W. Min, K.-S. Song, K.-C. Bae, C.-H. Cho, C.-H. Kang, and S.-Y. Kim, “The Effect of Stem Alignment on Results of Total Hip Arthroplasty with a Cementless Tapered-Wedge Femoral Component,” *The Journal of Arthroplasty*, vol. 23, no. 3, pp. 418–423, Apr. 2008, doi: 10.1016/j.arth.2007.04.002.
- [28] T. Amenabar, K. Marimuthu, G. Hawdon, A. Gildone, and S. McMahon, “Total Hip Arthroplasty Using a Short-Stem Prosthesis: Restoration of Hip Anatomy,” *J Orthop Surg (Hong Kong)*, vol. 23, no. 1, pp. 90–94, Apr. 2015, doi: 10.1177/230949901502300121.
- [29] S. S. Mahmood, S. S. Mukka, S. Crnalic, P. Wretenberg, and A. S. Sayed-Noor, “Association between changes in global femoral offset after total hip arthroplasty and function, quality of life, and abductor muscle strength: A prospective cohort study of 222 patients,” *Acta Orthopaedica*, vol. 87, no. 1, pp. 36–41, Jan. 2016, doi: 10.3109/17453674.2015.1091955.
- [30] S. van Drongelen, “Are changes in radiological leg alignment and femoral parameters after total hip replacement responsible for joint loading during gait?,” *BMC Musculoskeletal Disorders*, vol. 20, no. 1, doi: 10.1186/s12891-019-2832-5.
- [31] T. Renkawitz *et al.*, “Leg length and offset differences above 5 mm after total hip arthroplasty are associated with altered gait kinematics,” *Gait & Posture*, vol. 49, pp. 196–201, Sep. 2016, doi: 10.1016/j.gaitpost.2016.07.011.
- [32] M. J. Isaacson, K. J. Bunn, and S. J. Incavo, “Trochanteric impingement: is it a source of pain after THR?,” *Arthroplasty Today*, vol. 1, no. 3, pp. 73–75, Sep. 2015, doi: 10.1016/j.artd.2015.05.004.
- [33] P. J. Brooks, “Dislocation following total hip replacement,” vol. 95, no. 11, 2013.
- [34] M. Brenneis *et al.*, “Accuracy of Preoperative Templating in Total Hip Arthroplasty With Special Focus on Stem Morphology: A Randomized Comparison Between Common Digital and Three-Dimensional Planning Using Biplanar Radiographs,”

- The Journal of Arthroplasty*, vol. 36, no. 3, pp. 1149–1155, Mar. 2021, doi: 10.1016/j.arth.2020.10.016.
- [35] Y. Knafo, F. Houfani, B. Zaharia, F. Egrise, I. Clerc-Urmès, and D. Mainard, “Value of 3D Preoperative Planning for Primary Total Hip Arthroplasty Based on Biplanar Weightbearing Radiographs,” *BioMed Research International*, vol. 2019, pp. 1–7, Mar. 2019, doi: 10.1155/2019/1932191.
- [36] M. Müller, D. Crucius, C. Perka, and S. Tohtz, “The association between the sagittal femoral stem alignment and the resulting femoral head centre in total hip arthroplasty,” *International Orthopaedics (SICOT)*, vol. 35, no. 7, pp. 981–987, Jul. 2011, doi: 10.1007/s00264-010-1047-z.
- [37] J. Xu, J. Pierrepont, C. Madurawe, A. Shimmin, and W. Bruce, “The effect of varus stem placement on joint offset during total hip arthroplasty: a virtual study,” *HIP International*, vol. 32, no. 5, pp. 620–626, Sep. 2022, doi: 10.1177/1120700020978368.
- [38] R. B. Martin, D. B. Burr, N. A. Sharkey, and D. P. Fyhrie, *Skeletal Tissue Mechanics*. New York, NY: Springer New York, 2015. doi: 10.1007/978-1-4939-3002-9.
- [39] M. Ait Mokhtar, “Postero-posterolateral approach in total hip arthroplasty,” *International Orthopaedics (SICOT)*, vol. 44, no. 12, pp. 2577–2585, Dec. 2020, doi: 10.1007/s00264-020-04679-7.
- [40] E. Anyanwu, J. Ekezie, E. Obikili, and N. Onyemaechi, “Anatomical basis for surgical approaches to the hip,” *Ann Med Health Sci Res*, vol. 4, no. 4, p. 487, 2014, doi: 10.4103/2141-9248.139278.
- [41] D. A. Neumann, “Kinesiology of the Hip: A Focus on Muscular Actions,” 2010.
- [42] D. Kim, J. Unger, J. L. Lanovaz, and A. R. Oates, “The Relationship of Anticipatory Gluteus Medius Activity to Pelvic and Knee Stability in the Transition to Single-Leg Stance,” *PM&R*, vol. 8, no. 2, pp. 138–144, Feb. 2016, doi: 10.1016/j.pmrj.2015.06.005.
- [43] C. Chang, S. H. Jen, and M. Varacallo, “Anatomy, Bony Pelvis and Lower Limb, Piriformis Muscle”.
- [44] M. J. Philippon *et al.*, “Surgically Relevant Bony and Soft Tissue Anatomy of the Proximal Femur,” *Orthopaedic Journal of Sports Medicine*, vol. 2, no. 6, p. 232596711453518, Jun. 2014, doi: 10.1177/2325967114535188.
- [45] R. S. Garden, “THE STRUCTURE AND FUNCTION OF THE PROXIMAL END OF THE FEMUR,” *The Journal of Bone and Joint Surgery. British volume*, vol. 43-B, no. 3, pp. 576–589, Aug. 1961, doi: 10.1302/0301-620X.43B3.576.
- [46] Q. Zhang *et al.*, “The role of the calcar femorale in stress distribution in the proximal femur,” *Orthopaedic Surgery*, vol. 1, no. 4, pp. 311–316, Nov. 2009, doi: 10.1111/j.1757-7861.2009.00053.x.
- [47] O. for E. Co-operation and Development, *OECD Health Data...* Organisation for Economic Co-operation and Development, 2006.
- [48] A. J. Cassar-Gheiti, R. McColgan, M. Kelly, T. M. Cassar-Gheiti, P. Kenny, and C. G. Murphy, “Current concepts and outcomes in cemented femoral stem design and cementation techniques: the argument for a new classification system,” *EFORT*

- Open Reviews*, vol. 5, no. 4, pp. 241–252, Apr. 2020, doi: 10.1302/2058-5241.5.190034.
- [49] H. S. Khanuja, J. J. Vakil, M. S. Goddard, and M. A. Mont, “Cementless Femoral Fixation in Total Hip Arthroplasty,” *Journal of Bone and Joint Surgery*, vol. 93, no. 5, pp. 500–509, Mar. 2011, doi: 10.2106/JBJS.J.00774.
- [50] F. J. Dorey, “Survivorship Analysis of Surgical Treatment of the Hip in Young Patients:,” *Clinical Orthopaedics and Related Research*, vol. 418, pp. 23–28, Jan. 2004, doi: 10.1097/00003086-200401000-00005.
- [51] H. Ibrahim, S. N. Esfahani, B. Poorganji, D. Dean, and M. Elahinia, “Resorbable bone fixation alloys, forming, and post-fabrication treatments,” *Materials Science and Engineering: C*, vol. 70, pp. 870–888, Jan. 2017, doi: 10.1016/j.msec.2016.09.069.
- [52] R. B. Ruben, P. R. Fernandes, and J. Folgado, “On the optimal shape of hip implants,” *Journal of Biomechanics*, vol. 45, no. 2, pp. 239–246, Jan. 2012, doi: 10.1016/j.jbiomech.2011.10.038.
- [53] R. Narayanan, A. M. Elbuluk, K. K. Chen, N. Eftekhary, J. D. Zuckerman, and A. J. Deshmukh, “Does femoral morphology and stem alignment influence outcomes of cementless total hip arthroplasty with proximally coated double-tapered titanium stems?,” *HIP International*, vol. 31, no. 3, pp. 354–361, May 2021, doi: 10.1177/1120700019891702.
- [54] F. Stief, S. Van Drongelen, M. Brenneis, T. Tarhan, B. Fey, and A. Meurer, “Influence of Hip Geometry Reconstruction on Frontal Plane Hip and Knee Joint Moments During Walking Following Primary Total Hip Replacement,” *The Journal of Arthroplasty*, vol. 34, no. 12, pp. 3106–3113, Dec. 2019, doi: 10.1016/j.arth.2019.07.027.
- [55] E. Sariali, S. Klouche, A. Mouttet, and H. Pascal-Moussellard, “The effect of femoral offset modification on gait after total hip arthroplasty,” *Acta Orthopaedica*, vol. 85, no. 2, pp. 123–127, Apr. 2014, doi: 10.3109/17453674.2014.889980.
- [56] T. Wuestemann *et al.*, “Bone morphology of the proximal femoral canal: ethnicity related differences and the influence on cementless tapered wedge stem designs,” *HIP International*, vol. 31, no. 4, pp. 482–491, Jul. 2021, doi: 10.1177/1120700019895458.
- [57] P. Panichkul, S. Bavonratanavech, A. Arirachakaran, and J. Kongtharvonskul, “Comparative outcomes between collared versus collarless and short versus long stem of direct anterior approach total hip arthroplasty: a systematic review and indirect meta-analysis,” *Eur J Orthop Surg Traumatol*, vol. 29, no. 8, pp. 1693–1704, Dec. 2019, doi: 10.1007/s00590-019-02516-1.
- [58] M. R. Streit *et al.*, “Early Migration Predicts Aseptic Loosening of Cementless Femoral Stems: A Long-term Study,” *Clinical Orthopaedics & Related Research*, vol. 474, no. 7, pp. 1697–1706, Jul. 2016, doi: 10.1007/s11999-016-4857-5.
- [59] L. D. Dorr, M.-C. Faugere, A. M. Mackel, T. A. Gruen, B. Bognar, and H. H. Malluche, “Structural and cellular assessment of bone quality of proximal femur,” *Bone*, vol. 14, no. 3, pp. 231–242, May 1993, doi: 10.1016/8756-3282(93)90146-2.

- [60] J. Wilkerson and N. D. Fernando, “Classifications in Brief: The Dorr Classification of Femoral Bone,” *Clin Orthop Relat Res*, vol. 478, no. 8, pp. 1939–1944, Aug. 2020, doi: 10.1097/CORR.0000000000001295.
- [61] K. Watanabe, K. Mitsui, Y. Usuda, and K. Nemoto, “An increase in the risk of excessive femoral anteversion for relatively younger age and types of femoral morphology in total hip arthroplasty with direct anterior approach,” *J Orthop Surg (Hong Kong)*, vol. 27, no. 2, p. 230949901983681, May 2019, doi: 10.1177/2309499019836816.
- [62] B. The, N. Verdonshot, J. R. van Horn, P. M. A. van Ooijen, and R. L. Diercks, “Digital Versus Analogue Preoperative Planning of Total Hip Arthroplasties,” *The Journal of Arthroplasty*, vol. 22, no. 6, pp. 866–870, Sep. 2007, doi: 10.1016/j.arth.2006.07.013.
- [63] E. Schiffner *et al.*, “Is computerised 3D templating more accurate than 2D templating to predict size of components in primary total hip arthroplasty?,” *HIP International*, vol. 29, no. 3, pp. 270–275, May 2019, doi: 10.1177/1120700018776311.
- [64] I. Asayama, S. Chamnongkich, K. J. Simpson, T. L. Kinsey, and O. M. Mahoney, “Reconstructed Hip Joint Position and Abductor Muscle Strength After Total Hip Arthroplasty,” *The Journal of Arthroplasty*, vol. 20, no. 4, pp. 414–420, Jun. 2005, doi: 10.1016/j.arth.2004.01.016.
- [65] D. Mainard, O. Barbier, Y. Knafo, R. Belleville, L. Mainard-Simard, and J.-B. Gross, “Accuracy and reproducibility of preoperative three-dimensional planning for total hip arthroplasty using biplanar low-dose radiographs : A pilot study,” *Orthopaedics & Traumatology: Surgery & Research*, vol. 103, no. 4, pp. 531–536, Jun. 2017, doi: 10.1016/j.otsr.2017.03.001.
- [66] M. A. Belzunce, J. Henckel, A. Di Laura, and A. Hart, “Uncemented femoral stem orientation and position in total hip arthroplasty: A CT study,” *J Orthop Res*, vol. 38, no. 7, pp. 1486–1496, Jul. 2020, doi: 10.1002/jor.24627.
- [67] S. H. Lee, Y. J. Cho, and W. K. Choi, “Comparison of femoral sagittal axis between navigated total knee arthroplasty and conventional total knee arthroplasty in patients with knee osteoarthritis,” *Medicine*, vol. 101, no. 33, p. e30085, Aug. 2022, doi: 10.1097/MD.00000000000030085.
- [68] T. Andreassen, “Fast_Point2TriMesh,” *Fast_Point2TriMesh*, Jul. 14, 2023. https://www.mathworks.com/matlabcentral/fileexchange/129329-fast_point2trimesh (accessed Jul. 14, 2023).
- [69] E. S. Groot and W. J. Suntay, “A Joint Coordinate System for the Clinical Description of Three-Dimensional Motions: Application to the Knee,” *Journal of Biomechanical Engineering*, vol. 105, no. 2, pp. 136–144, May 1983, doi: 10.1115/1.3138397.
- [70] J. R. Landis and G. G. Koch, “The Measurement of Observer Agreement for Categorical Data,” *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977.
- [71] H. Hassani, S. Cherix, E. T. Ek, and H. A. Rüdiger, “Comparisons of Preoperative Three-Dimensional Planning and Surgical Reconstruction in Primary Cementless

- Total Hip Arthroplasty,” *The Journal of Arthroplasty*, vol. 29, no. 6, pp. 1273–1277, Jun. 2014, doi: 10.1016/j.arth.2013.12.033.
- [72] W. Fugit, “Revision Total Hip Femoral Stem Micromotion and Statistical Shape Modeling of the Knee,” *Electronic Theses and Dissertations*, Jan. 2022, [Online]. Available: <https://digitalcommons.du.edu/etd/2050>
- [73] T. Gruen and G. McNeice, “and| Amstutz, HC: ‘Modes of failure’ of cemented stem-type femoral components. A radiographic analysis of loosening,” *Clin. Orthop*, vol. 141, no. 17, pp. 1–1, 1979.
- [74] J. Shi, M. Browne, M. Strickland, G. Flivik, and M. Taylor, “Sensitivity analysis of a cemented hip stem to implant position and cement mantle thickness,” *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 17, no. 15, pp. 1671–1684, Nov. 2014, doi: 10.1080/10255842.2012.761693.

Appendix A: Segmentation Algorithm

The code presented in this appendix goes through the segmentation algorithm one function at a time in the order that they are run in the code. The appended code highlights what we believe are the important functions that pertain to the automatic segmentation code and all other functions have been removed for simplicity and reproducibility. The flow of the code first starts by loading the DICOM images and correcting any images/ converting to 16 unsigned bits. Then the ICC segmentation algorithm is ran (DU_GetCalcar_Final). The ICDM segmentation algorithm is then ran (DU_GetIntercorticalMesh_Final). Subsequently, the anatomic measurements were obtained using the DU_GetDorr_Final and DU_GetBowling_Final functions.

DU_GetCalcar_Final

```
function [calcar_output, TM_GtoS, T1] =  
DU_GetCalcar_Final(SUB, dcm, dcm_slices_input, bn_input, calcar_threshold_percents_in  
ut, cb_input, varargin)
```

DU_GetCalcar_Final by Sam Mattei 2/14/2023

```
%This code gathers the ICC nodes for virtual implanataion based on a user  
%requested threshold. More information regarding the inputs and outputs is  
%shown below.  
  
%INPUTS:  
  
%SUB - SUBJECT Files for each specimen loaded from DU_AnalyzeHipFemurV2  
  
%dcm - structure containing all DICOM images  
  
%dcm_slices_input - structure containing the DICOM slices for each Right  
%and Left Legs  
  
%bn_input - name of Right and left bones for further indexing
```

```

%cb_input - RL identifier 1- Left, 2- Right

%calcar_threshold_percents_input - input threshold value to be used,
%specified by the user

%OUTPUTS:
%calcar_output - Output calcar mesh transformed to the global space of the
%CT scan

%TM_GtoS - transformation from Global CT space to Sulcus Coordinate System

%T1 - table containing segmentation information

```

Top Level Code Overview

```

%STEP 1: Assigning Optional Parameters
%STEP 2: Gathering All Anatomic Information and Upsampling External Nodes
%STEP 3: ICC Node Segmentation
%STEP 4: ICC Node Smoothing

```

STEP 1: Varargin Parameters

```

numPts_input=400;
x_smoothed_points_input=75;
y_smoothed_points_input=75;
ccpParameters_input=[0 0];
save_img=0;
img_path='R:\Research Common\EBL\Anthropometrics\Hip_Implantations\Results\Calcar
Updates (1.3.2023)';
% Loading optional arguments
while ~isempty(varargin)
    switch lower(varargin{1})
        case 'numptsinterpolate'
            numPts_input = varargin{2};
        case 'xsmoothedpoints'
            x_smoothed_points_input = varargin{2};
        case 'ysmoothedpoints'
            y_smoothed_points_input = varargin{2};
        case 'plottoggle'
            plot_var = varargin{2};
        case 'savetoggle'
            save_img=varargin{2};
        case 'imagepath'
            img_path=varargin{2};
        case 'ccpparameters'
            ccpParameters_input=varargin{2};
    end
end

```



```

        otherwise
            error(['Unexpected option: ' varargin{1}])
        end
        varargin(1:2) = [];
    end
end

```

STEP 2: Gathering All Anatomic Information and Upsampling External

Nodes/ Optional Changing of Calcar Cut Plane

```

%storing the pixel spacing to the patient in a vector for
%transformation of coordinates
ps=[dcm.info{1}.PixelSpacing(1),dcm.info{1}.PixelSpacing(2)];
%Position to patient for coordinate transformation
pos=[dcm.info{1}.ImagePositionPatient(1),dcm.info{1}.ImagePositionPatient(2)];

%pulling the calcar nodes and transforming
if cb_input==1 %LFEMUR
    if any(ccpParamaters_input~=0)

[calcarNodes_G, TM_GtoS, outputResectedElemsList]=DU_ChangeCalcarCutPlane(SUB,cb_inp
ut, ccpParamaters_input);
        calcarNodes_S= (TM_GtoS*calcarNodes_G)';
    else
        TM_GtoS=SUB.TM.F.GtoSL;
        calcarNodes_A=SUB.GEO.A.REGIONS.LFemur.CALCAR_RESECTION;
        calcarNodes_G= (SUB.TM.F.GtoAPL*calcarNodes_A)'; %transforming to the
global space
        calcarNodes_S=(SUB.TM.F.GtoSL*calcarNodes_G)'; %transforming to the
sulcus space
    end
elseif cb_input==2 %RFEMUR

    if any(ccpParamaters_input~=0)

[calcarNodes_G, TM_GtoS, outputResectedElemsList]=DU_ChangeCalcarCutPlane(SUB,cb_inp
ut, ccpParamaters_input);
        calcarNodes_S= (TM_GtoS*calcarNodes_G)';
    else
        TM_GtoS=SUB.TM.F.GtoSR;
        calcarNodes_A=SUB.GEO.A.REGIONS.RFemur.CALCAR_RESECTION;
        calcarNodes_G= (SUB.TM.F.GtoAPR*calcarNodes_A)'; %transforming to the
global space
        calcarNodes_S=(SUB.TM.F.GtoSR*calcarNodes_G)';
    end
end

%interpolating the boundary of the sulcus nodes

```

```

slice_nodes_spline=ones(numPts_input,4);
%initializing the linear length vector
bndry_linear_length=0;
%Creating the Vector of spline points
x_pts=linspace(0,1,numPts_input);
y_pts=linspace(0,1,numPts_input);

% Finding the boundary of the given nodes to then be used
[indices_bndry]=boundary(calcarNodes_S(:,2),calcarNodes_S(:,3),0.35);

count_GCNA=1;
% Calculating the linear distance that each of the points ends up contributing to
for count_bndry=1:length(indices_bndry)
    %pulling the points for each boundary
    bndry=indices_bndry(count_bndry);
    %Logic if indices are on the end of the list
    if bndry == length(calcarNodes_S(:,1))
        pt1=calcarNodes_S(bndry,2:3);
        pt2=calcarNodes_S(1,2:3);
    else
        %pulling each of the nodal points
        pt1=calcarNodes_S(bndry,2:3);
        pt2=calcarNodes_S(bndry+1,2:3);
    end
    norm_bndry(count_GCNA)=norm(pt2-pt1);
    %finding the magnitude of the distance between each point and
    %summing those values
    bndry_linear_length=bndry_linear_length + norm_bndry(count_GCNA);
    %updating the count_GCNA
    count_GCNA=count_GCNA+1;
end

%calculating the normalized lengths for each point and organizing them
%using a cumulative summation
temp_norm_vector = cumsum(norm_bndry/bndry_linear_length)';

%using the cubic smothing spline to determine the smoothed curve
x_csaps=csaps(temp_norm_vector,calcarNodes_S(indices_bndry,2), 1.0);
y_csaps=csaps(temp_norm_vector,calcarNodes_S(indices_bndry,3), 1.0);

%evaluating the function at each point
x_spline_nodes=fnval(x_csaps,x_pts);
y_spline_nodes=fnval(y_csaps,y_pts);

%storing in the temp structure
slice_nodes_spline(:,2:3)=[x_spline_nodes;y_spline_nodes]';
slice_nodes_spline(:,4)=0; % 0 because the nodes are assumed to be planar

%Using the convex Hull to create the bounding polyshape to then be
%partitioned

```

```

conv_hull_ind=convhull(calcarNodes_S(:,2),calcarNodes_S(:,3));
temp_ps=polyshape(calcarNodes_S(conv_hull_ind,2),calcarNodes_S(conv_hull_ind,3));
[temp_xc, temp_yc]=centroid(temp_ps);
deg=0:360/numPts_input:360; %Creating the vector that goes by the rotational
increment to 360 degrees
v_length=100; %setting the vector length to be 100 pixels
count_GCNA=1;

for count_deg=1:length(deg)-1
    deg_inc=deg(count_deg); %extracting on rotational degree increment
    %Setting up the changes in the angles as a function of degrees
    sin=sind(deg_inc);cos=cosd(deg_inc);
    %updating the endpoints of the line
    temp_line_seg=[temp_xc , temp_yc; temp_xc + cos*v_length , temp_yc +
sin*v_length];
    %finding which portion of the line segment intersects with the
    %cortical bounding box

    [in_x,in_y]=polyxpoly(calcarNodes_S(conv_hull_ind,2),calcarNodes_S(conv_hull_ind,3
),temp_line_seg(:,1),temp_line_seg(:,2));
    calcarNodes_SInterpolated(count_GCNA,:)= [1,in_x,in_y,0]; %z should be 0 for
all nodes since they are planar
    count_GCNA=count_GCNA+1;
end
%transforming the interpolated nodes back to the global space
calcarNodes_GInterpolated=(inv(TM_GtoS)*calcarNodes_SInterpolated)';

%finding the centroid of the sulcus nodes in sulcus CS
centerInterpolatedCalcar_S=mean(calcarNodes_SInterpolated(:,:),1); %taking the
mean along the columns for x,y,z mean

%translating back to global
centerInterpolatedCalcar_G=(inv(TM_GtoS)*centerInterpolatedCalcar_S)';

%transforming those nodes to the Dicom space
calcarNodes_DCM=calcarNodes_GInterpolated;
%transforming the x and y coordinates to DICOM space
calcarNodes_DCM(:,2)=(calcarNodes_GInterpolated(:,2)- pos(1))/ps(1) + 1;
calcarNodes_DCM(:,3)=(calcarNodes_GInterpolated(:,3)- pos(2))/ps(2) + 1;

%Optional Visual Code
%getting the minimum and maximum indices
ind_viz=[min(calcarNodes_GInterpolated(:,4)) ,
max(calcarNodes_GInterpolated(:,4))];
%extracting the minimum and maximum values
[~,idmax]=min( abs(dcm_slices_input(:,2)- ind_viz(1)));
[~,idmin]=min( abs(dcm_slices_input(:,2)- ind_viz(2)));

if idmin>idmax
    temp=idmax;

```

```

    idmax=idmin;
    idmin=temp;
end

%pulling the dicom stack over that range and finding the maximum value over
%that range
dcm_stack_calcar=dcm.stack(:,:,idmin:idmax);
dcm_stack_calcar=double(dcm_stack_calcar);
maxValue=max(max(max(dcm_stack_calcar)));
%initializing the colormap
cmap=bone(maxValue);

```

STEP 3: ICC Segmentation

```

calcarDecimatedTempHUVector=ones(numPts_input,size(calcarNodes_DCM,1)); %this
structure will store all the information about the pixel locations between the
mean and the pixel of interest as well as the HU or BMD pixel data for each
calcarDecimatedTempCMAP=calcarDecimatedTempHUVector; %initializing the color map
for visualization

```

```

calcarAlgorithmNodes=ones(numPts_input,size(calcarNodes_DCM,2));
%looping through each of the sulcus nodes to determine the coordinate
%points
for countSulcusNodes=1:size(calcarNodes_GInterpolated,1)

```

```

%Using the sign of each value to determine the value that will be
%used to create the vectors

```

```

ind1X=centerInterpolatedCalcar_G(:,2);
ind2X=calcarNodes_GInterpolated(countSulcusNodes,2);

```

```

ind1Y=centerInterpolatedCalcar_G(:,3);
ind2Y=calcarNodes_GInterpolated(countSulcusNodes,3);

```

```

ind1Z=centerInterpolatedCalcar_G(:,4);
ind2Z=calcarNodes_GInterpolated(countSulcusNodes,4);

```

```

%creating a temporary vector that goes from the node of interest to the
%centroid coordinates

```

```

tempVecX_G=linspace(ind1X,ind2X,numPts_input);
tempVecY_G=linspace(ind1Y,ind2Y,numPts_input);
tempVecZ_G=linspace(ind1Z,ind2Z,numPts_input);
tempVec_P=linspace(0,1,numPts_input);

```

```

%storing

```

```

node_plot{countSulcusNodes}.nds=[ones(length(tempVecX_G),1), tempVecX_G',
tempVecY_G', tempVecZ_G' ];

```

```

%storing in a temporary matrix then transforming back to global space
%and dicom space afterwards

```

```

tempVec_G=ones(numPts_input,4);

```

```

tempVec_G(:,2:4)=[tempVecX_G;tempVecY_G;tempVecZ_G]';

%transforming to the sulcus coordinate system
tempVec_S=(inv(TM_GtoS)*tempVec_G)';

%Transforming to dicom space
tempVec_DCM=tempVec_G;
%transforming the x and y coordinates to DICOM space
tempVec_DCM(:,2)=(tempVec_G(:,2)- pos(1))/ps(1) + 1;
tempVec_DCM(:,3)=(tempVec_G(:,3)- pos(2))/ps(2) + 1;

%looping through each point and snagging the appropriate value from the
%Dicom
for countLinspace=1:numPts_input
    % pulling the list of minimum and maximum nodal coordinates to pull the
    % appropriate dicom slices
    [~,idx1]=min( abs(dcm_slices_input(:,2)- tempVec_DCM(countLinspace,4)));
    %extracting the current dicom slice in mm and in the other place
    tempZ=dcm_slices_input(idx1,1);

    %pulling and storing the pixel value at each of the equally spaced
    %nodes

calcarDecimatedTempHUVector(countLinspace,countSulcusNodes)=dcm.stack(fix(tempVec_
DCM(countLinspace,3)),fix(tempVec_DCM(countLinspace,2)),tempZ);

    temp=calcarDecimatedTempHUVector(countLinspace,countSulcusNodes);

    if temp<0
        temp=0; %rounding down to 0 if the pixel value is below 0 since scans
are unsigned
    end
    calcarDecimatedTempCMAP(countLinspace,countSulcusNodes)=temp/maxValue;
end

%this is the DCM profile line that has all the information on it for a
%given node
dcmProfileLineFull=calcarDecimatedTempHUVector(:,countSulcusNodes);

%pulling the maximum value on the full line and cropping each vector to
%find the cropped minimum value
[maxPTRange,maxIND]=max(dcmProfileLineFull);
temp_DCMLINE_CROPPED=dcmProfileLineFull(1:maxIND);

%Solving for the minimum value on the max value cropped DCM line
[minPTRange,minIND]=min(temp_DCMLINE_CROPPED);

%cropping plotting vectors
dcm_profile_line_cropped=dcmProfileLineFull(minIND:maxIND);
tempVec_P_cropped=tempVec_P(minIND:maxIND);

```

```

%Identifying points of intersection
target_height=((maxPTRange-minPTRange)*calcar_threshold_percents_input);
horizontal_line_height1= target_height + minPTRange;
pt1_line_seg=[min(tempVec_P) max(tempVec_P);horizontal_line_height1
horizontal_line_height1]';

[pt1_ind,~]=polyxpoly(pt1_line_seg(:,1),pt1_line_seg(:,2),tempVec_P_cropped,dcm_pr
ofile_line_cropped);
y_pt1=ones(length(pt1_ind),1)*horizontal_line_height1;

v=[ind2X,ind2Y,ind2Z]-[ind1X,ind1Y,ind1Z]; %calculating the overall vector
%finding the unit vector
uv=(v)/norm(v);

%finding x and y pixel values
for countIdentifiedPts=1:length(pt1_ind)
    %scaling the distance of the line based on the percentage
    scaled_norm=pt1_ind(countIdentifiedPts)*norm(v);
    pts=uv*scaled_norm;
    %storing

indices_of_pts(countIdentifiedPts,:)= [pt1_ind(countIdentifiedPts),centerInterpolat
edCalcar_G(:,2)+pts(1),centerInterpolatedCalcar_G(:,3)+pts(2),centerInterpolatedCa
lcar_G(:,4)+pts(3) ];
end

%storing each node in the dicom space
temp_output=ones(length(pt1_ind),4);
temp_output(:,2)=temp_output(:,2).*indices_of_pts(:,2);
temp_output(:,3)=temp_output(:,3).*indices_of_pts(:,3);
%returning to global space
temp_output(:,4)=temp_output(:,4).*indices_of_pts(:,4); %storing as the global
indices

%Storing Minimum Distance
distanceNodeToPoint= sqrt(sum(abs(temp_output(:,2:4)-
calcarNodes_GInterpolated(countSulcusNodes,2:4)).^2,2));
[~,ind_to_keep]=min(distanceNodeToPoint);
node_output=temp_output(ind_to_keep,1:4);
% Storing Raw Return variables
calcarAlgorithmNodes(countSulcusNodes,:)=node_output;

clear temp_output indices_of_pts

end

```

STEP 4: Smoothing the Selected Nodess

```

%transforming the global nodes to the sulcus space and performing some
>windowing and smoothing to those nodes
tempNodesSulcus=(TM_GtoS*calcarAlgorithmNodes)';

%interpolating
ind_bndry=unique(boundary(tempNodesSulcus(:,2),tempNodesSulcus(:,3),0.5),'stable')
;
vecToInterpolate=linspace(0,1,x_smoothed_points_input);
pts=interparc(vecToInterpolate,[tempNodesSulcus(ind_bndry,2);tempNodesSulcus(ind_b
ndry(1),2)],[tempNodesSulcus(ind_bndry,3);tempNodesSulcus(ind_bndry(1),3)],'linear
');

%storing in the temp structure
tempInterpolatedNodesSulcus=[ones(size(pts,1),1), pts , zeros(size(pts,1),1)];

%storing and transforming the final output
calcar_output=(inv(TM_GtoS)*tempInterpolatedNodesSulcus)'; %TM back to global
space

% Optional visuals
if plot_var==1
    %viz of the line and the tangent line
    DUGCF1=figure();set(gcf,'color',[1 1 1]);
    colorbar
    colormap bone
    light

    patch('Vertices',SUB.GEO.G.(bn_input{cb_input}).nds(:,2:4),'Faces',SUB.GEO.G.(bn_i
nput{cb_input}).faces(SUB.GEO.G.(bn_input{cb_input}).neck_element_list,2:4),'FaceC
olor',[227/256 218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.95,'Clipping','
off');hold on;

    plot3(calcarNodes_GInterpolated(:,2),calcarNodes_GInterpolated(:,3),calcarNodes_GI
nterpolated(:,4),'Marker','none','Color','k','LineStyle','-'); hold on

    plot3(calcarAlgorithmNodes(:,2),calcarAlgorithmNodes(:,3),calcarAlgorithmNodes(:,4
),'Marker','.','Color','r','LineStyle','none','MarkerSize',6)

    plot3(calcar_output(:,2),calcar_output(:,3),calcar_output(:,4),'Marker','o','Color
','c','LineStyle','none','MarkerSize',8)

    for countj=1:size(calcarNodes_GInterpolated,1)

        patch(node_plot{countj}.nds(:,2),node_plot{countj}.nds(:,3),node_plot{countj}.nds(
(:,4),calcarDecimatedTempCMAP(:,countj),'FaceColor','none','EdgeColor','interp','Li
newidth',4.0);
        end

```

```

plot3(centerInterpolatedCalcar_G(2),centerInterpolatedCalcar_G(3),centerInterpolatedCalcar_G(4), 'Marker','x','Color','c','LineStyle','none','MarkerSize',14)

    zlim([ (0.75*(max(SUB.GEO.G.((bn_input{cb_input})).nds(:,4))-
min(SUB.GEO.G.((bn_input{cb_input})).nds(:,4))) ) +
min(SUB.GEO.G.((bn_input{cb_input})).nds(:,4))
max(SUB.GEO.G.((bn_input{cb_input})).nds(:,4))])
    axis equal;axis off;light;
    title(['Calcar Segmentation: ' bn_input{cb_input} ' of ' SUB.ID
], 'FontSize',18)

    if save_img ==1
        %pathing for saving images
        fp=[img_path, '\', SUB.ID, '\', SUB.ID, '_', bn_input{cb_input},
'_CurrentWorkflowVFinal', '.fig'];
        saveas(DUGCF1, fp)
    end
end
% Storing Properties In A Table To Be Returned To The User
T1=table(string(SUB.ID()),string(bn_input{cb_input}),calcar_threshold_percents_input*100 ,x_smoothed_points_input,y_smoothed_points_input, 'VariableNames',{'Subject Identifier','Bone Identifier','Profile Line Threshold (%)','Number of X Intpr Pts','Number of Y Intpr Pts'});
disp(T1)
end

```

Helper Functions

```

%Cut Plane Optional Parameters Function
function [outputNodesG,outputTM_GtoSNew,outputResectedElemsList] =
DU_ChangeCalcarCutPlane(H,cb,newCutPlaneParameters)

% Explanation
%This code is designed to take in a mesh of the bone and information about
%the current sulcus coordiante system and augment it and shift it to a
%manually chosen offset and rotation if desired by the user. This code will
%return the newly selected nodes, updates transformation matrix and the
%resected elements list for plotting.

%H - Subject structure originates from the DU_AnalyzeHipFemurV2, contains
%the global and sulcus transformation matrices

%cb - Bone count (1-Left Femur || 2-Right Femur)

%newCutPlaneParameters - array with two integers, the first is the rotation
%of the cut plane in DEGREES (A positive integer means a steeper cut plane
%angle while a negative means a more shallow). The second is the offset

```



```

%along the femoral neck this is in MILLIMETERS (A positive integer
% corresponds to a translation up the femoral neck towards the femoral head
% and a negative integer means a translation away from the femoral head )

```

Top Level Code Overview

```

%Step 1: Identify the Global to Sulcus coordinate system transformation matrix
%Step 2: Create a sulcus to sulcus new transformation matrix
%Step 3: Identify nodes on the femoral neck
%Step 4: Store new CCP nodes by transforming back to the old CS

% Step 1: Identify the original coordinate transformation and initializing any
structures/ plotting variables
bn={'LFemur','RFemur'};
axisG=[1 0 0 0;...
       1 20 0 0;...
       1 0 0 0;...
       1 0 20 0;...
       1 0 0 0;...
       1 0 0 20;...
       1 0 0 0];
%creating a plane on the based about 0,0,0
planewidth=50;
plane_G=[ 1 planewidth planewidth 0;1 -planewidth planewidth 0;1 -planewidth -
planewidth 0;1 planewidth -planewidth 0 ];
plane_faces=[1 1 2 3 4];
% newCutPlaneParameters=[0 10];
plot_var=1;

%pulling the calcar nodes and transforming
if cb ==1 %LFEMUR
    TM_GtoS=H.TM.F.GtoSL;
    calcarNodes_A=H.GEO.A.REGIONS.LFemur.CALCAR_RESECTION;
    calcarNodes_G= (H.TM.F.GtoAPL*calcarNodes_A)'; %transforming to the global
space
    calcarNodes_S=(H.TM.F.GtoSL*calcarNodes_G)'; %transforming to the sulcus
space
elseif cb==2 %RFEMUR
    TM_GtoS=H.TM.F.GtoSR;
    calcarNodes_A=H.GEO.A.REGIONS.RFemur.CALCAR_RESECTION;
    calcarNodes_G= (H.TM.F.GtoAPR*calcarNodes_A)'; %transforming to the global
space
    calcarNodes_S=(H.TM.F.GtoSR*calcarNodes_G)'; %transforming to the sulcus
space
end

% Step 2: Create The New Calcar Resection Cut Based On The Input Parameters

```

```

%finding the center of the temp_bone and making that the center of the new
%calcar cut since it should be along the neck axis vector of the femoral
%head
temp_origin=mean(calcarNodes_S,1);
TM_StoSAligned=eye(4,4);
TM_StoSAligned(2:4,1)=-1.*temp_origin(2:4);

%offset neck cut transformation matrix
TM_SAlignedtoSOffset=eye(4,4);
TM_SAlignedtoSOffset(2:4,1)=TM_GtoS(4,2:4).*(-newCutPlaneParameters(2));
%establishing the offset

%rotation matrix
TM_SOffsettoSRotated=eye(4,4);
TM_SOffsettoSRotated(2:4,2:4)=[cosd(newCutPlaneParameters(1)) 0 -
sind(newCutPlaneParameters(1)); 0 1 0; sind(newCutPlaneParameters(1)) 0
cosd(newCutPlaneParameters(1))];

TM_GtoSNew=TM_SOffsettoSRotated*TM_SAlignedtoSOffset*TM_StoSAligned*TM_GtoS;

%preparing all the nodes for plotting
nodes_S=(TM_GtoS*H.GEO.G.(bn{cb}).nds)';
nodes_S_aligned=(TM_StoSAligned*nodes_S)';
nodes_S_offset=(TM_SAlignedtoSOffset*nodes_S_aligned)';
nodes_S_rotated=(TM_SOffsettoSRotated*nodes_S_offset)';

nodes_SNew=(TM_GtoSNew*H.GEO.G.(bn{cb}).nds)';

%Step 3: Identify nodes that intersect with the new cut plane
newCCPNodes_SNew=nodes_SNew(nodes_SNew(:,4)>=-0.3 & nodes_SNew(:,4)<=0.3,:);
newCCPNodes_SNew(:,4)=0; %setting the nodes to be planar as they will make up the
%boundary for the planar cut

%Transforming back to the global space
newCCPNodes_G=(inv(TM_GtoSNew)*newCCPNodes_SNew)';

%figuring out the new resected elements list
head_node_list = find(nodes_SNew(:,4)>0);
temp_element_list = [];
for i=1:length(head_node_list)
    temp_element_list = [temp_element_list;
    find(H.GEO.G.(bn{cb}).faces(:,2)==head_node_list(i));
    find(H.GEO.G.(bn{cb}).faces(:,3)==head_node_list(i));
    find(H.GEO.G.(bn{cb}).faces(:,4)==head_node_list(i))];
end
head_element_list = unique (temp_element_list);
neck_element_list = 1:length(H.GEO.G.(bn{cb}).faces);
neck_element_list(head_element_list)=[];

```

```
% Step 4: Store the data as an output
outputNodesG=newCCPNodes_G;
outputTM_GtoSNew=TM_GtoSNew;
outputResectedElemsList=neck_element_list;
```

```
end
```

DU_GetIntercorticalMesh_Final

```
function [nodeOutput,elemOutput] =
DU_GetIntercorticalMesh_Final(SUB,dcm,dcm_slices_input,RL,varargin)
```

DU_GetIntercorticalMesh_Final - By Sam Mattei 3/10/2023

```
%This purpose of this function is to return a finalized
%diaphyseal-metaphyseal mesh to the user a the heights specified using
%subject-specific DICOM images. This function will generate a series of raw
%nodes, smooth said nodes and create a triangular connctivity list for the
%surface mesh. This code has dependencies on DU_MeshDiaphysis_V4. Below is an
%explanation of the inputs and outputs
```

```
%Inputs:
```

```
%SUB - SUBJECT Files for each specimen loaded from DU_AnalyzeHipFemurV2
```

```
%dcm - structure containing all DICOM images
```

```
%dcm_slices_input - structure containing the DICOM slices for each Right
%and Left Legs
```

```
%RL - Right/Left identifier
```

```
%Outputs:
```

```
%nodeOutput - smoothed finalized nx4 list of nodes that make up the ICDM
%mesh to the users heights
```

```
%elemOutput - connectivity list for smoothed finalized nx4 list of nodes
%making up the ICDM mesh
```

Top Level Code overview

```
%STEP 1: Assigning Optional Parameters
%STEP 2: Identification of Desired Mesh Heights
```

```
%STEP 3: Raw ICDM Node Segmentation
%Step 4: Smoothing of Raw ICDM Node Segmentation
```

STEP 1: Assigning Optional Parameters

```
save_mesh_img=0;
save_table=0;
numPointsSliceInterpolate_input=300;

diaphysis_percents_input=0.5;
save_img=0;
img_path='R:\Research
Common\EBL\Anthropometrics\Hip_Implantations\Results\Diaphysis Meshing Updates\';
% Loading optional arguments
while ~isempty(varargin)
    switch lower(varargin{1})
        case 'diaphysispercents'
            diaphysis_percents_input= varargin{2};
        case 'numpointsinterpolate'
            numPointsSliceInterpolate_input = varargin{2};
        case 'upperlimit'
            upperLimit=varargin{2};
        case 'plottogglemain'
            plot_var = varargin{2};
        case 'plottogglesmooth'
            plot_var_smooth= varargin{2};
        case 'saveimgtoggle'
            save_img=varargin{2};
        case 'savemeshimgtoggle'
            save_mesh_img=varargin{2};
        case 'imagepath'
            img_path=varargin{2};
        otherwise
            error(['Unexpected option: ' varargin{1}])
    end
    varargin(1:2) = [];
end
```

STEP 2: Identification of Desired Mesh Heights

```
if strcmp(RL, 'R')==1
    boneName='RFemur';
    LT_Nodes= (SUB.TM.F.GtoAPR*SUB.GEO.A.REGIONS.RFemur.LESS_TROCH)';
    LowerLimit=SUB.LM.G.RFemur_Manual(4,4)+ 0.3*(max(SUB.GEO.G.RFemur.nds(:,4)) -
min(SUB.GEO.G.RFemur.nds(:,4)));
elseif strcmp(RL, 'L')==1
    boneName='LFemur';
```

```

LT_Nodes= (SUB.TM.F.GtoAPL*SUB.GEO.A.REGIONS.LFemur.LESS_TROCH)';
LowerLimit=SUB.LM.G.LFemur_Manual(4,4)+ 0.3*(max(SUB.GEO.G.LFemur.nds(:,4)) -
min(SUB.GEO.G.LFemur.nds(:,4)));
end

%identification of diaphysis nodes for thresholding
SI_d=find(dcm_slices_input(:,2)>=lowerLimit & dcm_slices_input(:,2)<=upperLimit );
SI_flag=find(dcm_slices_input(:,2)>=min(LT_Nodes(:,4))-10 &
dcm_slices_input(:,2)<=upperLimit);
%pulling the dicom stack over that range and finding the maximum value over
%that range
dcm_stack_diaphysis=dcm.stack(:, :, min(SI_d):max(SI_d));
dcm_stack_diaphysis=double(dcm_stack_diaphysis);
maxValue=max(max(max(dcm_stack_diaphysis)));

```

STEP 3: Raw ICDM Node Segmentation

```

count_nodes=1;
slicesBONE=dcm_slices_input(SI_d,:);

%pulling the slice thickness
ST=dcm.info{1}.SliceThickness;
%looping through all slices
for countSI= 1:length(slicesBONE)
    % Transforming all applicable structures

    %Getting the slice indices
    slice_dcm=slicesBONE(countSI,1);
    %checking SI_flag for metaphyseal femur
    if any(SI_flag==slicesBONE(countSI,2))
        diaphysisPercents=0.55;
    else
        diaphysisPercents=diaphysis_percents_input;
    end

    %storing the pixel spacing to the patient in a vector for
    ps=[dcm.info{slice_dcm}.PixelSpacing(1),dcm.info{slice_dcm}.PixelSpacing(2)];
    %transformation of coordinates

    pos=[dcm.info{slice_dcm}.ImagePositionPatient(1),dcm.info{slice_dcm}.ImagePosition
    Patient(2),dcm.info{slice_dcm}.ImagePositionPatient(3)];    %Position to patient
    for coordinate transformation
        % Using the indices of the bone to create a bounding box at that slice
        slice_range=[slicesBONE(countSI,2)- (ST*0.5) slicesBONE(countSI,2)+(ST*0.5)];

        nodes_slice=SUB.GEO.G.(boneName).nds(SUB.GEO.G.(boneName).nds(:,4)>=slice_range(1)
        & SUB.GEO.G.(boneName).nds(:,4)<=slice_range(2),:); %making everything else a nan
        for plotting/ further indexing purposes

```

```

%transforming to dicom space
nodes_slice_dcm=nodes_slice;
%transforming the x and y coordinates to DICOM space
nodes_slice_dcm(:,2)=(nodes_slice(:,2)- pos(1))/ps(1) + 1;
nodes_slice_dcm(:,3)=(nodes_slice(:,3)- pos(2))/ps(2) + 1;

%Extracting the centroid of the original nodes
centerInterpolatedNodes=mean(nodes_slice_dcm(:,:),1); %taking the mean along
the columns for x,y,z mean

% Interpolating the slice nodes
%initializing the linear length vector
bndry_linear_length=0; norm_bndry=[];

%Creating the vector of spline points
x_pts=linspace(0,1,numPointsSliceInterpolate_input);
y_pts=linspace(0,1,numPointsSliceInterpolate_input);

% Finding the boundary of the given nodes to then be used

[indices_bndry]=unique(boundary(nodes_slice_dcm(:,2),nodes_slice_dcm(:,3),0.35), 's
table');

% find the line length and normalize each point to that for x and y points
% Calculating the linear distance that each of the points ends up contributing
to
count=1;
for count_bndry=1:length(indices_bndry)
    %pulling the points for each boundary
    bndry=indices_bndry(count_bndry);
    %Logic if indices are on the end of the list
    if bndry == length(nodes_slice_dcm(:,1))
        pt1=nodes_slice_dcm(bndry,2:3);
        pt2=nodes_slice_dcm(1,2:3);
    else
        %pulling each of the nodal points
        pt1=nodes_slice_dcm(bndry,2:3);
        pt2=nodes_slice_dcm(bndry+1,2:3);
    end
    norm_bndry(count)=norm(pt2-pt1);
    %finding the magnitude of the distance between each point and
    %summing those values
    bndry_linear_length=bndry_linear_length + norm_bndry(count);
    %updating the count
    count=count+1;
end

if bndry_linear_length ~=0
    %calculating the normalized lengths for each point and organizing them

```

```

%using a cumulative summation
temp_norm_vector = cumsum(norm_bndry/bndry_linear_length)';

%Determination of Spline Determined Points
%using the cubic smoothing spline to determine the smoothed curve
x_csaps=csaps(temp_norm_vector,nodes_slice_dcm(indices_bndry,2), 0.99999);
y_csaps=csaps(temp_norm_vector,nodes_slice_dcm(indices_bndry,3), 0.99999);

%evaluating the function at each point
x_spline_nodes=fval(x_csaps,x_pts);
y_spline_nodes=fval(y_csaps,y_pts);

%storing in the temp structure
slice_nodes_spline(:,2:3)=[x_spline_nodes;y_spline_nodes]';
slice_nodes_spline(:,4)=slicesBONE(countSI,2);

%Using the convex HULL to create the bounding polyshape to then be
%partitioned
conv_hull_ind=convhull(slice_nodes_spline(:,2),slice_nodes_spline(:,3));

temp_ps=polyshape(slice_nodes_spline(conv_hull_ind,2),slice_nodes_spline(conv_hull_
_ind,3));
[temp_xc, temp_yc]=centroid(temp_ps);
deg=0:360/numPointsSliceInterpolate_input:360; %Creating the vector that
goes by the rotational increment to 360 degrees
v_length=100; %setting the vector length to be 100 pixels
count=1;

if any(SI_flag==slicesBONE(countSI,2))
    for count_deg=1:length(deg)-1
        deg_inc=deg(count_deg); %extracting on rotational degree increment
        %Setting up the changes in the angles as a function of degrees
        s=sind(deg_inc);c=cosd(deg_inc);
        %updating the endpoints of the line
        temp_line_seg=[temp_xc , temp_yc; temp_xc + c*v_length , temp_yc +
s*v_length];
        temp_line_seg2=[temp_xc-10 , temp_yc; temp_xc-10 + c*v_length ,
temp_yc + s*v_length];
        temp_line_seg3=[temp_xc+10 , temp_yc; temp_xc+10 + c*v_length ,
temp_yc + s*v_length];

        %finding which portion of the line segment intersects with the
        %cortical bounding box

[in_x,in_y]=polyxpoly(slice_nodes_spline(conv_hull_ind,2),slice_nodes_spline(con
v_hull_ind,3),temp_line_seg(:,1),temp_line_seg(:,2));

[in_x2,in_y2]=polyxpoly(slice_nodes_spline(conv_hull_ind,2),slice_nodes_spline(con
v_hull_ind,3),temp_line_seg2(:,1),temp_line_seg2(:,2));

```

```

[in_x3,in_y3]=polyxpoly(slice_nodes_spline(conv_hull_ind,2),slice_nodes_spline(con
v_hull_ind,3),temp_line_seg3(:,1),temp_line_seg3(:,2));

    %finding the average of the selected points
    avgx=mean([in_x,in_x2,in_x3]);
    avgy=mean([in_y,in_y2,in_y3]);

    %Storing
    temp_nodes_slice_rot(count,:)= [1,avgx,avgy,pos(3)];

    %Updating the count
    count=count+1;
end

else

    for count_deg=1:length(deg)-1
        deg_inc=deg(count_deg); %extracting on rotational degree increment
        %Setting up the changes in the angles as a function of degrees
        s=sind(deg_inc);c=cosd(deg_inc);
        %updating the endpoints of the line
        temp_line_seg=[temp_xc , temp_yc; temp_xc + c*v_length , temp_yc +
s*v_length];
        %finding which portion of the line segment intersects with the
        %cortical bounding box

[in_x,in_y]=polyxpoly(slice_nodes_spline(conv_hull_ind,2),slice_nodes_spline(conv_
hull_ind,3),temp_line_seg(:,1),temp_line_seg(:,2));

        temp_nodes_slice_rot(count,:)= [1,in_x,in_y,pos(3)]; %storing if
the line intersects with the bounding polygon and if the slice has a zero at the
rounded pixel value

        count=count+1;
    end
end
%Storing
nodesSliceDCMInterpolated=temp_nodes_slice_rot;
end
% Clearing All Loop Variables
clear norm_bndry temp_nodes_slice temp_norm_vector temp_nodes_slice_rot

%Looping Through Upsampled Slice Nodes

%initializing all vectors before entering the loop

sliceDecimatedTempHUVector=ones(numPointsSliceInterpolate_input,size(nodesSliceDCM
Interpolated,1)); %this structure will store all the information about the pixel
locations between the mean and the pixel of interest as well as the HU or BMD
pixel data for each

```



```

calcarDecimatedTempCMAP=sliceDecimatedTempHUVector; %initializing the color
map for visualization
for countsSliceNodes=1:size(nodesSliceDCMInterpolated,1)

    % Radial Profile Line Nodes Using The Algorithm
    %Using the sign of each value to determine the value that will be
    %used to create the vectors
    ind1X=centerInterpolatedNodes(:,2);
    ind2X=nodesSliceDCMInterpolated(countsSliceNodes,2);

    ind1Y=centerInterpolatedNodes(:,3);
    ind2Y=nodesSliceDCMInterpolated(countsSliceNodes,3);

    %creating a temporary vector that goes from the node of interest to the
    %centroid coordinates
    tempVec_XDCM=linspace(ind1X,ind2X,numPointsSliceInterpolate_input);
    tempVec_YDCM=linspace(ind1Y,ind2Y,numPointsSliceInterpolate_input);
    tempVec_ZDCM=ones(size(tempVec_XDCM,2),1).*slicesBONE(countSI,2);
    tempVec_P=linspace(0,1,numPointsSliceInterpolate_input);

    tempVec_DCM=[ones(length(tempVec_XDCM),1 ),tempVec_XDCM', tempVec_YDCM' ,
tempVec_ZDCM];
    %storing
    node_plot{countsSliceNodes}.nds= tempVec_DCM;

    %looping through each point and snagging the appropriate value from the
    %Dicom
    for countLinspace=1:numPointsSliceInterpolate_input
        %pulling and storing the pixel value at each of the equally spaced
        %nodes

sliceDecimatedTempHUVector(countLinspace,countSliceNodes)=dcm.stack(fix(tempVec_DC
M(countLinspace,3)),fix(tempVec_DCM(countLinspace,2)),slice_dcm);
        temp=sliceDecimatedTempHUVector(countLinspace,countSliceNodes);
        if temp<0
            temp=0; %rounding down to 0 if the pixel value is below 0 since
scans are unsigned
        end
        calcarDecimatedTempCMAP(countLinspace,countSliceNodes)=temp/maxValue;
    end

    %finding the maximum indices of the profile line, ideally the
    %two largest should be the two peaks of interest
    [maxDCMValue,maxDCM]=max(sliceDecimatedTempHUVector(:,countSliceNodes));
    [~,minDCM]=min(sliceDecimatedTempHUVector(1:maxDCM,countSliceNodes));

dcm_profile_line_cropped=sliceDecimatedTempHUVector(minDCM:maxDCM,countSliceNodes)
;
    tempVec_P_cropped=tempVec_P(minDCM:maxDCM);

```

```

%finding the global minimum
[min_value_cropped,~]=min(dcm_profile_line_cropped);

%Identifying points of intersection
target_height=((maxDCMValue-min_value_cropped)*diaphysisPercents);
horizontal_line_height1= target_height + min_value_cropped;
pt1_line_seg=[min(tempVec_P) max(tempVec_P);horizontal_line_height1
horizontal_line_height1]';

[pt1_ind,~]=polyxpoly(pt1_line_seg(:,1),pt1_line_seg(:,2),tempVec_P_cropped,dcm_pr
ofile_line_cropped);
y_pt1=ones(length(pt1_ind),1)*horizontal_line_height1;

v=[ind2X,ind2Y]-[ind1X,ind1Y];
%finding the unit vector
uv=(v)/norm(v);

%finding x and y pixel values
for countIdentifiedPts=1:length(pt1_ind)
    %scaling the distance of the line based on the percentage
    scaled_norm=pt1_ind(countIdentifiedPts)*norm(v);
    pts=uv*scaled_norm;
    %storing

indices_of_pts(countIdentifiedPts,:)= [pt1_ind(countIdentifiedPts),centerInterpolat
edNodes(:,2)+pts(1),centerInterpolatedNodes(:,3)+pts(2) ];
end

%storing each node in the dicom space
temp_output=ones(length(pt1_ind),4);
temp_output(:,2)=temp_output(:,2).*indices_of_pts(:,2);
temp_output(:,3)=temp_output(:,3).*indices_of_pts(:,3);
%returning to global space
temp_output(:,4)=temp_output(:,4).*slicesBONE(countSI,2); %storing as the
global indices

%Storing the Minimum Distance
distanceNodeToAPoint= sqrt(sum(abs(temp_output(:,2:3))-
nodesSliceDCMInterpolated(countSliceNodes,2:3)).^2,2));
[~,ind_to_keep]=min(distanceNodeToAPoint);

%Condition to remove or keep the given nodes based on distance to
%centroid
if vecnorm(centerInterpolatedNodes(:,2:3)-
temp_output(ind_to_keep,2:3),2)<0.5
    nodeOutputDCM(count_nodes,:)= [0 temp_output(ind_to_keep,2:4
else
    nodeOutputDCM(count_nodes,:)=temp_output(ind_to_keep,1:4);
end
end

```

```

% Clearing and Updating Count
clear temp_output indices_of_pts

count_nodes=count_nodes+1;
end
% Transforming back to Global CT Space
nodeOutputUnsmoothed=nodeOutputDCM;
nodeOutputUnsmoothed(:,2)=((nodeOutputDCM(:,2)-1)*ps(1))+pos(1);
nodeOutputUnsmoothed(:,3)=((nodeOutputDCM(:,3)-1)*ps(2))+pos(2);
end

```

Step 4: Smoothing of Raw ICDM Node Segmentation

```

%renaming and removing any weirdly identified nodes
nodeOutputUnsmoothed_Final=nodeOutputUnsmoothed(nodeOutputUnsmoothed(:,1)==1,:);

%identifying the unique SI heights
uniqueIndices=unique(nodeOutputUnsmoothed_Final(:,4));
for countj=1:length(uniqueIndices)
    temp_SI(countj,1)=find(slicesBONE(:,2)==uniqueIndices(countj));
end
%Smoothing the Nodes
[nodeOutput,elemOutput]=DU_MeshDiaphysis_V4(SUB,RL,nodeOutputUnsmoothed_Final,dcm,
slicesBONE,temp_SI,SI_flag,'numptsinterpolate',100,'plottoggle',1);

disp('Plotting Final ICDM Mesh')
if strcmp(RL,'R')==1
    figure();set(gcf,'color',[1 1 1]);
    %patch('Vertices',
SUB.GEO.G.RFemur.nds(:,2:4),'Faces',SUB.GEO.G.RFemur.faces(SUB.GEO.G.RFemur.neck_e
lement_list,2:4),'FaceColor',[227/256 218/256
201/256],'FaceLighting',' gouraud','EdgeColor','none','SpecularStrength',.25,'FaceA
lpha',0.35); hold on

    patch('Vertices',nodeOutput(:,2:4),'Faces',elemOutput(:,2:4),'FaceColor','c','Face
Lighting',' gouraud','EdgeColor','k','EdgeAlpha',0.3,'SpecularStrength',.25,'FaceA
lpha',1.0,'Clipping','off'); hold on

    %patch('X',nodeOutputUnsmoothed_Final(:,2),'Y',nodeOutputUnsmoothed_Final(:,3),'Z'
,nodeOutputUnsmoothed_Final(:,4),'FaceColor',[0.5 0.5
0.5],'FaceLighting',' gouraud','EdgeColor','k','EdgeAlpha',0.3,'SpecularStrength',.
25,'FaceAlpha',0.1)
    light
    title(['Diaphysis Segmentation: ', SUB.ID, ' RFemur'],'FontSize',18)
    axis equal; axis off;
else
    figure();set(gcf,'color',[1 1 1]);
    %patch('Vertices',
SUB.GEO.G.LFemur.nds(:,2:4),'Faces',SUB.GEO.G.LFemur.faces(SUB.GEO.G.LFemur.neck_e

```

```

lement_list,2:4),'FaceColor',[227/256 218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','SpecularStrength',.25,'FaceAl
pha',0.35); hold on

patch('Vertices',nodeOutput(:,2:4),'Faces',elemOutput(:,2:4),'FaceColor','c','Face
Lighting','gouraud','EdgeColor','k','EdgeAlpha',0.3,'SpecularStrength',.25,'FaceAl
pha',1.0,'Clipping','off'); hold on

%patch('X',nodeOutputUnsmoothed_Final(:,2),'Y',nodeOutputUnsmoothed_Final(:,3),'Z'
,nodeOutputUnsmoothed_Final(:,4),'FaceColor',[0.5 0.5
0.5],'FaceLighting','gouraud','EdgeColor','k','EdgeAlpha',0.3,'SpecularStrength',.
25,'FaceAlpha',0.1)
    light
    title(['Diaphysis Segmentation: ' SUB.ID, ' LFemur'],'FontSize',18)
    axis equal; axis off;
end
end

```

DU_MeshDiaphysis_Final

```

function [nodes_o,elems_o] =
DU_MeshDiaphysis_Final(H,RL,pc_nodes,dcm,dcm_slices,SI_v,SI_flag,varargin)

%DU_MeshDiaphysis Code by Sam Mattei 10/26/2022
%This code will take a given unfiltered point cloud and filter it and return
%a final mesh to the user for the given nodes with equal spacing and number of
points.

%The Gibbon Code Toolbox has been added to this heirarchy for additional
%smoothing schtuff :) make sure it is present when you run.

%Explanation of Input Variables
% H - subject specific stl structure
% dcm - structure containing all dicom files and information related
% SI_vector - vector describing the overall number of slices the code will
% be ran on
% numPtsSlice_input - the number of points per each mesh slice

%sliceThresh_Input - the spline threshold that can range from 0-1 * I put this
%closer to 1 than 0 found it better approximates the per slice graphs

%ZThresh_Input - the spline threshold that will range over the entire SI_v also
%goes from 0-1.

%plot_var - integer describing if variables will be plotted
%save_img - integer describing if variables will be saved

%Output Variables

```

```

%nodes - structure containing all the nodes for a given mesh
%elems - structure containing all the connection definitions for the given splined
mesh

% Looping through each slice

```

Loading in Optional Parameters

```

numPtsSlice_input=100;
sliceThresh_Input=[0.99999,0.99999];
ZThresh_Input=[1E-2,1E-2];
windowLength_input=5;
plot_var=1;
save_img=0;
img_path='R:\Research Common\EBL\Anthropometrics\Hip_Implantations\Results\Calcar
Updates (1.3.2023)';
% Loading optional arguments
while ~isempty(varargin)
    switch lower(varargin{1})
        case 'numptsinterpolate'
            numPtsSlice_input = varargin{2};
        case 'slicethresh'
            sliceThresh_Input = varargin{2};
        case 'zthresh'
            ZThresh_Input = varargin{2};
        case 'plottoggle'
            plot_var = varargin{2};
        case 'savetoggle'
            save_img=varargin{2};
        case 'imagepath'
            img_path=varargin{2};
        case 'windowlength'
            windowLength_input=varargin{2};
        otherwise
            error(['Unexpected option: ' varargin{1}])
    end
    varargin(1:2) = [];
end

if strcmp(RL,'R')==1
    fv.vertices=H.GEO.G.RFemur.nds(:,2:4);
    fv.faces=H.GEO.G.RFemur.faces(:,2:4);
else
    fv.vertices=H.GEO.G.LFemur.nds(:,2:4);
    fv.faces=H.GEO.G.LFemur.faces(:,2:4);
end

```

Changing Node Numbering on each slice and downsampling

```

nodes=[0,0,0,0]; %initializing the matrix to hold the nodes
for count_SI= 1:length(SI_v)

```

Step 1: find the line length and normalize each point to that for x and y points

```

%pulling the indices of the dicom in millimeters
dcm_slice_mm=dcm_slices(SI_v(count_SI),2);
pos=dcm.info{SI_v(count_SI)}.ImagePositionPatient;
ps=dcm.info{SI_v(count_SI)}.PixelSpacing;
%Pulling the nodes for the given slice location in mm
temp_nodes_slice=pc_nodes(pc_nodes(:,4) == dcm_slice_mm,:);
nodeFilterDistance=zeros(length(temp_nodes_slice(:,1)),1);
centroidTempNodes=mean(temp_nodes_slice,1);

for count_nodes_slice=1:length(temp_nodes_slice(:,1))

nodeFilterDistance(count_nodes_slice,1)=sqrt(sum(abs(temp_nodes_slice(count_nodes_
slice,2:3)-centroidTempNodes(:,2:3)).^2,2));
end

mu=mean(nodeFilterDistance);
sigma=std(nodeFilterDistance);

% Identify Boundary of raw nodes
temp_nodes_slice_DistanceFiltered=temp_nodes_slice(nodeFilterDistance>=(mu-
2.5*sigma),:);
% Finding the boundary of the given nodes to then be used

[ib]=unique(boundary(temp_nodes_slice_DistanceFiltered(:,2),temp_nodes_slice_Dista
nceFiltered(:,3),0.5),'stable');
indices_bndry=[ib;ib(1)];

%Equal Spacing of Nodes on X and Y axes using interparc function to 100
%nodes
%figure out equidistant upsampling distances
upsampleDistanceInt=linspace(0,1,100);

upSampledPoints=interparc(upsampleDistanceInt,temp_nodes_slice_DistanceFiltered((i
ndices_bndry),2),temp_nodes_slice_DistanceFiltered((indices_bndry),3),'linear');

```

Reordering Node Numbering

```

%using identified slice centroid to draw a vertical plumbline and
%identifying the most posterior portion of the slice shape using this.
%+50 is to compensate for how the DICOM image draws the plumbline since
%up and down are flipped
clear centroidTempNodes

```

```

centroidTempNodes=mean(upSampledPoints,1);

tempLineSegment=[centroidTempNodes; [centroidTempNodes(1)
centroidTempNodes(2)+50]];

[xi,yi]=polyxpoly(upSampledPoints(:,1),upSampledPoints(:,2),tempLineSegment(:,1),t
empLineSegment(:,2));

%bit of logic to store points
if isempty(xi) || isempty(yi)
    SP=SP_last;
    SP_last=SP; %storing the last indice
    orderedNodesIND=(SP:length(upSampledPoints) )'; (1:SP-1)'; %creating a
new vector of the ordered indices
    upSampledPoints_SORTED=upSampledPoints(orderedNodesIND,:);
else
    %Identifying the nearest node to the right
    [IDX,D]=knnsearch(upSampledPoints,[xi,yi],"k",size(upSampledPoints,1));
    A=[IDX',D'];

    SP=A(1,1); %pulling the indices of the point with the minimum distance to
the plumbline
    SP_last=SP; %storing the last indice
    orderedNodesIND=(SP:length(upSampledPoints) )'; (1:SP-1)'; %creating a
new vector of the ordered indices
    upSampledPoints_SORTED=upSampledPoints(orderedNodesIND,:);
end
%storing in a smaller temporary array

nodes=vertcat(nodes,[ones(size(upSampledPoints_SORTED,1),1),upSampledPoints_SORTED
, ones(size(upSampledPoints_SORTED,1),1).*dcm_slice_mm]);

end

```

Pulling each downsampled node and tracking it over the volume of the mesh

```

nodes=nodes(nodes(:,1)~=0,:); %removing the first row
count=1;
elems=[0 0 0 0];

node_offset1=0;
node_offset2=numPtsSlice_input;
%pulling all unique nodes, excluding any slices that have been removed for
%meshing consistency
uns=sort(unique(nodes(:,4)),'ascend');

for count_SI=1:length(uns)-1
    for count_nodes=1:numPtsSlice_input
        if count_nodes ~=numPtsSlice_input

```

```

        %storing each element separately
        temp_elm1=[1 , node_offset1+count_nodes , node_offset1+count_nodes+1
,   node_offset2+count_nodes];
        temp_elm2=[1 , node_offset1+count_nodes+1 , node_offset2+count_nodes+1
,   node_offset2+count_nodes];
        %storing
        temp_elems_slice(count:count+1,:)=[temp_elm1;temp_elm2];
    else
        %storing each element separately
        temp_elm1=[1 , node_offset1+count_nodes , node_offset1+1 ,
node_offset2+count_nodes];
        temp_elm2=[1 , node_offset1+1 , node_offset2+1 ,
node_offset2+count_nodes];
        %storing
        temp_elems_slice(count:count+1,:)=[temp_elm1;temp_elm2];
    end
    %updating the count
    count=count+2;
end
node_offset1=node_offset1 + numPtsSlice_input;
node_offset2=node_offset2 + numPtsSlice_input;
end
%storing in the global structure
elems=temp_elems_slice(1:length(temp_elems_slice),:);

```

Splining over the volume to remove any final noise

```

temp_indices_mm=unique(nodes(:,4));
num_slices=[0:length(temp_indices_mm)-1]';
nodes_spline=ones(size(nodes,1),4);

%pulling the point on each slice
for count_points=1:numPtsSlice_input
    %disp(count_points)
    temp_indices=count_points+(numPtsSlice_input*(num_slices))'; %storing each
point in its own array
    totalNodesToSmooth=nodes(temp_indices,:);

    [~,tempNodesX]=movingSmartWindow(num_slices,totalNodesToSmooth(:,2));
    [croppedX,tempNodesY]=movingSmartWindow(num_slices,totalNodesToSmooth(:,3));

nodesToSmooth=[ones(size(croppedX,1),1),tempNodesX,tempNodesY,ones(size(croppedX,1)
),1)]; %setting first and last columns to ones for consistency
    %including a bit of outlier removal code, removing anything outside of
%2.5 standard deviations
    node_rangeX=[mean(nodesToSmooth(:,2))-
3*std(nodesToSmooth(:,2)),mean(nodesToSmooth(:,2))+3*std(nodesToSmooth(:,2))];

```



```

node_rangeY=[mean(nodesToSmooth(:,3))-
3*std(nodesToSmooth(:,3)),mean(nodesToSmooth(:,3))+3*std(nodesToSmooth(:,3))];

tempNodesToSmoothX=find(nodesToSmooth(:,2)<=node_rangeX(1) |
nodesToSmooth(:,2)>=node_rangeX(2));
tempNodesToSmoothY=find(nodesToSmooth(:,3)<=node_rangeY(1) |
nodesToSmooth(:,3)>=node_rangeY(2));

tempNodesToSmooth=unique([tempNodesToSmoothX;tempNodesToSmoothY]);
%tempNodesToSmooth=[];
%setting the outliers to Nan
nodesToSmooth(tempNodesToSmooth,2:4)=NaN;

weights=ones(size(num_slices));
%
weights(floor(0.8*(length(num_slices))):end)=weights(floor(0.8*length(num_slices))
:end).*0.05;

%using the cubic smothing spline to determine the smoothed curve
x_csaps_z=csaps(num_slices,nodesToSmooth(:,2), 1E-2,[],weights);
y_csaps_z=csaps(num_slices,nodesToSmooth(:,3), 1E-2,[],weights);

% Pulling the smoothed values of each curve
x_spline_nodes_z=fnval(x_csaps_z,num_slices);
y_spline_nodes_z=fnval(y_csaps_z,num_slices);

%Viz of this
plot_var=0;
if plot_var ==1
test_fig2=figure(11);set(gcf,'color',[1 1 1]); hold on
% set(gcf, 'Position', get(0, 'Screensize'));
subplot(1,2,1)
%plotting as a function of normalized length
plot(num_slices,nodesToSmooth(:,2),'linestyle','-
','color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
%plotting the x coordinates
fplot(x_csaps_z,'r:')
plot(num_slices,x_spline_nodes_z,'linestyle','-
','color','k','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
',0.5);
title('X Nodal Location Versus Dicom Slice Number' )
ylabel('Nodal Coordinate Location [mm]')
xlabel('Dicom Slice Number')
% legend(['X Nodal Locations','CSAPS X Nodal Predictions'])

xlim([min(num_slices) max(num_slices)])
ylim([min(nodes(temp_indices,2))-10, max(nodes(temp_indices,2))+10])

subplot(1,2,2)

```

```

        plot(num_slices,nodesToSmooth(:,3),'linestyle','-
        ','Color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
    %plotting the y coordinates
        fplot(y_csaps_z,'r:')
        plot(num_slices,y_spline_nodes_z,'linestyle','-
        ','Color','k','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
        ',0.5);

        title('Y Nodal Location Versus Dicom Slice Number')
        ylabel('Nodal Coordinate Location [mm]')
        xlabel('Dicom Slice Number')
        %         legend(['Y Nodal Locations','CSAPS Y Nodal Predictions'])
        xlim([min(num_slices) max(num_slices)])
        ylim([min(nodes(temp_indices,3))-10, max(nodes(temp_indices,3))+10])

        pause(0.25)
        close (test_fig2)
    end
    %Storing the updated points
    nodes_spline(temp_indices,2:4)=[x_spline_nodes_z , y_spline_nodes_z ,
temp_indices_mm];
end

```

Cubic Spline Smoothing Functions

```

nodes_spline=ones(size(nodes,1),4);
breakVector=length(SI_v)-length(SI_flag); %finding where the diaphysis ends
%looping through each of the splined points picked by the user
for count_points=1:numPtsSlice_input

    num_slices_diaphysis=0:length(SI_v)-length(SI_flag)-1;
    num_slices_metaphysis=0:length(SI_flag)-1;

    %pulling the generic indices for all the points
    temp_indices=count_points+(numPtsSlice_input*(num_slices))'; %storing each
point in its own array
    totalNodesToSmooth=nodes(temp_indices,:); %every node on the list

    movingSmartWindow(num_slices,totalNodesToSmooth(:,2));
    movingSmartWindow(num_slices,totalNodesToSmooth(:,3));

```

Smoothing Diaphysis

```

diaphysisNodesToSmooth=totalNodesToSmooth(1:breakVector,:);
num_slices_diaphysis_temp=num_slices_diaphysis;
diaphysisNodesToSmooth_temp=diaphysisNodesToSmooth;
%including a bit of outlier removal code, removing anything outside of
%2.5 standard deviations

```

```

node_rangeX=[mean(diaphysisNodesToSmooth_temp(:,2))-
2*std(diaphysisNodesToSmooth_temp(:,2)),mean(diaphysisNodesToSmooth_temp(:,2))+2*std(diaphysisNodesToSmooth_temp(:,2))];
node_rangeY=[mean(diaphysisNodesToSmooth_temp(:,3))-
2*std(diaphysisNodesToSmooth_temp(:,3)),mean(diaphysisNodesToSmooth_temp(:,3))+2*std(diaphysisNodesToSmooth_temp(:,3))];

tempNodesToSmoothX=find(diaphysisNodesToSmooth_temp(:,2)<=node_rangeX(1) |
diaphysisNodesToSmooth_temp(:,2)>=node_rangeX(2));
tempNodesToSmoothY=find(diaphysisNodesToSmooth_temp(:,3)<=node_rangeY(1) |
diaphysisNodesToSmooth_temp(:,3)>=node_rangeY(2));

tempNodesToSmooth=unique([tempNodesToSmoothX;tempNodesToSmoothY]);
%tempNodesToSmooth=[];
%setting the outliers to Nan
diaphysisNodesToSmooth_temp(tempNodesToSmooth,:)=[];
num_slices_diaphysis_temp(tempNodesToSmooth)=[];
%using the cubic smothing spline to determine the smoothed curve
x_csaps_z=csaps(num_slices_diaphysis_temp,diaphysisNodesToSmooth_temp(:,2),
1E-3);
y_csaps_z=csaps(num_slices_diaphysis_temp,diaphysisNodesToSmooth_temp(:,3),
1E-3);

% Pulling the smoothed values of each curve
xDiaphysisNodes=fnval(x_csaps_z,0:length(SI_v)-length(SI_flag)-1);
yDiaphysisNodes=fnval(y_csaps_z,0:length(SI_v)-length(SI_flag)-1);

```

Smoothing Metaphysis

```

num_slices_metaphysis_temp=num_slices_metaphysis;

metaphysisNodesToSmooth=totalNodesToSmooth(breakVector+1:length(totalNodesToSmooth),:);

metaphysisNodesToSmooth_temp=metaphysisNodesToSmooth;
%including a bit of outlier removal code, removing anything outside of
%2.5 standard deviations
node_rangeX=[mean(metaphysisNodesToSmooth_temp(:,2))-
2*std(metaphysisNodesToSmooth_temp(:,2)),mean(metaphysisNodesToSmooth_temp(:,2))+2*std(metaphysisNodesToSmooth_temp(:,2))];
node_rangeY=[mean(metaphysisNodesToSmooth_temp(:,3))-
2*std(metaphysisNodesToSmooth_temp(:,3)),mean(metaphysisNodesToSmooth_temp(:,3))+2*std(metaphysisNodesToSmooth_temp(:,3))];

tempNodesToSmoothX=find(metaphysisNodesToSmooth_temp(:,2)<=node_rangeX(1) |
metaphysisNodesToSmooth_temp(:,2)>=node_rangeX(2));
tempNodesToSmoothY=find(metaphysisNodesToSmooth_temp(:,3)<=node_rangeY(1) |
metaphysisNodesToSmooth_temp(:,3)>=node_rangeY(2));
tempNodesToSmoothM=unique([tempNodesToSmoothX;tempNodesToSmoothY]);

```

```

metaphysisNodesToSmooth_temp(tempNodesToSmoothM,:)=[];
num_slices_metaphysis_temp(tempNodesToSmoothM)=[];
%using the cubic smothing spline to determine the smoothed curve
x_csaps_zM=csaps(num_slices_metaphysis_temp,metaphysisNodesToSmooth_temp(:,2),
1E-3);
y_csaps_zM=csaps(num_slices_metaphysis_temp,metaphysisNodesToSmooth_temp(:,3),
1E-3);

% Pulling the smoothed values of each curve
xMetaphysisNodes=fnval(x_csaps_zM,0:length(SI_flag)-1);
yMetaphysisNodes=fnval(y_csaps_zM,0:length(SI_flag)-1);

%Debugging Visuals
if plot_var ==10 && strcmp(debug,'d')==1
    test_fig2=figure(11);set(gcf,'color',[1 1 1]); hold on
    %       set(gcf, 'Position', get(0, 'Screensize'));
    subplot(1,2,1)
    %plotting as a function of normalized length
    plot(num_slices_diaphysis,diaphysisNodesToSmooth(:,2),'linestyle','-
','color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
%plotting the x coordinates
    fplot(x_csaps_z,'r:')
    plot(num_slices_diaphysis,xDiaphysisNodes,'linestyle','-
','color','k','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
',0.5);
    title('X Nodal Location Versus Dicom Slice Number' )
    ylabel('Nodal Coordinate Location [mm]')
    xlabel('Dicom Slice Number')
    %       legend(['X Nodal Locations','CSAPS X Nodal Predictions'])

    xlim([min(num_slices_diaphysis) max(num_slices_diaphysis)])
    ylim([min(nodes(temp_indices,2))-10, max(nodes(temp_indices,2))+10])

    subplot(1,2,2)
    plot(num_slices_diaphysis,diaphysisNodesToSmooth(:,3),'linestyle','-
','color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
%plotting the y coordinates
    fplot(y_csaps_z,'r:')
    plot(num_slices_diaphysis,yDiaphysisNodes,'linestyle','-
','color','c','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
',0.5);
    title('Y Nodal Location Versus Dicom Slice Number')
    ylabel('Nodal Coordinate Location [mm]')
    xlabel('Dicom Slice Number')
    %       legend(['Y Nodal Locations','CSAPS Y Nodal Predictions'])
    xlim([min(num_slices_diaphysis) max(num_slices_diaphysis)])
    ylim([min(nodes(temp_indices,3))-10, max(nodes(temp_indices,3))+10])

    pause(0.25)
    close (test_fig2)
elseif plot_var ==10 && strcmp(debug,'m')==1

```

```

test_fig2=figure(11);set(gcf,'color',[1 1 1]); hold on
%       set(gcf, 'Position', get(0, 'Screensize'));
subplot(1,2,1)
%plotting as a function of normalized length
plot(num_slices_metaphysis,metaphysisNodesToSmooth(:,2),'linestyle','-
','color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
%plotting the x coordinates
fnp1t(x_csaps_zM,'r:')
plot(num_slices_metaphysis,xMetaphysisNodes,'linestyle','-
','color','c','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
',0.5);
title('X Nodal Location Versus Dicom Slice Number' )
ylabel('Nodal Coordinate Location [mm]')
xlabel('Dicom Slice Number')
%       legend(['X Nodal Locations','CSAPS X Nodal Predictions'])

xlim([min(num_slices_metaphysis) max(num_slices_metaphysis)])
ylim([min(nodes(temp_indices,2))-10, max(nodes(temp_indices,2))+10])

subplot(1,2,2)
plot(num_slices_metaphysis,metaphysisNodesToSmooth(:,3),'linestyle','-
','color','k','Marker','o','MarkerFaceColor','r','MarkerSize',2); hold on
%plotting the y coordinates
fnp1t(y_csaps_zM,'r:')
plot(num_slices_metaphysis,yMetaphysisNodes,'linestyle','-
','color','k','Marker','o','MarkerFaceColor','b','MarkerEdgeColor','b','MarkerSize
',0.5);
title('Y Nodal Location Versus Dicom Slice Number')
ylabel('Nodal Coordinate Location [mm]')
xlabel('Dicom Slice Number')
%       legend(['Y Nodal Locations','CSAPS Y Nodal Predictions'])
xlim([min(num_slices_metaphysis) max(num_slices_metaphysis)])
ylim([min(nodes(temp_indices,3))-10, max(nodes(temp_indices,3))+10])

pause(0.25)
close (test_fig2)
end

% Storing the updated points
x_spline_nodes_z=[xDiaphysisNodes, xMetaphysisNodes ]';
y_spline_nodes_z=[yDiaphysisNodes, yMetaphysisNodes ]';
nodes_spline(temp_indices,2:4)=[x_spline_nodes_z , y_spline_nodes_z ,
temp_indices_mm];

end
%}

%Laplacian Based Volume Smoothing and Outlier Removal
%Gibbon Toolbox Smoothing Function
[Vs]=trisurfSmoothFourier(elems(:,2:4),nodes_spline(:,2:4),500);
%[Vs]=taubinsmooth(elems(:,2:4),nodes_spline(:,2:4),50,0.75,0.78);

```

```

nodes_smoothed=[ones(size(Vs,1),1),Vs];
%Assigning Outputs
nodes_o=nodes_smoothed;
elems_o=elems;

```

```
end
```

Helper Functions

```

function [x_crop,v_crop]=movingSmartwindow(x,v)
%This function is designed to be a smart moving window filter to remove
%noise in the x,y signals for the meshes prior to CSAPS splining. It
%does so by not including the first 10 points of the mesh to build the
>window and then correcting and adding each point to the moving window.

iwl=10; %setting the initial window length
%Creating the corrected vectors
corrected_v=[v,v,v];
corrected_x=[x,x,x];

for i=1:size(v,1)-10
    if i<=iwl %excluding the first series of points prior to adding to the
window
        window=[v(1:iwl,1)];
        %calculate the window, std, and avg of all prior points
        w_avg=mean(window);w_std=std(window);

w_median=median(window);w_quan=quantile(window,[0.25,0.75]);w_iqr=iqr(window);

        %checking the current vectors trajectory with respect to the
        %points that exist already
        ptToCheck=v(i);

        %check 1 looks at the mean and standard deviation to remove
        %data
        check1=[ptToCheck>=w_avg+3*w_std, ptToCheck<=w_avg-3*w_std ];
        %check 2 uses the interquartile range to remove data
        %check2=[ptToCheck<= w_quan(1)- 1.5*(w_iqr) , ptToCheck>= w_quan(2) +
1.5*(w_iqr) ];
        if any(check1==1)
            %augmenting the point by either removing the node temporarily by Nan
            or correcting it to
            %the biased moving average
            corrected_v(i,1)=w_avg;
            corrected_v(i,2)=w_median;
            corrected_v(i,3)=NaN;
            corrected_x(i,3)=NaN;
        end
    end
end

```

```

else %for all points outside of the initial window
    window=[corrected_v(i-iw1:i-1,1)]; %setting the window to have all
corrected points up until this point
    %calculate the window, std, and avg of all prior points
    w_avg=mean(window);w_std=std(window);
    w_quan=quantile(window,[0.25,0.75]);w_iqr=iqr(window);
    %checking the current vectors trajectory with respect to the
    %points that exist already
    ptToCheck=v(i);
    %check 1 looks at the mean and standard deviation to remove
    %data
    check1=[ptToCheck>=w_avg+3*w_std, ptToCheck<=w_avg-3*w_std ];
    %check 2 uses the interquartile range to remove data
    check2=[ptToCheck<= w_quan(1) - 1.5*(w_iqr) , ptToCheck>= w_quan(2) +
1.5*(w_iqr) ];
    %couple of checks
    if any(check1==1)
        %augmenting the point by either removing the node temporarily by Nan
or correcting it to
        %the biased moving average
        corrected_v(i,1)=mean(corrected_v(i-3:i,1));
        corrected_v(i,2)=mean(corrected_v(i-3:i,1));
        corrected_v(i,3)=NaN;
        corrected_x(i,3)=NaN;
    elseif any(check2==1)
        corrected_v(i,1)=median(corrected_v(i-3:i,2));
        corrected_v(i,2)=median(corrected_v(i-3:i,2));
        corrected_v(i,3)=NaN;
        corrected_x(i,3)=NaN;
    end
end
end
end
%Plotting
%{
MSW1=figure();set(gcf,'color',[1 1 1]); hold on
plot(x,v,'LineStyle','-','Marker','.','Color','k','MarkerSize',8);hold on
%plotting initial nodal signal
plot(corrected_x(:,1),corrected_v(:,1),'LineStyle','-
','Marker','none','Color','r','MarkerSize',8);hold on
plot(corrected_x(:,2),corrected_v(:,2),'LineStyle','-
','Marker','none','Color','g','MarkerSize',8)
plot(corrected_x(:,3),corrected_v(:,3),'LineStyle','-
','Marker','none','Color','b','MarkerSize',8)
legend('Slice Smoothed Node Position','Mean Window Averaged Values','Median WInow
Averaged Values','NaN Values','Orientation','Vertical','FontSize',14)

%}

%storing returned outputs

```

```
x_crop=corrected_x(:,3);
v_crop=corrected_v(:,3);
end
```

DU_GetDorr_Final

```
function [Dorr_output] =
DU_GetDorr_Final(SUB,RL,diaphysisNodes,diaphysisElems,plot_var)
```

DU_GetDorr - By Sam Mattei 7/25/2022

```
%This function is designed to get the Dorr classification of a femur from
%a mesh of the inner cortical wall. This can either be fed in from ScanIP
%or done by using the prior code. If the ScanIP method is used extra code
%will need to be written to allow for functionality of the software. This code
%uses https://www.orthobullets.com/recon/5003/tha-implant-fixation
%as a resource for selecting the correct Dorr classification and work done by
%Dorr et al. to calculate the ratios below. A more
%detailed explanation of the inputs and outputs are below:
```

```
%Inputs:
```

```
%SUB - SUBJECT Files for each specimen loaded from DU_AnalyzeHipFemurV2
```

```
%RL - Identifier for Right and Left
```

```
%Outputs:
```

```
%Dorr - a structure containing the Dorr classification, both inner cortical
%diameters as well as the ratio of the two. The Cortical Index as well and
%the CTRC
```

Top Level Code Overview

```
%STEP 1: Identification of the Lesser Trochanter (LT) and appropriate nodes
%STEP 2: Calculation of the canal-to-calcar ratio (CTCR) and cortical index
%(CI)
%STEP 3: Storing information for return
```

STEP 1: Identification of LT

```
if strcmp(RL,'R')==1
    %pulling the lesser trochanter nodes in the global space
    LT_nodes=(SUB.GEO.A.REGIONS.RFemur.LESS_TROCH)';
    diaphysisNodesA=(inv(SUB.TM.F.GtoAPR)*diaphysisNodes)';
```



```

boneA=SUB.GEO.A.RFemur.nds;
boneElems=SUB.GEO.G.RFemur.faces;
elseif strcmp(RL,'L')==1
    %pulling the lesser trochanter nodes in the global space
    LT_nodes=(SUB.GEO.A.REGIONS.LFemur.LESS_TROCH)';
    diaphysisNodesA=(inv(SUB.TM.F.GtoAPL)*diaphysisNodes)';
    boneA=SUB.GEO.A.LFemur.nds;
    boneElems=SUB.GEO.G.LFemur.faces;
end
%finding the mean of the supplied lesser trochanter mesh to determine the
%centroid of the mesh
centroid_LT= mean(LT_nodes,1);
% First location for the Dorr calculation, 3 cm distal from the midpoint of the
lesser trochanter (LT)
Dorr.LT3P=centroid_LT(:,4)-30; %mm correction factor of 10
% Second location for the Dorr calculation, 10 cm distal from the mid point of LT
Dorr.LT10P=centroid_LT(:,4)-100; %mm correction factor of 10

% STEP 2: Calculation of the CTCR Ratio
%pulling the nodes of the diaphysis that correspond to the 3cm and 10cm
%points
tempNodes3cm=find(diaphysisNodesA(:,4) >= Dorr.LT3P-0.3 & diaphysisNodesA(:,4)
<= Dorr.LT3P+0.3);
tempNodes10cm=find(diaphysisNodesA(:,4) >= Dorr.LT10P-0.3 & diaphysisNodesA(:,4)
<= Dorr.LT10P+0.3);
tempNodesBone10cm=find(boneA(:,4) >= Dorr.LT10P-0.3 & boneA(:,4) <=
Dorr.LT10P+0.3);
%ML direction is the 1 direction

tempv1=max(diaphysisNodesA(tempNodes3cm,2)) -
min(diaphysisNodesA(tempNodes3cm,2));
tempv2=max(diaphysisNodesA(tempNodes10cm,2)) -
min(diaphysisNodesA(tempNodes10cm,2));
tempv3=max(boneA(tempNodesBone10cm,2)) - min(boneA(tempNodesBone10cm,2));

%Third step is calculating the actual CTCR
Dorr.CalcarToCanalRatio=tempv2/tempv1; %Calcar to Canal ratio

%calculating the cortical thickness
Dorr.CorticalThickness=(tempv3-tempv2)/tempv3;

%fourth step is classifying the bone under a specific type of thingy
if Dorr.CalcarToCanalRatio <0.5
    Dorr.Classification='Type A';
elseif Dorr.CalcarToCanalRatio>=0.5 && Dorr.CalcarToCanalRatio<=0.75
    Dorr.Classification='Type B';
elseif Dorr.CalcarToCanalRatio>0.75
    Dorr.Classification='Type C';

```

```

end
[~,maxind3]=max(diaphysisNodesA(tempNodes3cm,2));
[~,minind3]=min(diaphysisNodesA(tempNodes3cm,2));

[~,maxind10]=max(diaphysisNodesA(tempNodes10cm,2));
[~,minind10]=min(diaphysisNodesA(tempNodes10cm,2));

[~,maxindB10]=max(boneA(tempNodesBone10cm,2));
[~,minindB10]=min(boneA(tempNodesBone10cm,2));

%Viz
if plot_var==1
    DUGD1=figure();set(gcf,'color',[1 1 1]);
    %patching the bone
    patch('Vertices',boneA(:,2:4),'Faces',boneElems(:,2:4),'FaceColor',[227/256
218/256 201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5);hold
on;
    %plotting the inner cortical mesh

    patch('Vertices',diaphysisNodesA(:,2:4),'Faces',diaphysisElems(:,2:4),'FaceColor',
[227/256 218/256
256/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5);
        plotting the 3cm horizontal line

    plot3([diaphysisNodesA(tempNodes3cm(minind3),2),diaphysisNodesA(tempNodes3cm(maxin
d3),2)], [diaphysisNodesA(tempNodes3cm(minind3),3),diaphysisNodesA(tempNodes3cm(max
ind3),3)], [diaphysisNodesA(tempNodes3cm(minind3),4),diaphysisNodesA(tempNodes3cm(m
axind3),4)],...
        'LineStyle','--','Linewidth',3.5,'Color','r');
        plotting the 10 cm horizontal line

    plot3([diaphysisNodesA(tempNodes10cm(minind10),2),diaphysisNodesA(tempNodes10cm(ma
xind10),2)], [diaphysisNodesA(tempNodes10cm(minind10),3),diaphysisNodesA(tempNodes1
0cm(maxind10),3)], [diaphysisNodesA(tempNodes10cm(minind10),4),diaphysisNodesA(temp
Nodes10cm(maxind10),4)],...
        'LineStyle','--','Linewidth',3.5,'Color','g');

    plot3(centroid_LT(:,2),centroid_LT(:,3),centroid_LT(:,4),'LineStyle','none','Marke
r','o','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14);

    plot3(0,0,150,'LineStyle','none','Marker','o','MarkerFaceColor','m','MarkerEdgeCo
lor','k','MarkerSize',14)

    view([180 0]) %Frontal
    axis([-70 70 -70 70 -250 50])
    axis equal; axis off
    legend(SUB.ID,[SUB.ID ' Inner Cortical Mesh'],'3cm Horizontal Line','10cm
Horizontal Line')
    title([SUB.ID,' ', RL, ' Dorr
Classification:',Dorr.Classification],'FontSize',18)
    light

```

```
end

% STEP 3: Storing Information For The User
Dorr_output=Dorr;
```

```
end
```

DU_GetBowling_Final

```
function [Bow_output] = DU_GetBowling_Final(SUB,RL,plot_var)
```

DU_GetBowling - By Sam Mattei 5/31/2023

```
%Inputs:

%SUB - SUBJECT Files for each specimen loaded from DU_AnalyzeHipFemurV2

%RL - Identifier for Right and Left

%plot_var - plotting toggle

%Outputs:

%Bow_output - Bowing structure containing GCBA, GSBA, LSBA, FNSA
%calculations
```

Top Level Code Overview

```
%STEP 1: Identification of the Lesser Trochanter (LT) and loading of appropriate
nodes
%STEP 2: Identification of cross sections needed to calculate all anatomic
%measurements
%STEP 3: Analyze GCBA
%STEP 4: Analyze GSBA
%STEP 5: Analyze LSBA
%STEP 6: Analyze FNSA
%STEP 7: Store Outputs
```

STEP 1: Identification of LT

```
if strcmp(RL,'R')==1
    %pulling the lesser trochanter nodes in the global space
    LT_nodes=(SUB.GEO.A.REGIONS.RFemur.LESS_TROCH)';
    B.nds=SUB.GEO.A.RFemur.nds;
```

```

B.faces=SUB.GEO.G.RFemur.faces;
B.hl=SUB.GEO.G.RFemur.head_element_list;
B.nl=SUB.GEO.G.RFemur.neck_element_list;

AL=SUB.LM.A.RFemur_Dist;
AL_P=SUB.LM.A.RFemur_Prox;

%Sulcus Cut
calcarNDSA=SUB.GEO.A.REGIONS.RFemur.CALCAR_RESECTION;
elseif strcmp(RL,'L')==1
X_Reflect = [1 0 0 0; 0 -1 0 0; 0 0 1 0; 0 0 0 1];
B.faces = SUB.GEO.G.LFemur.faces;
B.hl=SUB.GEO.G.LFemur.head_element_list;
B.nl=SUB.GEO.G.LFemur.neck_element_list;
B.nds = (X_Reflect* SUB.GEO.A.LFemur.nds)';
AL = (X_Reflect*SUB.LM.A.LFemur_Dist)';
AL_P= (X_Reflect*SUB.LM.A.LFemur_Prox)';
%pulling the lesser trochanter nodes in the global space
LT_nodes=(X_Reflect*SUB.GEO.A.REGIONS.LFemur.LESS_TROCH)';

%Sulcus Cut
calcarNDSA=(X_Reflect* SUB.GEO.A.REGIONS.LFemur.CALCAR_RESECTION)';

end

```

Step 2: Identification of the Slices needed to find the sagittal and coronal bowing

```

%pulling the height of the distal lesser trochanter
[distal_LT,max_indLT]=min(LT_nodes(:,4));
distal_LT=distal_LT-10; %offsetting the minimum point by 10mm because the other
code struggles with identifying the full surface
%identifying the midpoint of the condyles
midpoint_DistalCondyles=[1,0.5.*(AL(1,2:4) + AL(4,2:4))]; %finding the midpoint
between the distal epicondyles

%getting the appropriate structure ready for intersectPlaneSurf
fv.vertices=B.nds(:,2:4);
fv.faces=B.faces(:,2:4);

%finding the exact line of intersection with the femoral mesh
lin1C=intersectPlaneSurf(fv,[midpoint_DistalCondyles(1,2:3)
midpoint_DistalCondyles(:,4)+75],[0 0 1]);
if numel(lin1C)>1
MaxResults = cellfun(@(x)size(x,2), lin1C);
[~,indToKeepMax]=max(MaxResults);

```

```

        lin1=lin1C{indToKeepMax}';
    else
        lin1=lin1C{1}';
    end
    lin2C=intersectPlaneSurf(fv,[midpoint_DistalCondyles(1,2:3)
    midpoint_DistalCondyles(:,4)+125],[0 0 1]);
    if numel(lin2C)>1
        MaxResults = cellfun(@(x)size(x,2), lin2C);
        [~,indToKeepMax]=max(MaxResults);
        lin2=lin2C{indToKeepMax}';
    else
        lin2=lin2C{1}';
    end
    lin3C=intersectPlaneSurf(fv,LT_nodes(max_indLT,2:4),[0 0 1]);
    if numel(lin3C)>1
        MaxResults = cellfun(@(x)size(x,2), lin3C);
        [~,indToKeepMax]=max(MaxResults);
        lin3=lin3C{indToKeepMax}';
    else
        lin3=lin3C{1}';
    end
    lin4C=intersectPlaneSurf(fv,[LT_nodes(max_indLT,2:3) distal_LT-50],[0 0 1]);
    if numel(lin4C)>1
        MaxResults = cellfun(@(x)size(x,2), lin4C);
        [~,indToKeepMax]=max(MaxResults);
        lin4=lin4C{indToKeepMax}';
    else
        lin4=lin4C{1}';
    end

    %storing the lines of intersection
    lin={lin1,lin2,lin3,lin4};

```

STEP 3: Analyze Global Coronal Bowing (GCBA)

```

%Finding the center of each identified line
for i=1:4
    %storing
    lin{2,i}=mean(lin{1,i});
    lin{3,i}=[min(lin{1,i}(:,1)) max(lin{1,i}(:,1)) min(lin{1,i}(:,2))
    max(lin{1,i}(:,2))];
end

%calculating 2D unit vectors and extending them
uvProx= ([lin{2,3}(:,1),lin{2,3}(:,3)]-[lin{2,4}(:,1),lin{2,4}(:,3)])/
norm([lin{2,3}(:,1),lin{2,3}(:,3)]-[lin{2,4}(:,1),lin{2,4}(:,3)]);
tv=[lin{2,3}(:,1),lin{2,3}(:,3)];
t11=[tv-uvProx*1000;tv+uvProx*1000];

```

```

uvDist= ([lin{2,2}(:,1),lin{2,2}(:,3)]-[lin{2,1}(:,1),lin{2,1}(:,3)])/
norm([lin{2,2}(:,1),lin{2,2}(:,3)]-[lin{2,1}(:,1),lin{2,1}(:,3)]);
tv=[lin{2,1}(:,1),lin{2,1}(:,3)];
t12=[tv-uvDist*1000;tv+uvDist*1000];
%finding the intersection between the 2 lines
[xinter,~]=polyxpoly(t11(:,1),t11(:,2),t12(:,1),t12(:,2));

a=uvProx;
b=uvDist;

%finding the geometric mean of the slice that intersects with
if xinter>=0 %lateral bowing
    %Using the dot product to calculate the angle of bowing
    angleCoronalBow=acosd(dot(a,b)/ (norm(a)*norm(b)));
else %Medial bowing
    %Using the dot product to calculate the angle of bowing
    angleCoronalBow=-acosd(dot(a,b)/ (norm(a)*norm(b)));
end

if plot_var==1
    fig1=figure();set(gcf,'color',[1 1 1])
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.n1,2:4),'FaceColor',[227/256
218/256 201/256],'FaceLighting',' gouraud','EdgeColor','none','FaceAlpha',0.5);hold
on
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.h1,2:4),'FaceColor',[227/256
218/256 201/256],'FaceLighting',' gouraud','EdgeColor','none','FaceAlpha',0.5);hold
on

plot3(lin{2,1}(:,1),lin{2,1}(:,2),lin{2,1}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

plot3(lin{2,2}(:,1),lin{2,2}(:,2),lin{2,2}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

plot3(lin{2,3}(:,1),lin{2,3}(:,2),lin{2,3}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

plot3(lin{2,4}(:,1),lin{2,4}(:,2),lin{2,4}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on
    %plot3(t11_P(:,1),t11_P(:,2),t11_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m')
    %plot3(t12_P(:,1),t12_P(:,2),t12_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m')
    title([SUB.ID,' ', RL, ' Coronal Bowing:'],'FontSize',18)
    axis equal; axis off
    view([-180 0]) %Frontal
    light
end

```

STEP 4: Analyze Global Sagittal Bowing (GSBA)

```
%calculating 2D unit vectors and extending them
uvProx= ([lin{2,3}(:,2),lin{2,3}(:,3)]-[lin{2,4}(:,2),lin{2,4}(:,3)])/
norm([lin{2,3}(:,2),lin{2,3}(:,3)]-[lin{2,4}(:,2),lin{2,4}(:,3)]);
tv=[lin{2,3}(:,2),lin{2,3}(:,3)];
t11=[tv-uvProx*1000;tv+uvProx*1000];
uvDist= ([lin{2,2}(:,2),lin{2,2}(:,3)]-[lin{2,1}(:,2),lin{2,1}(:,3)])/
norm([lin{2,2}(:,2),lin{2,2}(:,3)]-[lin{2,1}(:,2),lin{2,1}(:,3)]);
tv=[lin{2,1}(:,2),lin{2,1}(:,3)];
t12=[tv-uvDist*1000;tv+uvDist*1000];
%finding the intersection between the 2 lines
[xinter,yinter]=polyxpoly(t11(:,1),t11(:,2),t12(:,1),t12(:,2));

%Proximal Vector
a=uvProx;
b=uvDist;

%Using the dot product to calculate the angle of bowing
angleSagittalBow=acosd(dot(a,b)/(norm(a)*norm(b)));

if plot_var==1
    fig2=figure();set(gcf,'color',[1 1 1])
    %updating figures
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.n1,2:4),'FaceColor',[227/256
218/256 201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5);hold
on
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.h1,2:4),'FaceColor',[227/256
218/256 201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5);hold
on

    plot3(lin{2,1}(:,1),lin{2,1}(:,2),lin{2,1}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    plot3(lin{2,2}(:,1),lin{2,2}(:,2),lin{2,2}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    plot3(lin{2,3}(:,1),lin{2,3}(:,2),lin{2,3}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    plot3(lin{2,4}(:,1),lin{2,4}(:,2),lin{2,4}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on
    %plot3(t11_P(:,1),t11_P(:,2),t11_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m')
    %p4=plot3(t12_P(:,1),t12_P(:,2),t12_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m');
```

```

axis equal; axis off
title([SUB.ID, ' ', RL, ' Sagittal Bowing:'], 'FontSize', 18)
light
view([-90 0]) %Frontal
end

```

STEP 5: Analyze Local Proximal Sagittal Bowing (LSBA)

```

%finding the unit vector and extending the vector from the proximal
%points identified using Lee et al.'s method
uvProx_LPSB= ([lin{2,3}(:,2),lin{2,3}(:,3)]-[lin{2,4}(:,2),lin{2,4}(:,3)])/
norm([lin{2,3}(:,2),lin{2,3}(:,3)]-[lin{2,4}(:,2),lin{2,4}(:,3)]);
tv_LPSB=[lin{2,3}(:,2),lin{2,3}(:,3)];
t11_LPSB=[tv_LPSB-uvProx_LPSB*1000;tv_LPSB+uvProx_LPSB*1000];

%finding the second vector, which will be a vertical line from the
%femoral head center
uv2_LPSB=[0 -1];
tv_LPSB=[AL_P(1,3),AL_P(1,4)];
t12_LPSB=[tv_LPSB-uv2_LPSB*1000;tv_LPSB+uv2_LPSB*1000];

%finding the point of intersection between the two points
[xinter,yinter]=polyxpoly(t11_LPSB(:,1),t11_LPSB(:,2),t12_LPSB(:,1),t12_LPSB(:,2))
;

%cropping each vector and identifying the dot product
a=uvProx_LPSB;
b=[0 1];

%Using the dot product to calculate the angle of bowing
angleLocalizedSagittalBow=acosd(dot(a,b)/(norm(a)*norm(b)));

if acosd(dot(a,b)/(norm(a)*norm(b)))>=45
disp('1')
end

if plot_var==1
fig3=figure();set(gcf,'color',[1 1 1])
%plotting anatomic bones
patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.n1,2:4),'FaceColor',[227/256
218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5,'Clipping','off');hold on
patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.h1,2:4),'FaceColor',[227/256
218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5,'Clipping','off');hold on

```



```

%Plotting the Proximal Points

plot3(lin{2,3}(:,1),lin{2,3}(:,2),lin{2,3}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

plot3(lin{2,4}(:,1),lin{2,4}(:,2),lin{2,4}(:,3),'LineStyle','none','Marker','o','M
arkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on
%Plotting Anatomic head Center

plot3(AL_P(1,2),AL_P(1,3),AL_P(1,4),'LineStyle','none','Marker','o','MarkerFaceCol
or','m','MarkerEdgeColor','k','MarkerSize',14); hold on
%plotting each vector
plot3(t11_LPSB_P(:,1),t11_LPSB_P(:,2),t11_LPSB_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m')
plot3(t12_LPSB_P(:,1),t12_LPSB_P(:,2),t12_LPSB_P(:,3),'LineStyle','-
','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','Mar
kerSize',14,'Color','m')

axis equal; axis off
title([SUB.ID, ' ', RL, ' Localized Sagittal Bowing:'],'FontSize',18)
light
view([-90 0]) %Frontal
end

```

STEP 6: Analyze Femoral-Neck-Shaft-Angle (FNSA)

```

%finding the intersection point of the line
%Proximal vector
uvProx_FNA= ([lin{2,4}(:,1),lin{2,4}(:,3)]-[lin{2,3}(:,1),lin{2,3}(:,3)])/
norm([lin{2,4}(:,1),lin{2,4}(:,3)]-[lin{2,3}(:,1),lin{2,3}(:,3)]);
tv=[lin{2,3}(:,1),lin{2,3}(:,3)];
t11_FNA=[tv-uvProx_FNA*1000;tv+uvProx_FNA*1000];

%Femoral Neck Axis Vector
centroidSulcusCutAnatomic=mean(calcarNDSA(:,2:4)); %finding the centroid of the
sulcus cut in anatomic space
uvFNA= (AL_P(1,2:3)-centroidSulcusCutAnatomic(1,[1,3]))/ norm (AL_P(1,2:3)-
centroidSulcusCutAnatomic(1,[1,3])); %finding the neck unit vector
t12_FNA= [centroidSulcusCutAnatomic(1,[1,3])+ uvFNA*1000;
centroidSulcusCutAnatomic(1,[1,3])-uvFNA*1000] ; %extending the vector
%finding the point of intersection between the two points
[xinter,yinter]=polyxpoly(t11_FNA(:,1),t11_FNA(:,2),t12_FNA(:,1),t12_FNA(:,2));

%cropping the vectors
a=[(lin{2,3}(:,1)-xinter),(lin{2,3}(:,3)-yinter)];
b=[(AL_P(1,1)-xinter),(AL_P(1,3)-yinter)];

```

```

%finding the Femoral Neck Shaft Angle (FNSA)
angleFemNeckShaft=norm(acosd(dot(a,b)/ (norm(a)*norm(b))));

if plot_var==1
    %Initializing plotting vectors
    uvProx_FNA=(lin{2,4}-lin{2,3})/ norm(lin{2,4}-lin{2,3});
    tv_LPSB=[lin{2,3}(:,1),lin{2,3}(:,2),lin{2,3}(:,3)];
    t11_LPSB_P=[tv_LPSB-uvProx_FNA*250;tv_LPSB+uvProx_FNA*250];
    uvFNA= (AL_P(1,2:4)-centroidSulcusCutAnatomic(1,:)) / norm (AL_P(1,2:4)-
centroidSulcusCutAnatomic(1,:)); %finding the neck unit vector
    t12_LPSB_P=[centroidSulcusCutAnatomic(1,:)-
uvFNA*100;centroidSulcusCutAnatomic(1,:)+uvFNA*250];

    fig4=figure();set(gcf,'color',[1 1 1])
    %plotting anatomic bones
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.n1,2:4),'FaceColor',[227/256
218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5,'Clipping','off');hold on
    patch('Vertices',B.nds(:,2:4),'Faces',B.faces(B.h1,2:4),'FaceColor',[227/256
218/256
201/256],'FaceLighting','gouraud','EdgeColor','none','FaceAlpha',0.5,'Clipping','off');hold on
    %plotting the Proximal Points

    plot3(lin{2,3}(:,1),lin{2,3}(:,2),lin{2,3}(:,3),'LineStyle','none','Marker','o','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    plot3(lin{2,4}(:,1),lin{2,4}(:,2),lin{2,4}(:,3),'LineStyle','none','Marker','o','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on
    %Plotting Anatomic head Center

    plot3(AL_P(1,2),AL_P(1,3),AL_P(1,4),'LineStyle','none','Marker','o','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    plot3(centroidSulcusCutAnatomic(1,1),centroidSulcusCutAnatomic(1,2),centroidSulcusCutAnatomic(1,3),'LineStyle','none','Marker','o','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14); hold on

    %plotting each vector
    plot3(t11_LPSB_P(:,1),t11_LPSB_P(:,2),t11_LPSB_P(:,3),'LineStyle','-','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14,'Color','m')
    plot3(t12_LPSB_P(:,1),t12_LPSB_P(:,2),t12_LPSB_P(:,3),'LineStyle','-','Linewidth',2.0,'Marker','none','MarkerFaceColor','m','MarkerEdgeColor','k','MarkerSize',14,'Color','m')
    axis equal; axis off
    title([SUB.ID,' ', RL, ' Femoral Neck Shaft Angle:'],'FontSize',18)
    light

```

```
view([-180 0]) %Frontal  
end
```

STEP 7: Storing Return Variables

```
Bow_output.SagittalAngle=angleSagittalBow;  
Bow_output.LocalSagittalAngle=angleLocalizedSagittalBow;  
Bow_output.CoronalAngle=angleCoronalBow;  
Bow_output.FemoralNeckShaftAngle=angleFemNeckShaft;  
end
```

Appendix B: Implantation Algorithm

The code presented in this appendix (DU_ImplantFemur_Final) was used by the implantation algorithm used for the calculation and verification of 2500 LHC perturbations to each size of femoral stem from the nominal position. This code was ran on one subject and right and left legs were handled separately. The nominal position of the stem was verified within this algorithm as well as all created transformation matrices. This code was indexed through each specimen and the appropriate structures were updated as the code progressed.

DU_ImplantFemur_Final

```
function [PATIENT,STOP_LOOP] =
DU_ImplantFemur_Final(SUB,RL,HIP,STEM_SIZE_input,combs_DOE_input,PATIENT,varargin)

%DU_ImplantFemur_Final - Code By Sam Mattei 5/15/2023
%This code takes inputs from DU_GetIntercorticalMesh and
%Du_GetCalcar_Final as well as Du_AnalyzeHipFemurV2 and performs virtual
%implantations using a specified numbre of Latin Hypercube Perturbations to
%the stem. Perturbations are described from the "nominal" position of the
%implant and are used to quantify motion of the femoral stem about its most
%distal point. This code also uses the parallel computing toolbox to
%increase efficiency of calculations performed.

%this function has dependencies on the function "Fast_Point2Trimesh" By
%Thor Andreassen to complete overclosure calculations, there are helper
%functions to verify the directions of surface normals on eash mesh though
%the additional functions included by Andreassen et al. can calculate
%these.

%Below is a list of all inputs and outputs

%Inputs:

%SUB - SUBJECT Files for each specimen containing information from
%DU_AnalyzeHipFemurV2, DU_GetIntercorticalMesh, and DU_GetCalcar_Final

%RL - Right vs Left Identifeir

%HIP - structure containing all landmarked information about the femoral
%stem, reduced patches of all implantable sizes, and nodes and connectivity
```

```

%lists

%STEM_SIZE_input - Input parameter that notates the size of femoral stem
%used for the desired LHC implantation

%combs_DOE_input - LHC perturbations nx6. where n is the number of
%combinations of the [ADAB, FE, IE, ML, AP,SI] Directions of stem
%translation in the femoral stem coordinate system

%PATIENT - the structure for the data to be appended to, specified as a
%cell array {n,1} represents where the implatnation data will appended to and
{n,2} is the subject identifier

%Outputs:

%PATIENT - Updated array containing newly created subject specific and RL
%specific implatnation data

%STOP_LOOP - toggles the stopping of implantation loop based off of
%overclosure tolerance of nominal stem position. 0 or 1 for toggle.

```

Top Level Code Overview

```

%STEP 1: Assigning of Optional Parameters and Initialization of Structures
%STEP 2: Loading Right vs Left Subject Boney Geometry, ICC and ICDM meshes
%STEP 3: Using SVD Define the TM from Local Anatomic Coordinate System to Implant
Coordinate System
%STEP 4: Transforming Nodes From Local Anatomic Coordinate System to
%Implant Coordinate System/ Establishing Other Transformation Matrices
%STEP 5: Calculating Surface Normals for Stem, Identifying Top Nodes on Slice for
Verification
%STEP 6: Verify Nominal Implant Position
%STEP 7: Latin Hypercube (LHC) Perturbation and Implantation

```

STEP 1: Assigning of Optional Parameters and Initialization of Structures

```

implantOriginSpecifier_Input='DistalTip';
LMIdentifierInput='FH';
numDiaphysisSliceInput=20;
user_threshold=0.25; %mm of overclosure
plot_var=1;
input_percent=[0.4,0.2]; %percentage of implant height to analyze
% Loading optional arguments
while ~isempty(varargin)
    switch lower(varargin{1})
        case 'implantorigin'
            implantOriginSpecifier_Input = varargin{2};

```

```

case 'lmidentifier'
    LMIdentifierInput = varargin{2};
case 'diaphysislice'
    numDiaphysisSliceInput = varargin{2};
case 'plottoggle'
    plot_var = varargin{2};
case 'pt2trithresh'
    user_threshold=varargin{2};
otherwise
    error(['Unexpected option: ' varargin{1}])
end
varargin(1:2) = [];
end
STEM_SIZE_CONV={'8 STD135'; '9 STD135'; '10 STD135'; '11 STD135'; '12 STD135'; '13
STD135'; '14 STD135'; '15 STD135'; '16 STD135'; '18 STD135'; '20 STD135'} ;

implant_elements = HIP{STEM_SIZE_input}.ELS;
fitDiaphysisColor=zeros(length(combs_DOE_input),1);
fitCalcarColor=zeros(length(combs_DOE_input),1);
implantHeadNodeDOE=zeros(length(combs_DOE_input),4);
implantHeadNORMDOE=zeros(length(combs_DOE_input),3);

```

STEP 2: Loading Right vs Left Subject Boney Geometry, ICC and ICDM

meshes

```

if strcmp(RL,'R')==1
    B.nds = SUB.GEO.A.RFemur.nds; % B.nds and B.faces which define the femoral
patch in the femoral-anatomic coordinate system
    B.faces = SUB.GEO.G.RFemur.faces;
    AL = SUB.LM.A.RFemur_Prox;
    TM_AtoI = SUB.TM.RFemur.AtoI;
    TM_AtoS = SUB.TM.RFemur.AtoS;
    neck_element_list=SUB.GEO.G.RFemur.neck_element_list;
    head_element_list=SUB.GEO.G.RFemur.head_element_list;
    calcar_resection = SUB.GEO.A.REGIONS.RFemur.CALCAR_RESECTION;
    %calcar
    calcar.nds=SUB.GEO.G.RFemur.calcarNodes;
    calcar.ndsA=(inv(SUB.TM.F.GtoAPR)*calcar.nds)'; %transforming to the anatomic
space for visualization
    calcar.ndsS=(TM_AtoS*calcar.ndsA)';
    %diaphysis
    diaphysis.nds=SUB.GEO.G.RFemur.diaphysisNodes;
    diaphysis.elms=SUB.GEO.G.RFemur.diaphysisFaces;
    diaphysis.elmsToCrop=SUB.GEO.G.RFemur.diaphysisElmsCrop;
    diaphysis.ndsA=(inv(SUB.TM.F.GtoAPR)*diaphysis.nds)';%transforming to
anatomic space

```

```

%Loading the Correspondence List for Further Cropping
CL=SUB.GEO.G.RFemur.CLDIA_R;

else
X_Reflect = [1 0 0 0; 0 -1 0 0; 0 0 1 0; 0 0 0 1];
% Reflect the left knee geometry into a right knee
B.nds = SUB.GEO.A.LFemur.nds;
B.faces = SUB.GEO.G.LFemur.faces;
B.nds = (X_Reflect*B.nds)';
AL = (X_Reflect*SUB.LM.A.LFemur_Prox)';
TM_AtoI = SUB.TM.LFemur.AtoI;
TM_AtoS = SUB.TM.LFemur.AtoS;
neck_element_list=SUB.GEO.G.LFemur.neck_element_list;
head_element_list=SUB.GEO.G.LFemur.head_element_list;
calcar_resection = (X_Reflect*SUB.GEO.A.REGIONS.LFemur.CALCAR_RESECTION)';
%calcar
calcar.nds=SUB.GEO.G.LFemur.calcarNodes;
calcar.ndsA=(inv(SUB.TM.F.GtoAPL)*calcar.nds)'; %Transforming to the anatomic
space for visualization
calcar.ndsA=(X_Reflect*calcar.ndsA)';
calcar.ndsS=(TM_AtoS*calcar.ndsA)';
%diaphysis
diaphysis.nds=SUB.GEO.G.LFemur.diaphysisNodes;
diaphysis.elms=SUB.GEO.G.LFemur.diaphysisFaces;
diaphysis.elmsToCrop=SUB.GEO.G.LFemur.diaphysisElmsCrop;
diaphysis.ndsA=(inv(SUB.TM.F.GtoAPL)*diaphysis.nds)'; %transforming to
anatomic space
diaphysis.ndsA=(X_Reflect*diaphysis.ndsA)'; %Reflecting L legs to be Right

%Loading the Correspondence List for Further Cropping
CL=SUB.GEO.G.LFemur.CLDIA_L;
end

```

STEP 3: Using SVD Define the TM from Local Anatomic Coordinate System to Implant Coordinate System

```

%getting the unique indices of the diaphysis mesh
diaphysisSliceHeights=linspace(mean(diaphysis.nds(:,4))-
50,max(diaphysis.nds(:,4)),numDiaphysisSliceInput);

%initializing
slice_indicator=zeros(numDiaphysisSliceInput,1);
temp_diaphysisNodesToCheck=[];
for counti=1:numDiaphysisSliceInput
%finding the mimimum difference between nodes and the slices and using
%this to keep specific indices, for some of thes the slices will be the

```

```

    %same
    [~,ind1_slice]=min(abs(diaphysis.nds(:,4)- diaphysisSliceHeights(counti)));
    %since the inputs for this function have been smoothed and interpolated
    %this should be a uniform process
    slice_indicator(counti,1)=diaphysis.nds(ind1_slice,4);
    %pulling the smoothed nodes indices

temp_diaphysisNodesToCheck{counti,1}=find(diaphysis.nds(:,4)>=slice_indicator(counti,1)-0.5 & diaphysis.nds(:,4)<=slice_indicator(counti,1) +0.5);
end

%pulling the implant dimensions for adequate transformations
implant_height=norm(HIP{STEM_SIZE_input}.LENGTH);
centroidDiaphysis=zeros(numDiaphysisSliceInput,4);
%looping through each slice and extracting the centroid and storing them
for countSlices=1:numDiaphysisSliceInput
    %Pulling the appropriate indices of the nodes
    nodes_slice_ind=temp_diaphysisNodesToCheck{countSlices};
    %extracting the diaphysis slices
    centroidDiaphysis(countSlices,:)=mean(diaphysis.ndsA(nodes_slice_ind,:),1);
end
%fitting a line of best fit to the nodes using SVD
centroidDiaphysis_ave=mean(centroidDiaphysis(:,2:4),1);
dx=bsxfun(@minus,centroidDiaphysis(:,2:4),centroidDiaphysis_ave);
C=(dx'*dx)/(size(centroidDiaphysis,1)-1); % variance-covariance matrix
of X
[R,D,V]=svd(C,0); % singular value decomposition of C; C=R*D*R'
uvSI=V(:,1);
x=dx*R(:,1); % project residuals on R(:,1)
x_min=min(x);
x_max=max(x);
dx=x_max-x_min;
Xa=(x_min-0.05*dx)*R(:,1)' + centroidDiaphysis_ave;
Xb=(x_max+0.05*dx)*R(:,1)' + centroidDiaphysis_ave;
X_end=[1 Xa;1 Xb];
if strcmp(LMIdentifierInput,'FH')==1
    %Calculate temp TM based on implant coordinate system
    vSI = (Xa - Xb)/norm((Xa - Xb)); %z-axis along femoral anatomic axis
    vtemp = Xa*-1/norm(Xa); %temp axis forming anterior plane
    vAP = cross(vtemp,vSI); vAP=vAP/norm(vAP); %y-axis perpendicular to anterior
plane
    vML=cross(vAP,vSI); vML=vML/norm(vML); %Re-cross to get x-axis
    [~,point1]=min(calcar.ndsA(:,4)); %pulling the indices of the medial sulcus
point on the calcar cut
    % Assesemble TM
    TM_AtoI= eye(4,4);
    TM_AtoI(2:4,2:4) = [vML' vAP' vSI'];
    TM_AtoI(2:4,1) =Xa+(vSI)*implant_height; %Setting the starting point to be the
top slice of the diaphysis with the implant centered on it
elseif strcmp(LMIdentifierInput,'MC')==1

```



```

[~,point1]=min(calcar.ndsA(:,4)); %pulling the indices of the medial sulcus
point on the calcar cut
%Calculate temp TM based on implant coordinate system
vSI = (Xa - Xb)/norm((Xa - Xb)); %z-axis along femoral anatomic axis
vtemp = (calcar.ndsA(point1,2:4)-Xa)/norm(calcar.ndsA(point1,2:4)-Xa); %temp
axis forming anterior plane
VAP = cross(vtemp,vSI); vAP=vAP/norm(vAP); %y-axis perpendicular to anterior
plane
VML=cross(vAP,vSI); vML=vML/norm(vML); %Re-cross to get x-axis
% Assemble TM
TM_AtoI= eye(4,4);
TM_AtoI(2:4,2:4) = [vML' vAP' vSI'];
TM_AtoI(2:4,1) = Xa+(vSI)*implant_height; %Setting the starting point to be
the top slice of the diaphysis with the implant centered on it
end

```

STEP 4: Transforming Nodes From Local Anatomic Coordinate System to Implant Coordinate System/ Establishing Other Transformation Matrices

```

temp_bone_nodes = (inv(TM_AtoI)*B.nds')';
calcar_resection_I = (inv(TM_AtoI)*calcar_resection')';
diaphysis_nodes_I=(inv(TM_AtoI)*diaphysis.ndsA')';
calcar_IC_I=(inv(TM_AtoI)*calcar.ndsA')';
AL_I=(inv(TM_AtoI)*AL')';
neck_axis = (inv(TM_AtoI)*[1 0 0 0; AL(9,:)])';
X_end_I=(inv(TM_AtoI)*X_end')';

%creating the larger transformation matrix from implant to sulcus for right
%and left bones
if strcmp(RL,'R')==1
    TM_ItoS=SUB.TM.F.GtoSR*SUB.TM.F.GtoAPR*TM_AtoI;
    %Updating PATIENT structure
    PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.TM_ItoS=TM_ItoS;
    PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.TM_AtoI=TM_AtoI;
    PATIENT.R.CORAIL135STD{STEM_SIZE_input,2}=STEM_SIZE_CONV{STEM_SIZE_input};
elseif strcmp(RL,'L')==1
    TM_ItoS=SUB.TM.F.GtoSL*SUB.TM.F.GtoAPL*inv(X_Reflect)*TM_AtoI;
    %Updating PATIENT structure
    PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.TM_ItoS=TM_ItoS;
    PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.TM_AtoI=TM_AtoI;
    PATIENT.L.CORAIL135STD{STEM_SIZE_input,2}=STEM_SIZE_CONV{STEM_SIZE_input};
end

%finding the magnitude to translate the stem down to establish nominal
%position
heightToMedialCalcar=abs(HIP{STEM_SIZE_input}.MEDIAL_ALIGN(4)-
calcar_IC_I(point1,4));

```

```

%physically translating the stem into the position where the stem has its
%most medial point aligned with the calcar position
implant_nodes=[ ones(size(HIP{STEM_SIZE_input}.NDS,1),1) ,
HIP{STEM_SIZE_input}.NDS(:,2:3),HIP{STEM_SIZE_input}.NDS(:,4)-
heightToMedialCalcar];
implantHead=[1,HIP{STEM_SIZE_input}.TAPER_ORIGIN(:,2:3),HIP{STEM_SIZE_input}.TAPER
_ORIGIN(:,4)-heightToMedialCalcar];
implantDistalTip=[1,HIP{STEM_SIZE_input}.DISTAL_TIP(:,2:3),HIP{STEM_SIZE_input}.DI
STAL_TIP(:,4)-heightToMedialCalcar];
%defining the implant origin in the new space
if strcmp(implantOriginSpecifier_Input,'FemHead')==1
    implantOriginOptimize=implantHead;
elseif strcmp(implantOriginSpecifier_Input,'DistalTip')==1
    implantOriginOptimize=implantDistalTip;
end
%Pulling the heights of the implant to analyze from its nominal position
heightsToAnalyze=[min(implant_nodes(:,4))+input_percents(1)*(max(implant_nodes(:,4
))-min(implant_nodes(:,4))) ;
min(implant_nodes(:,4))+input_percents(2)*(max(implant_nodes(:,4))-
min(implant_nodes(:,4)))];

temp1.vertices=diaphysis_nodes_I(:,2:4);
temp1.faces=diaphysis.elms(:,2:4);

```

STEP 5: Calculating Surface Normal for Stem, Identifying Top Nodes on

Slice for Verification

```

p_dia=patch('vertices',diaphysis_nodes_I(:,2:4),'Faces',diaphysis.elms(diaphysis.e
lmsToCrop,2:4),'visible','off');
rpDIA.vertices=p_dia.vertices;rpDIA.faces=p_dia.faces;
%storing as inputs for Thor's code
input.nodes=rpDIA.vertices;
input.faces=rpDIA.faces;

%checking the surface normals
input=checkSurfaceNormals(input);
%creating a KD Tree for the diaphysis for nodal overclosures
tree1 = KDTreeSearcher(input.face_mean_nodes);
input.tree_model=tree1;

%finding the nodes of the top slice based off of connectivity, should be
%less than the other nodes of the mesh
temp=cell2mat(cellfun(@size,CL,'UniformOutput',false));
indToFindNodesOnTopSlice=find(temp(:,2)==3);
for cind=1:length(indToFindNodesOnTopSlice)
    indToFindNodesOnTopSlice(cind,2)=length(CL{indToFindNodesOnTopSlice(cind)});

```

```

%pulling the numbers of elements
end

%pulling free surface nodes
[minPT,minPTIND]=min(diaphysis_nodes_I(indToFindNodesOnTopSlice(:,1),4));
[~,maxPTIND]=max(diaphysis_nodes_I(indToFindNodesOnTopSlice(:,1),4));
nodesDiaphysisSlice_I=diaphysis_nodes_I(indToFindNodesOnTopSlice(:,1),:);
p0TopSlice=nodesDiaphysisSlice_I(minPTIND,2:4);

%Calculating the top face normals
if maxPTIND~=3
    NTopSlice= cross((nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(maxPTIND,2:4) ) ,(nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(3,2:4) ) ) ; %getting the normal vector to each plane
based on how the CS was defined??
elseif minPT~=3
    NTopSlice= cross((nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(maxPTIND,2:4) ) ,(nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(5,2:4) ) ) ; %getting the normal vector to each plane
based on how the CS was defined??
else
    NTopSlice= cross((nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(maxPTIND,2:4) ) ,(nodesDiaphysisSlice_I(minPTIND,2:4) -
nodesDiaphysisSlice_I(7,2:4) ) ) ; %getting the normal vector to each plane
based on how the CS was defined??
end

%verifying the output of the function
inputTemp.vertices=implant_nodes(:,2:4);
inputTemp.faces=implant_elements(:,2:4);
tempL=intersectPlaneSurf(inputTemp,p0TopSlice,NTopSlice);

if numel(tempL)>1
    MaxResults = cellfun(@(x)size(x,2), tempL);
    [~,indToKeepMax]=max(MaxResults);
    tempL=tempL{indToKeepMax}';
else
    tempL=tempL{1}';
end
[~,max_ind]=max(tempL(:,3));
%finding if the point is within the plane specified
tempDP=round(dot((tempL(max_ind(1),1:3)) -p0TopSlice(1,1:3) ,NTopSlice));
%flipping normal direction based on sign convention of Z
%direction
if tempDP~=0
    NTopSlice(3)=NTopSlice(3)*-1;
end
end

```

STEP 6: Verify Nominal Implant Position

```

%need reduced diaphysis mesh
indImplantToTestDIANOM=implant_nodes(implant_nodes(:,4)>=min(implantDistalTip(1,4)
) & implant_nodes(:,4)<=minPT,:);
%Combining to generate the input points
ptsDIANOM=[indImplantToTestDIANOM(:,2:4);tempL];

%Also need
[distancesDIANOM,~,~]=fastPoint2TriMesh(input,ptsDIANOM,0);
percentOVC=length(find(distancesDIANOM>0.5))/length(distancesDIANOM) * 100;

%breaking loop conditions
if max(distancesDIANOM)>=0.5 && percentOVC>=5.0
    %setting condition to stop loop
    STOP_LOOP=1;
else
    STOP_LOOP=0;
end

% Calculating SI LHC perturbations Based on SI height
SI_Range=AL_I(1,4)-implantHead(1,4);

```

STEP 7: Latin Hypercube (LHC) Perturbation and Implantation

```

combs_DOE_input(1,:)=[0 0 0 0 0 0];
if STOP_LOOP==0

    calcarNodesToCheck=find(calcar_IC_I(:,4)<=
min(calcar_IC_I(:,4))+0.85*(max(calcar_IC_I(:,4))-min(calcar_IC_I(:,4)))));
    p0_CA=calcar.ndss(point1,2:4);
    tempOutput={};
    parfor countCombinations=1:size(combs_DOE_input,1)

```

Perturbing Implant

```

%Displaying Implantation Information To The User
if mod(countCombinations,size(combs_DOE_input,1))==0
    disp([SUB.ID, ' implanted ' num2str(countCombinations) ' of '
num2str(size(combs_DOE_input,1))])
end
%Creating a Transformation Matrix using the Perturbations of the
%Implant's nominal position
rotateMatrixADAB=[cosd(combs_DOE_input(countCombinations,1)) 0 -
sind(combs_DOE_input(countCombinations,1)); 0 1 0;
sind(combs_DOE_input(countCombinations,1)) 0
cosd(combs_DOE_input(countCombinations,1))]; % Varus-Valgus rotation

```

```

        rotateMatrixFE= [1 0 0; 0 cosd(combs_DOE_input(countCombinations,2)) -
        sind(combs_DOE_input(countCombinations,2)); 0
        sind(combs_DOE_input(countCombinations,2))
        cosd(combs_DOE_input(countCombinations,2)) ; ];% Flexion-Extension Rotation
        rotateMatrixIE=[cosd(combs_DOE_input(countCombinations,3)) -
        sind(combs_DOE_input(countCombinations,3)) 0;
        sind(combs_DOE_input(countCombinations,3))
        cosd(combs_DOE_input(countCombinations,3)) 0; 0 0 1]; % Abduction-Adduction
Rotation
        translateML=combs_DOE_input(countCombinations,4); %Medial-Lateral
translations
        translateAP=combs_DOE_input(countCombinations,5); %Anterior-Posterior
translations
        translateSI=combs_DOE_input(countCombinations,6)*SI_Range; %Superior-
Inferior translations

        %establishing an Euler XYZ rotation matrix and integrating to the
        %overall transformation matrix
        RM= rotateMatrixIE*rotateMatrixFE*rotateMatrixADAB;
        TM_ItoIDOE=eye(4,4);
        TM_ItoIDOE(2:4,2:4)=RM;
        TM_ItoIDOE(2:4,1)=[translateML,translateAP,translateSI];

        TM{countCombinations,1}=TM_ItoIDOE; %Storing for later

        %Transforming the implant nodes and implanted head center to the
        %perturbed space
        IO=implantOriginOptimize;
        implantNodes=[ones(size(implant_nodes,1),1) implant_nodes(:,2:4)-
        IO(1,2:4)];%translating to global origin
        implantNodesDOE=(TM_ItoIDOE*implantNodes)'; %transforming the nodes in
the XYZ Euler angle way
        implantNodesDOE=[ones(size(implantNodesDOE,1),1)
        implantNodesDOE(:,2:4)+IO(1,2:4)];%translating from global origin
        implantHeadDOE=[1 implantHead(:,2:4)-IO(:,2:4)];
        implantHeadDOE=(TM_ItoIDOE*implantHeadDOE)';
        implantHeadDOE=[1 implantHeadDOE(:,2:4)+IO(:,2:4)];

        implantHeadNodeDOE(countCombinations,:)=implantHeadDOE;
        implantHeadNORMDOE(countCombinations,:)=
        (RM*HIP{STEM_SIZE_input}.TAPER_NORM)';

```

Evaluating ICDM Mesh Overclosures

```

BREAKWHILE=0;
countMesh=1;
while countMesh<=3 && BREAKWHILE~=1
    %Dealing with different mesh densities

```

```

        if countMesh==1
            imN=[ones(size(HIP{STEM_SIZE_input}.RP005V,1),1),
HIP{STEM_SIZE_input}.RP005V(:,2:3)-IO(1,2:3),HIP{STEM_SIZE_input}.RP005V(:,4)-
IO(1,4)-heightToMedialCalcar];
            imNDOE=(TM_ItoIDOE*imN)'; %transforming the nodes in the XYZ
Euler angle way
            imNDOE=[ones(size(imNDOE,1),1)
imNDOE(:,2:4)+IO(1,2:4)];%translating from global origin
            ie=HIP{STEM_SIZE_input}.RP005E;
            fvr.vertices=imNDOE(:,2:4); fvr.faces=ie(:,2:4);
            %finding the intersection of the top plane with the implant to
            %further increase the nodes of intersection
            lin_temp1=intersectPlaneSurf(fvr,p0TopSlice,NTopSlice);
            %bit of logic to pull the larger boundary to pass to the
            %later functions
            if numel(lin_temp1)>1
                MaxResults = cellfun(@(x)size(x,2), lin_temp1);
                [~,indToKeepMax]=max(MaxResults);
                lin1=lin_temp1{indToKeepMax}';
            else
                lin1=lin_temp1{1}';
            end
        elseif countMesh==2
            imN=[ones(size(HIP{STEM_SIZE_input}.RP05V,1),1),
HIP{STEM_SIZE_input}.RP05V(:,2:3)-IO(1,2:3),HIP{STEM_SIZE_input}.RP05V(:,4)-
IO(1,4)-heightToMedialCalcar];
            imNDOE=(TM_ItoIDOE*imN)'; %transforming the nodes in the XYZ
Euler angle way
            imNDOE=[ones(size(imNDOE,1),1)
imNDOE(:,2:4)+IO(1,2:4)];%translating from global origin
            ie=HIP{STEM_SIZE_input}.RP05E;
            fvr.vertices=imNDOE(:,2:4); fvr.faces=ie(:,2:4);
        elseif countMesh ==3
            imN=[ones(size(implant_nodes,1),1) implant_nodes(:,2:4)-
IO(1,2:4)];%translating to global origin
            imNDOE=(TM_ItoIDOE*imN)'; %transforming the nodes in the XYZ
Euler angle way
            imNDOE=[ones(size(imNDOE,1),1)
imNDOE(:,2:4)+IO(1,2:4)];%translating from global origin
            ie=HIP{STEM_SIZE_input}.ELS;
            fvr.vertices=imNDOE(:,2:4); fvr.faces=ie(:,2:4);
        end

        %cropping the input nodes to the allowed heights
        indImplantToTest=imNDOE(imNDOE(:,4)>=min(implantDistalTip(1,4)) &
imNDOE(:,4)<=minPT,:);
        %combining to generate the input points
        pts=[indImplantToTest(:,2:4);lin1];

        %finding the overclosure amounts

```

```

[distances,~,~]=fastPoint2TriMesh(input,pts,0);

if all(distances<0)
    fitDiaphysisColor(countCombinations,1)=1;
elseif max(distances)<=user_threshold
    fitDiaphysisColor(countCombinations,1)=1;
else
    fitDiaphysisColor(countCombinations,1)=0;
    %disp(num2str(max(distances)))
    BREAKWHILE=1;
end

%if the mesh count is at the final iteration AND the initial
%diaphysis is 1
if countMesh==3 && fitDiaphysisColor(countCombinations,1)==1
    %storing
    l1=intersectPlaneSurf(fvr,[0,0 heightsToAnalyze(1)],[0 0 1]);
    l2=intersectPlaneSurf(fvr,[0,0 heightsToAnalyze(2)],[0 0 1]);
    l3=intersectPlaneSurf(temp1,[0,0 heightsToAnalyze(1)],[0 0 1]);
    l4=intersectPlaneSurf(temp1,[0,0 heightsToAnalyze(2)],[0 0 1]);

    %storing the nodes,elements, RL data, and TM to Sulcus from
    %implant

inputFit.nds=imNDOE;inputFit.elms=ie;inputFit.RL=RL;inputFit.TM_ItoS=TM_ItoS;
    inputFit.NumCombinations=countCombinations;
inputFit.CountImplant=STEM_SIZE_input;
    % Store Implant Head Position and DOE Normal Vector

inputFit.HP=implantHeadDOE;inputFit.HNV=implantHeadNORMDOE(countCombinations,:);

    %storing slice based views
    inputFit.ProxI=[ones(size(l1{1}',1),1) l1{1}'];
    inputFit.DistI=[ones(size(l2{1}',1),1) l2{1}'];
    inputFit.ProxD=[ones(size(l3{1}',1),1) l3{1}'];
    inputFit.DistD=[ones(size(l4{1}',1),1) l4{1}'];
    inputFit.TM_AtoI=TM_AtoI;
    inputFit.sliceDecimals=input_percents;
    %performing the fit test
    a=DU_ExtractFitFill(inputFit,SUB,0);

    % Updating the output variable
    tempOutput(countCombinations,:)=a(1,:);
end
countMesh=countMesh+1;
end
% Finding the distance from the head center to each point and creating a
colormap
tempvec1=(10*HIP{STEM_SIZE_input}.TAPER_NORM+implantHeadDOE(:,2:4))-
implantHeadDOE(:,2:4);

```

```

tempvec2=AL_I(1,2:4)-implantHeadDOE(2:4); %vector from implant COR to
Femoral head
D(countCombinations,1)=norm(cross(tempvec2,tempvec1))/norm(tempvec1);

```

Evaluating ICC Node Overclosures

```

%first transforming implant nodes to the sulcus coordinate system
tempimplantNodesDOE_sulcus=(TM_ItoS*implantNodesDOE)';
tempcalcar=(TM_ItoS*calcar_IC_I)';

neckNodes=tempimplantNodesDOE_sulcus(tempimplantNodesDOE_sulcus(:,4)<=min(tempcalcar(:,4))+0.5 & tempimplantNodesDOE_sulcus(:,4)>=min(tempcalcar(:,4))-0.5 ,:);
neckNodes(:,4)=0;
centroidImplantNodesDOE=mean(neckNodes,1); %pulling the centroid of the
nodes

%Identifying points of intersection
rightLS=[centroidImplantNodesDOE(:,2)
centroidImplantNodesDOE(:,2);centroidImplantNodesDOE(:,3)
centroidImplantNodesDOE(:,3)+100]';
leftLS=[centroidImplantNodesDOE(:,2)
centroidImplantNodesDOE(:,2);centroidImplantNodesDOE(:,3)
centroidImplantNodesDOE(:,3)-100]';

[rLS,~]=polyxpoly(rightLS(:,1),rightLS(:,2),tempcalcar(:,2),tempcalcar(:,3));

[lLS,~]=polyxpoly(leftLS(:,1),leftLS(:,2),tempcalcar(:,2),tempcalcar(:,3));

if xor(isempty(rLS),isempty(lLS))==1 || (isempty(rLS) && isempty(lLS))
    fitCalcarColor(countCombinations,1)=0;
else
    fv.vertices=tempimplantNodesDOE_sulcus(:,2:4);
    fv.faces=implant_elements(:,2:4);

    %finding the exact points of intersection
    lin_temp=intersectPlanesurf(fv,p0_CA,[0,0,1]);

    %determining if the selected nodes are the larger shape or not
    if numel(lin_temp)>1
        MaxResults = cellfun(@(x)size(x,2), lin_temp);
        [~,indToKeepMax]=max(MaxResults);
        lin=lin_temp{indToKeepMax}';
    else
        lin=lin_temp{1}';
    end
    bndryImplant=unique(boundary(lin(:,1),lin(:,2),0.5),'stable');

[insideCalcar,~]=inpolygon(tempcalcar(calcarNodesToCheck,2),tempcalcar(calcarNodesToCheck,3),lin(bndryImplant,1),lin(bndryImplant,2));

```



```

        if all(insideCalcar==0)
            fitCalcarColor(countCombinations,1)=1;
        end
    end
end
% Clearing the inputs for a fresh parfor loop
fv=[];inputFit=[];fvr=[];

end
%UPDATING PATIENT STRUCTURE
if strcmp(RL,'R')==1
    %# find valid cells
    globalIND=logical(fitCalcarColor.*fitDiaphysisColor);
    cellsToKeep=find(globalIND==1, 1);
    if isempty(cellsToKeep)~=1
        PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT={};

PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(tempOutput,1),1:size(tempOutput,2))=tempOutput;

PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),4)=TM(:,1);

PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),1)=num2cell(logical(fitCalcarColor));

PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),2)=num2cell(logical(fitDiaphysisColor));

PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),3)=num2cell(logical(fitCalcarColor.*fitDiaphysisColor));
        PATIENT.R.CORAIL135STD{STEM_SIZE_input,3}=
find(cell2mat(PATIENT.R.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),3)) == 1);

PATIENT.R.CORAIL135STD{STEM_SIZE_input,4}=length(PATIENT.R.CORAIL135STD{STEM_SIZE_input,3});
        %Including the Reduced Patch Elements and Nodes for plotting

PATIENT.R.CORAIL135STD{STEM_SIZE_input,6}=[ones(size(HIP{STEM_SIZE_input}.RP05V,1),1), HIP{STEM_SIZE_input}.RP05V(:,2:3),HIP{STEM_SIZE_input}.RP05V(:,4)-heightToMedialCalcar];
        PATIENT.R.CORAIL135STD{STEM_SIZE_input,7}=HIP{STEM_SIZE_input}.RP05E;
    else
        disp('No More Valid Combinations')
    end

elseif strcmp(RL,'L')==1
    %# find valid cells
    globalIND=logical(fitCalcarColor.*fitDiaphysisColor);
    cellsToKeep=find(globalIND==1, 1);

```

```

        if isempty(cellsToKeep)~=1
            %Updating PATIENT structure
            PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT={};

PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(tempOutput,1),1:size(
tempOutput,2))=tempOutput;

PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),
4)=TM(:,1);

PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),
1)=num2cell(logical(fitCalcarColor));

PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),
2)=num2cell(logical(fitDiaphysisColor));

PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs_DOE_input,1),
3)=num2cell(logical(fitCalcarColor.*fitDiaphysisColor));
            PATIENT.L.CORAIL135STD{STEM_SIZE_input,3}=
find(cell2mat(PATIENT.L.CORAIL135STD{STEM_SIZE_input,1}.STEMALIGNMENT(1:size(combs
_DOE_input,1),3)) == 1);

PATIENT.L.CORAIL135STD{STEM_SIZE_input,4}=length(PATIENT.L.CORAIL135STD{STEM_SIZE_
input,3});
            %Including the Reduced Patch Elements and Nodes for plotting

PATIENT.L.CORAIL135STD{STEM_SIZE_input,6}=[ones(size(HIP{STEM_SIZE_input}.RP05V,1)
,1), HIP{STEM_SIZE_input}.RP05V(:,2:3),HIP{STEM_SIZE_input}.RP05V(:,4)-
heightToMedialCalcar];
            PATIENT.L.CORAIL135STD{STEM_SIZE_input,7}=HIP{STEM_SIZE_input}.RP05E;
        else
            disp('No More Valid Combinations')
        end
    end
    fitCalcarColor=logical(fitCalcarColor);
    fitDiaphysisColor=logical(fitDiaphysisColor);
    fitColor=logical(fitCalcarColor.*fitDiaphysisColor);
    %updating the viz
    D_plot=D(:,1)/max(D(:,1));
end
end

```

Helper Function(s)

```

%Check surface Normals
function input=checkSurfaceNormals(input)

```

```

%generating the surface normals and checking them
[testNodes,test_normals]=getFaceCenterAndNormals(input.faces,input.nodes);

cond1=0;
cDIA=mean(input.nodes(:,1:3),1);
for counti=1:size(test_normals,1)
    %drawing a vector from the center face node in the direction of the normal
    vec1= testNodes(counti,1:3) -[cDIA(1,1:2),testNodes(counti,3)] ;
    %Dot Product
    tempDot=dot(vec1,test_normals(counti,1:3));
    if tempDot<0
        %Swap The Face Nodes
        input.faces(counti,:)=[input.faces(counti,2) input.faces(counti,1)
input.faces(counti,3)];
        cond1=1;
    end
end
if cond1==1
    %updating the structure

[input.face_mean_nodes,input.face_normals]=getFaceCenterAndNormals(input.faces,input.nodes);
else
    input.face_mean_nodes=testNodes;
    input.face_normals=test_normals;
end

%%% Verifying the surface normals
%%% figure()
%%% patch('Vertices',input.nodes,'Faces',input.faces,'FaceColor',[1 0
0],'FaceLighting','gouraud','EdgeAlpha',0.3,'EdgeColor','none','SpecularStrength',
.25,'FaceAlpha',0.35); hold on
%%% for countNormals=1:length(input.face_mean_nodes)
%%%
plot3([input.face_mean_nodes(countNormals,1),input.face_mean_nodes(countNormals,1)
+10*input.face_normals(countNormals,1)],...
%%%
[input.face_mean_nodes(countNormals,2),input.face_mean_nodes(countNormals,2)+10*input.face_normals(countNormals,2)],...
%%%
[input.face_mean_nodes(countNormals,3),input.face_mean_nodes(countNormals,3)+10*input.face_normals(countNormals,3)],'k-')
%%% end
%%% axis equal
end

%Extract Fit and Fill Information
function [STEMALIGNMENT]=DU_ExtractFitFill(input,SUB)

```

Step 1. Transforming all data to correct cosys and calculating area ratio at user specified heights

```
%transforming the nodes of each slice from the implant Coordinate System to
%the Anatomic
ProxD_A=(input.TM_AtoI*input.ProxD)';
ProxI_A=(input.TM_AtoI*input.ProxI)';
DistD_A=(input.TM_AtoI*input.DistD)';
DistI_A=(input.TM_AtoI*input.DistI)';

% from these calculating the area
A1_Prox=polyarea(input.ProxI(:,2),input.ProxI(:,3));
A2_Prox=polyarea(input.ProxD(:,2),input.ProxD(:,3));
proxAreaRatio=A1_Prox/A2_Prox * 100;

A1_Dist=polyarea(input.DistI(:,2),input.DistI(:,3));
A2_Dist=polyarea(input.DistD(:,2),input.DistD(:,3));
distAreaRatio=A1_Dist/A2_Dist * 100;
```

Step 2: Figuring out slice based minimum distances at user specified heights

```
%finding proximal min and max nodes
[~,ML_Lower]=min(ProxI_A(:,2));[~,ML_Upper]=max(ProxI_A(:,2));
[~,AP_Lower]=min(ProxI_A(:,3));[~,AP_Upper]=max(ProxI_A(:,3));

%storing
proxImplantNodes=[ML_Lower ML_Upper AP_Lower AP_Upper];
%clearing
clear ML_Lower ML_Upper AP_Lower AP_Upper

%finding distal min and max nodes
[~,ML_Lower]=min(DistI_A(:,2));[~,ML_Upper]=max(DistI_A(:,2));
[~,AP_Lower]=min(DistI_A(:,3));[~,AP_Upper]=max(DistI_A(:,3));

%storing
distImplantNodes=[ML_Lower ML_Upper AP_Lower AP_Upper];
clear ML_Lower ML_Upper AP_Lower AP_Upper

%initializing the distances array
distances=zeros(2,4);
%Finding the Distances in anatomic space
for i=1:4
    if i==1 || i==2
        %pulling the proximal distances
        [xP,~]=polyxpoly([ProxI_A(proxImplantNodes(i),2)-50
ProxI_A(proxImplantNodes(i),2)+50],[ProxI_A(proxImplantNodes(i),3)
```

```

ProxI_A(proxImplantNodes(i),3],ProxD_A(:,2),ProxD_A(:,3));
    temp=ones(size(xP,1),1).*[ProxI_A(proxImplantNodes(i),2)] ;
    %pulling the minimum distance to the node of interest
    distancesP=xP-temp;
    [~,indToKeep]=min(abs(distancesP));
    distances(1,i)=distancesP(indToKeep(1));
    clear temp indToKeep
    %pulling the distal distances
    [xD,~]=polyxpoly([DistI_A(distImplantNodes(i),2)-50
DistI_A(distImplantNodes(i),2)+50],[DistI_A(distImplantNodes(i),3)
DistI_A(distImplantNodes(i),3)],DistD_A(:,2),DistD_A(:,3));
    temp=ones(size(xD,1),1).*[DistI_A(distImplantNodes(i),2)] ;
    %pulling the minimum distance to the node of interest
    distancesD=xD-temp;
    [~,indToKeep]=min(abs(distancesD));
    distances(2,i)=distancesD(indToKeep(1));
    clear temp indToKeep
else
    %pulling the proximal distances
    [~,yP]=polyxpoly([ProxI_A(proxImplantNodes(i),2)
ProxI_A(proxImplantNodes(i),2)], [ProxI_A(proxImplantNodes(i),3)-50
ProxI_A(proxImplantNodes(i),3)+50],ProxD_A(:,2),ProxD_A(:,3));
    temp=ones(size(yP,1),1).*[ProxI_A(proxImplantNodes(i),3)] ;
    %pulling the minimum distance to the node of interest
    distancesP=yP-temp;
    [~,indToKeep]=min(abs(distancesP));
    distances(1,i)=distancesP(indToKeep(1));
    clear temp indToKeep
    %pulling the distal distances
    [~,yD]=polyxpoly([DistI_A(distImplantNodes(i),2)
DistI_A(distImplantNodes(i),2)], [DistI_A(distImplantNodes(i),3)-50
DistI_A(distImplantNodes(i),3)+50],DistD_A(:,2),DistD_A(:,3));
    temp=ones(size(yD,1),1).*DistI_A(distImplantNodes(i),3) ;
    %pulling the minimum distance to the node of interest
    distancesD=yD-temp;
    [~,indToKeep]=min(abs(distancesD));
    distances(2,i)=distancesD(indToKeep(1));
    clear temp indToKeep
end
end
%Finding Knn based distances
[~,DP]=knnsearch(ProxD_A(:,2:3),ProxI_A(:,2:3));
[~,DD]=knnsearch(DistD_A(:,2:3),DistI_A(:,2:3));
%pulling minimum for plotting and updating the table
[minValP,~]=min(DP);
[minValD,~]=min(DD);

```

Step 3: Find Projected Sulcus Cut Femoral Head Intersection

```

if strcmp(input.RL,'R')==1
    %Extracting and Transforming Subject Data
    AL_G = (SUB.TM.F.GtoAPR*SUB.LM.A.RFemur_Prox)';
    AL_S = (SUB.TM.F.GtoSR*AL_G)';

    %Pulling the Implant information from the DOE
    tempLocations=input.HP;%pulling the head position
    implantLocations_S=(input.TM_ItoS*tempLocations)'; %transforming to sulcus
coordinate system
    tempNORMVEC=(input.TM_ItoS(2:4,2:4)*input.HNV)'; %calculating the normal
vector from the neck trunion axis

    d=-dot([0,0,1],AL_S(1,2:4));
    t= -(d + dot([0,0,1],implantLocations_S(1,2:4))) / dot([0,0,1],tempNORMVEC);
    projectedHeadLocation=[1, implantLocations_S(1,2:4) + tempNORMVEC*t] ;

else
    AL_G = (SUB.TM.F.GtoAPL*SUB.LM.A.LFemur_Prox)';
    AL_S = (SUB.TM.F.GtoSL*AL_G)';

    tempLocations=input.HP; %pulling the head position
    implantLocations_S=(input.TM_ItoS*tempLocations)'; %transforming to sulcus
coordinate system
    tempNORMVEC=(input.TM_ItoS(2:4,2:4)*input.HNV)'; %calculating the normal
vector from the neck trunion axis

    %project the sulcus plane to the anatomic head in sulcus space, also
    %extracting the point of intersection between the plane and the head
    %vector
    d=-dot([0,0,1],AL_S(1,2:4));
    t= -(d + dot([0,0,1],implantLocations_S(1,2:4))) / dot([0,0,1],tempNORMVEC);
    projectedHeadLocation=[1, implantLocations_S(1,2:4) + tempNORMVEC*t] ;

end

%finding the components of the projected vector and overall distance to the
%femoral head. [X distance, Y Distance, Overall Magnitude Distance]
tempTrunionAnalysisPTS=[dot(AL_S(1,2:4)-projectedHeadLocation(1,2:4),[1 0 0]) ,...
    dot(AL_S(1,2:4)-projectedHeadLocation(1,2:4),[0 1 0]),norm(AL_S(1,2:4)-
projectedHeadLocation(1,2:4))];

%Storing Implant Head Position and Trunion Vector of DOE Perturbation in
%Implant COSYS
STEMALIGNMENT{1,5}=tempLocations;
STEMALIGNMENT{1,6}=tempNORMVEC;
%Storing the 20% and 60% Distances and Areas
STEMALIGNMENT{1,7}=proxAreaRatio;
STEMALIGNMENT{1,8}=[distances(1,:),minValP];
STEMALIGNMENT{1,9}=distAreaRatio;
STEMALIGNMENT{1,10}=[distances(2,:),minValD];
%Storing the Trunion Vector

```

```
STEMALIGNMENT{1,11}=tempTrunionAnalysisPTS;  
STEMALIGNMENT{1,12}=projectedHeadLocation;
```

```
end
```