



PROJECT DELIVERABLE REPORT



Project Title:

Advanced personalised, multi-scale computer models preventing osteoarthritis
SC1-PM-17-2017 - Personalised computer models and in-silico systems for well-being

Deliverable number	3.1
Deliverable title	Personalized computer biomechanical models
Submission month of deliverable	18
Issuing partner	LJMU
Contributing partners	CERTH, UPAT
Dissemination Level (PU/PP/RE/CO):	PU
Project coordinator	University of Nicosia (UNIC)
Tel:	+357 22 841 528
Fax:	+357 22 357481
Email:	felekkis.k@unic.ac.cy& giannaki.c@unic.ac.cy
Project web site address	www.OACTIVE.eu

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 777159

Revision History

Version	Date	Responsible	Description/Remarks/Reason for changes
1.0	14/11/2019	LJMU	Draft

List of Contributors

Name	Affiliation
David Britzman	LJMU
Alexander Abel	LJMU
Lefteris Benos	CERTH
Giorgos Giarmatzis	UPAT
Dimitrios Tsaopolous	CERTH
Vasilios Baltzopoulos	LJMU

Table of Contents

1	Introduction.....	6
2	Pipeline Overview.....	6
3	Musculoskeletal Model.....	7
3.1	Motion Capture Data.....	7
3.2	Musculoskeletal Pipeline	9
3.2.1	Introduction to OpenSim.....	9
3.2.2	Pre-processing.....	10
3.2.3	Generic Models.....	10
3.2.4	Scaling of Generic Model.....	14
3.2.5	Inverse Kinematics.....	14
3.2.6	Inverse Dynamics	16
3.2.7	Static Optimization.....	18
3.2.8	Joint Reaction Analysis	20
3.3	Automated Pipeline.....	21
3.3.1	MskPreProcessing.....	21
3.3.2	MskProcessing.....	22
3.3.3	MskOutput.....	23
4	Finite Element Model	26
4.1	Introduction to FEA.....	26
4.1.1	How does FEA work?	27
4.2	Manual Geometry Generation	28
4.2.1	Segmentation	28
4.2.2	Smoothing.....	30
4.2.3	Meshing.....	31
4.3	Automatic Geometry Generation.....	32
4.3.1	Segmentation	32
4.3.2	Meshing.....	33
4.4	Model Definition	36
4.4.1	Knee Anatomy	36
4.4.2	Material Properties.....	37
4.4.3	Definition of articulating surfaces.....	42
4.4.4	Load Application and Boundary Conditions	43
4.4.5	Validation of FeBio contact algorithm.....	43
5	Digital Media	47
6	References.....	47

List of figures

Figure 1 - Biomechanical pipeline for OActive project including musculoskeletal and finite element analyses	6
Figure 2 – (Right): CGM2.3 marker set for the left leg (is repeated on the right leg as well). Markers with dashed outlines are needed for static trials only to aid with model scaling (adapted from Peters et al 2019) (Left): Thigh marker positions to reduce sensitivity of femur segment orientation (Cockroft et al 2016) ...	8
Figure 3 - Location of the body segment frames (https://simtkconfluence.stanford.edu/display/OpenSim/Gait+2392+and+2354+Models#Gait2392and+2354Models-Kinematics)	12
Figure 4 - Lerner knee model (REF)	13
Figure 5 - Schematic showing how geometric scaling is achieved by comparing experimental markers with those on the model	14
Figure 6 - Illustrative depiction of inverse kinematics process (https://simtkconfluence.stanford.edu:8443/display/OpenSim/Getting+Started+with+Inverse+Kinematics)	15
Figure 7-Average knee flexion angle for one subject walking	16
Figure 8 - Knee joint adduction moment for one subject undergoing walking, sit to stand and step up and over trials. Data is normalized to the patient’s body mass	17
Figure 9 - Illustrative depiction of static optimization process (https://simtkconfluence.stanford.edu:8443/display/OpenSim/Getting+Started+with+Static+Optimization)	18
Figure 10 - Example frames from sit to stand, walking and step up trials showing marker locations as well as muscle activations. Muscles experiencing higher activations appear redder	19
Figure 11 - Average muscle activations for one subject walking. Muscles displayed are the Rectus Femoris (acting around the hip); Soleus (acting around the ankle) and Sartorius (acting around the knee)	20
Figure 12 - Knee JCF for one subject undergoing sit to stand (STST), walking (Walk) and step up and over (Step) trials during stance phase	21
Figure 13 - Example input file for MskProcessing function.....	23
Figure 14 - Screenshot showing csv file used as input to MskOutput function. This example would output mean left and right hip flexion for walking, sit to stand and step up trials	24
Figure 15 - Example of subject data file for inputting masses to the function.....	25
Figure 16 - Screenshot showing an output file that was created using the function. Mean left and right hip flexion across 3 types of activity are reported for 5 subjects. This data can either be stored or can be transferred into the OActive database to be used in the neural networking module	25
Figure 17 - Screenshot showing graphical output from the function (if requested). The hip flexion angles can be seen for all 5 subjects for right and left legs for each of the 3 activities	26
Figure 19 - Linear 4-node and quadratic 10-node tetrahedral element.....	28
Figure 20 MRI of advanced medial osteoarthritis with visible osteophytes. Cross-section images from the axial, sagittal and frontal plane.....	29
Figure 21 Completed segmentation and created surface geometries	30
Figure 22 Original and smoothed surface geometry of the femur.....	31
Figure 23 Femur surface geometry before and after decimation	32
Figure 23 - Methodology of creating the volumetric mesh of the femoral articular cartilage: (a) Scanning of the initial geometry in radial and angular directions utilizing cylindrical coordinations, (b) Definition of initial nodes, (c) Creation of initial volumet	34

Figure 24 - Anterior (left) and posterior (right) view of the 3D volumetric mesh of: (a,b) Femur and (c,d) Tibia	35
Figure 25 - 3D volumetric mesh of: (a) lateral (red) and medial (blue) menisci, (b) outer layer of the femoral articular cartilage and (c) outer layer of the tibial articular cartilage; medial (green) and lateral (pink) part.....	36
Figure 26 - Schematic illustration of knee joint anatomy	36
Figure 27 - Definition of medial and lateral articulating surfaces on the femur.....	42
Figure 28 - Definition of medial and lateral articulating surfaces on the tibia.....	42
Figure 29 Prescribed rotations and forces along the simulation time steps	43
Figure 30 - Confined compression stress-relaxation analysis in plane strain: (a) The contact model consists of two slabs constrained as depicted (slab width = 12 mm, height = 2 mm) and (b) The no-contact model consists of a single slab (width = 12 mm, height = 4mm) [24]	44
Figure 31 - The prescribed displacement, u_y , on the top surface	44
Figure 32 - Transient response of the confined compression at all the nodes of the contact surfaces in the contact model and the corresponding nodes with the same y coordinate of the no-contact model: (a) Nodal fluid pressure and (b) Nodal vertical displacement.....	45
Figure 33 - Confined compression stress-relaxation analysis in plane strain: (a) The contact model consists of two slabs constrained; Mesh at the beginning of the displacement ($t = 0$ s) and deformed mesh at the end of the displacement ramp ($t = 1$ s) and after reaching equilibrium ($t = 105$ s), (b) The no-contact model consists of a single slab; Mesh at the beginning of the displacement ($t = 0$ s), [24]	46
Figure 34 - Nodal fluid pressure across the nodes of the contact surfaces in the contact model and the corresponding nodes of the no-contact model.....	46

Abbreviations

Short	Long
EC	European Commission
EU	European Union
WP	Work Package
KOA	Knee Osteoarthritis
OA	Osteoarthritis
GRF	Ground Reaction Forces
DOF	Degree of Freedom
COP	Centre of Pressure
JCF	Joint Contact Force
IK	Inverse Kinematics
SO	Static Optimization
JR	Joint Reaction
MOCAP	Motion Capture

1 Introduction

D3.1 presented here provides an overview of the biomechanical models developed as part of the OActive project. These pipelines will produce the subject specific biomechanical loading that will be used as an input to the neural networking modules developed elsewhere. In this report there is documentation describing the overall pipeline as well as detailed documentation describing the individual components of this pipeline and how they can be used to produce subject specific knee joint biomechanics.

2 Pipeline Overview

The pipeline described below highlights how data collected during the project can be used to drive musculoskeletal and finite element models to produce subject specific joint loading parameters. The details of these procedures can be found in the relevant chapters in this deliverable, but the pipeline is described briefly below:

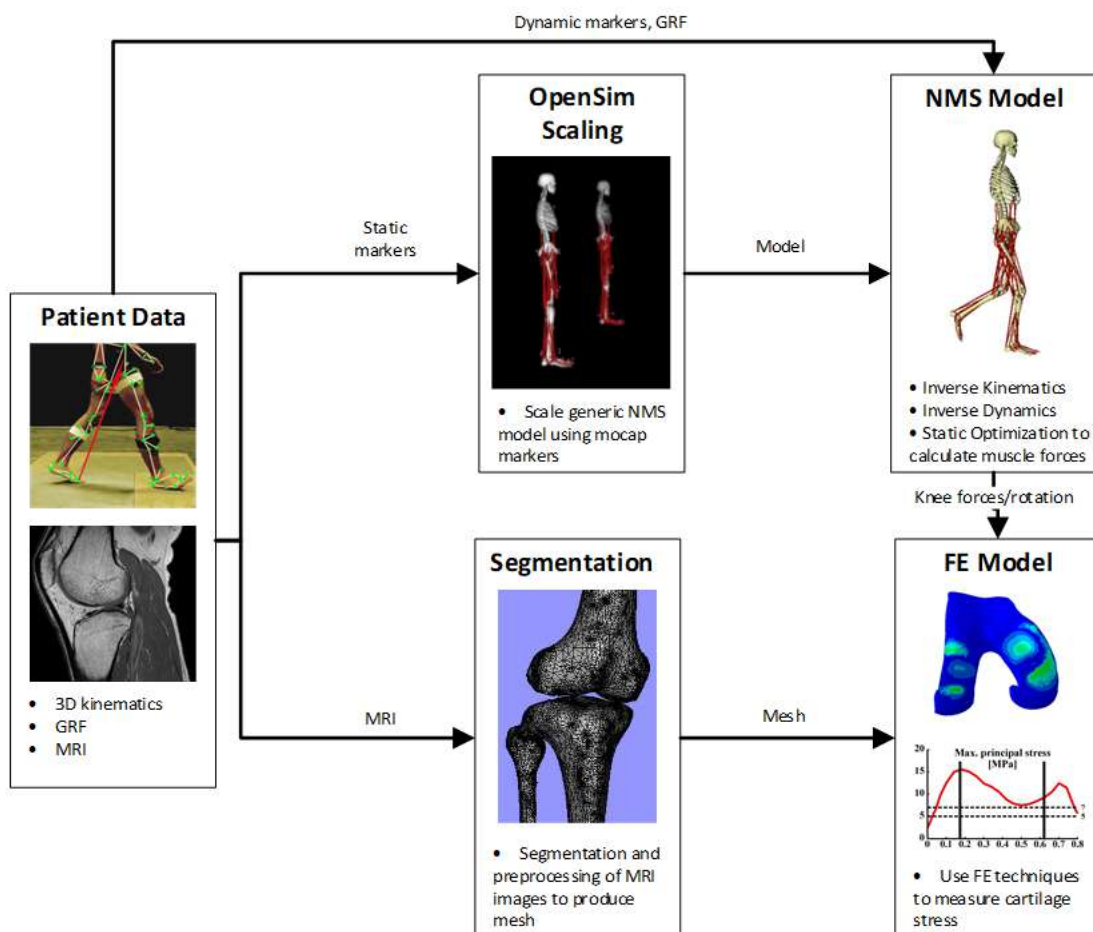


Figure 1 - Biomechanical pipeline for OActive project including musculoskeletal and finite element analyses

Pre-processing

Before carrying out any musculoskeletal analyses the patient data that has been collected must be pre-processed in a consistent manner. Custom MATLAB scripts are used to process the motion capture data

collected in the lab. A *.C3D* file containing marker trajectories, GRF and EMG data (if available) are converted into *.trc* and *.mot* files for use with the rest of the described pipeline. Using standardised processes ensures that data is processed consistently even when collected in different labs using different motion capture equipment.

a) Geometric Scaling

Once the data has been pre-processed, a generic model must be scaled to match as best as possible the anatomy of the subject under consideration. Our approach is to use the scaling function within OpenSim to scale a generic model. The generic model consists of 12 segments, a torso, pelvis, left and right thigh, shank, talus, calcaneus and toes. The pelvis is modelled as a free joint with 6 degrees of freedom (DOF), the torso and hip with 3 DOF, the knee as a sliding hinge joint with 2 DOF, and the ankle with 1 DOF. Marker data from a static trial (*.trc*) will be used to carry out the scaling. Following scaling, if a functional axis of rotation of the knee is to be used, the *.osim* file will be adjusted accordingly. This step will depend on the availability of appropriate data and the outcome of preliminary studies comparing functional and anatomical axis of rotation.

b) Muscle and Joint Force Calculation

OpenSim will be used to perform inverse kinematics and inverse dynamics using the scaled model. This will involve the use of marker data from a dynamic walking trial (*.trc*) alongside GRF data (*.mot*) to produce joint angles and moments. Static optimization will then be used to derive musculotendon and joint contact forces in OpenSim.

c) Finite Element Analysis

Joint kinematics as well as forces from static optimization will be used to drive a finite element model of the knee in FEBio. The forces from the muscle force calculations will define the natural boundary conditions. Combining the mesh derived from MRI data with the personalized muscle forces will result in a patient specific finite element simulation, providing joint stress distributions.

3 Musculoskeletal Model

3.1 Motion Capture Data

Motion capture data is collected for each patient using standard motion capture techniques. This includes the collection of marker based kinematic data as well as kinetic data from force plates. Although the specific methods may vary depending on the equipment/software available at each of the centres, the general principles remain the same and are summarized here.

Marker Set

The marker set used for these studies was based on the conventional gait model (CGM), which has been used widely in motion capture studies in some form since the 1980's. The CGM dataset has 28 markers on the lower limbs, which can be seen in Figure 2 (right) for the left leg. The model differs from previous versions of the CGM in that it includes a series of 3 markers on both the femur and shank (LTHAP, LTHI, LTAD; LTIAP, LTIB, LTIAD). The location of these markers has been designed to minimize soft tissue artefacts, particularly on the femur. Placing the thigh markers as seen in Figure 2 (left) helps to reduce the sensitivity of the model to the orientation of the femur and skin artefact errors.

The nomenclature of the markers is described in Table 1.

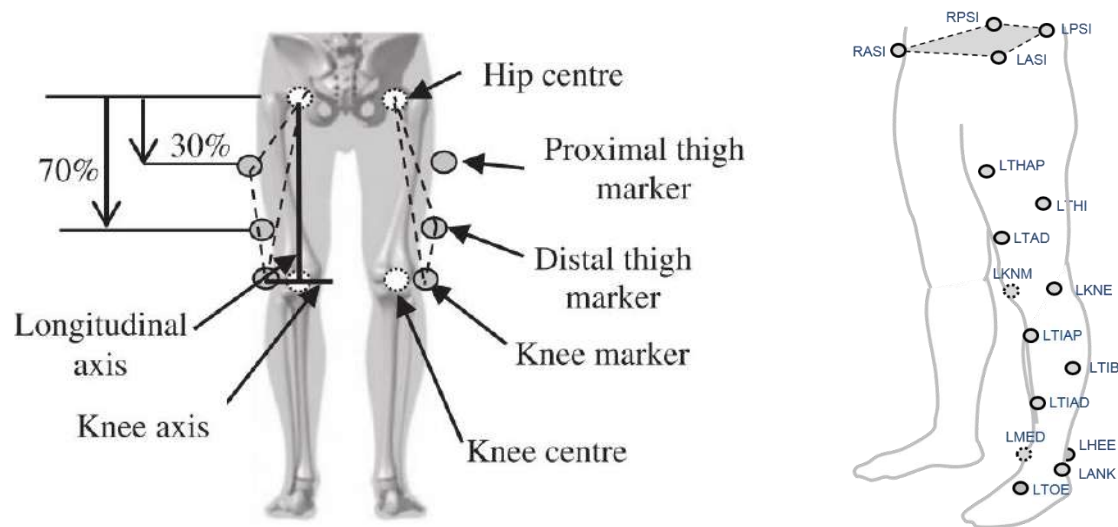


Figure 2 – (Right): CGM2.3 marker set for the left leg (is repeated on the right leg as well). Markers with dashed outlines are needed for static trials only to aid with model scaling (adapted from Peters et al 2019) (Left): Thigh marker positions to reduce sensitivity of femur segment orientation (Cockroft et al 2016)

Table 1 – Marker-set used and locations of the markers ¹Markers used in dynamic trials

Marker Name	Location
RPSI ¹	Right Posterior Superior Iliac
RASI ¹	Right Anterior Superior Iliac
LPSI ¹	Left Posterior Superior Iliac
LASI ¹	Left Anterior Superior Iliac
RTHIAU ¹	Right Thigh Anterior Up
RTHI ¹	Right Thigh
RTHIA ¹	Right Thigh Anterior
LTHIAU ¹	Left Thigh Anterior Up
LTHI ¹	Left Thigh
LTHIA ¹	Left Thigh Anterior
RKNEM	Right Knee Medial
RKNE ¹	Right Knee
LKNEM	Left Knee Medial
LKNE ¹	Left Knee
RTIBAU ¹	Right Tibia Anterior Up

RTIB ¹	Right Tibia
RTIBA ¹	Right Tibia Anterior
LTIBAU ¹	Left Tibia Anterior Up
LTIB ¹	Left Tibia
LTIBA ¹	Left Tibia Anterior
RANKM	Right Ankle Medial
RANK ¹	Right Ankle
LANKM	Left Ankle Medial
LANK ¹	Left Ankle
RTOE ¹	Right Toe
RHEE ¹	Right Heel
LTOE ¹	Left Toe
LHEE ¹	Left Heel

Trials

For each patient a static trial was captured, as well as functional trials for both the knee and hip for joint centre calculation. Dynamic trials of the subject walking, going from sit to stand and stepping up and over a step are also collected. A summary of these can be found in Table 2.

Table 2 - Summary of data types collected during motion capture analysis

Trial	Trial ID	Description
Static	STATIC	Static trial used for model scaling
Functional Knee Left	FCH_L	Functional trial for the left hip used for evaluating the left hip joint centre
Functional Knee Right	FCH_R	Functional trial for the left hip used for evaluating the right hip joint centre
Sit to Stand	STST	Dynamic trial of the subject going from seated to standing position (ideally without the use of hand rests) with ground reaction forces for both legs
Walking Left	WALK_L	Dynamic trial of the patient walking with ground reaction forces for the left leg
Walking Right	WALK_R	Dynamic trial of the patient walking with ground reaction forces for the right leg
Step Up Left	STEP_L	Dynamic trial of the patient stepping up and over a stationary step with ground reaction forces for the left leg
Step Up Right	STEP_R	Dynamic trial of the patient stepping up and over a stationary step with ground reaction forces for the right leg

Data Export

Data is processed and exported using a standard workflow, with gaps in marker trajectories being filled and markers being labelled appropriately. When using appropriate systems, SCORE and SARA algorithms are used to export the functional hip and knee joint centres based on the functional trials. A *.c3d* file needs to be produced for each trial that contains labelled marker trajectories as well as synchronized ground reaction force data.

3.2 Musculoskeletal Pipeline

3.2.1 Introduction to OpenSim

Musculoskeletal models enable us to study neuromuscular coordination, analyse athletic performance and estimate musculoskeletal loads using collected motion capture data. OpenSim is an open-source software that allows users to develop, analyse, and visualize models of the musculoskeletal system, and to generate



dynamic simulations of movement. In OpenSim, a musculoskeletal model consists of a series of rigid body segments connected by joints. Muscles span these joints and generate forces and motion. Once a musculoskeletal model is created, OpenSim enables users to create custom studies, including investigating the effects of musculoskeletal geometry, joint kinematics, and muscle-tendon properties on the forces and joint moments the muscles can produce.

3.2.2 Pre-processing

Before undertaking any musculoskeletal analysis, data has to be pre-processed in a consistent manner so that it can be used. In this project, because the data sources are varied, it is not possible to have a single pre-processing technique that works with all data. Motion capture data collected on different systems comes in differing file formats and therefore needs to be treated differently. In general, though, data is handled in a similar way regardless of origin. Firstly, the 3 dimensional locations of markers and ground reaction force data is extracted from the original file

Relevant frames of interest then have to be selected as processing the entire trials would be time consuming and would not be standardised. The frames of interest for walking, step up and sit to stand trials are shown below:

Trial Type	Start Frame	End Frame
Walking	Heel strike	Toe Off
Step Up	Heel strike	Toe Off
Sit to Stand	Initial load application	

3.2.3 Generic Models

In order to run any simulations of dynamic behaviour, an anatomical model is required upon which these simulations can be run. In this study three separate generic models have been proposed, which can be chosen based on the degree of complexity required and the input data available

Gait2392

The most commonly used model is the Gait2392 model, which has been commonly used in studies of gait. The Gait2392 model is based on cadaveric studies of human anatomy and by default, represents a 1.8m tall man with a weight of 75.16kg. The model contains 7 right body segments, each corresponding to a particular skeletal section. These are the pelvis; femur; patella; tibia/fibula; talus; foot and toes. Each of these segments has a mass and inertia, which are scaled linearly when the model is adapted to a particular subject. Table 3 below summarises the body segments and their default body segment parameters

Table 3 - Segment parameters

Body Segment	Mass (kg)	Moments of Inertia		
		xx	yy	zz
Torso	34.237	0.148	0.756	1.431
Pelvis	11.777	0.103	0.087	0.058
Femur	9.301	0.134	0.035	0.141
Tibia	3.708	0.005	0.005	0.051
Talus	0.100	0.001	0.001	0.001
Calcaneus	1.250	0.001	0.004	0.004
Toe	0.217	0.000	0.000	0.001

Each of these segments has its own reference frame which is centred at the points defined below:

- **Pelvis:** Midpoint of the line between the line connecting the left and right anterior iliac spine
- **Femur:** Centre of the femoral head
- **Tibia:** Midpoint of the line between the medial and lateral femoral epicondyles
- **Patella:** Most distal point of patella
- **Talus:** Midpoint of the line between the apices of medial and lateral malleoli
- **Calcaneus:** Most interior, lateral point on the posterior surface of calcaneus
- **Toe:** Base of second metatarsal

These segments move relative to each other based on the definition of joints: pelvis; hip; knee; ankle; subtalar and metatarphalangeal. The definition of these joints is outlined below:

- **Pelvis:** The pelvic joint exists between the pelvis and the ground and has 6 degrees of freedom (x,y,z translation, x,y,z rotation).
- **Hip:** The hip is defined as a ball and socket joint between the hip and femoral segments. It is modelled as a rotation around three orthogonal axis (flexion, rotation and adduction)
- **Knee:** The knee is defined as a joint between the femur and tibia. In reality there is also a joint between the knee and patella, but including this would significantly impact the robustness and usability of the model. Rotation of the joint is constrained to a single axis (flexion) and translations of the joint in 3 dimensions are defined as a function of this rotation.
- **Ankle:** The ankle is defined as a revolute joint with 2 degrees of freedom

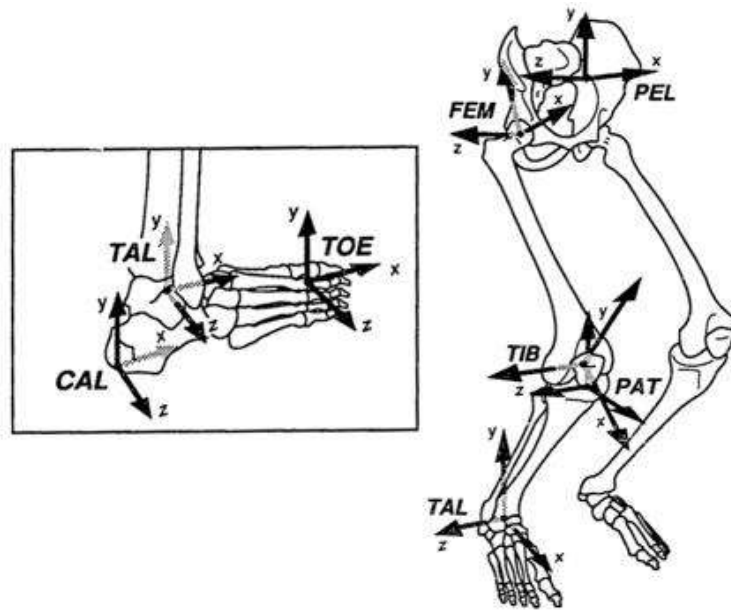


Figure 3 - Location of the body segment frames
 (<https://simtkconfluence.stanford.edu/display/OpenSim/Gait+2392+and+2354+Models#Gait2392and2354Models-Kinematics>)

Forces are applied to these segments, and across these joints by 92 muscle-tendon actuators, which account for 76 muscles acting on the lower limbs and torso. The paths of these muscle-tendon actuators are defined by a series of line segments starting at an origin point on one segment and finishing at an insertion point on another. In some cases these two points are sufficient to describe the geometry of the muscle path, but in other cases via points are required to stop the muscle lines from intersecting the bone. This applies mostly to muscles which wrap over or around bones. Wrapping objects are also included in the model which constrain the lines that muscles can take. They prevent the muscles from intersecting with them and represent real objects such as the underlying skeletal geometry with which the muscles could not intersect in real life.

Gait2392 with Knee Adduction

In order to improve the accuracy and subject specificity of the model, a variety of other model options were considered for potential usage. The first of these was an adaptation of the gait2392 model described above. It has been shown previously that the knee adduction angle can play a key role in the identification of accurate joint contact forces in patients with knee osteoarthritis. The standard gait2392 model does not allow for adduction of the knee and instead models it as a planar joint with a single rotational degree of freedom in the flexion/extension axis. In order to allow for a limited amount of adduction the model was adapted by adding a second rotational degree of freedom in the knee to allow for abduction.

Lerner Model

The second improved model provided as part of this deliverable is based on one created by Lerner et al [25]. The knee joint is slightly more complex than those found on the previously described models so that it can account for subject specific differences in Varus knee alignment and medial/lateral contact points. In patients with osteoarthritis in particular, Varus alignment is known to change significantly and so this was seen to be important to be able to replicate in the model.

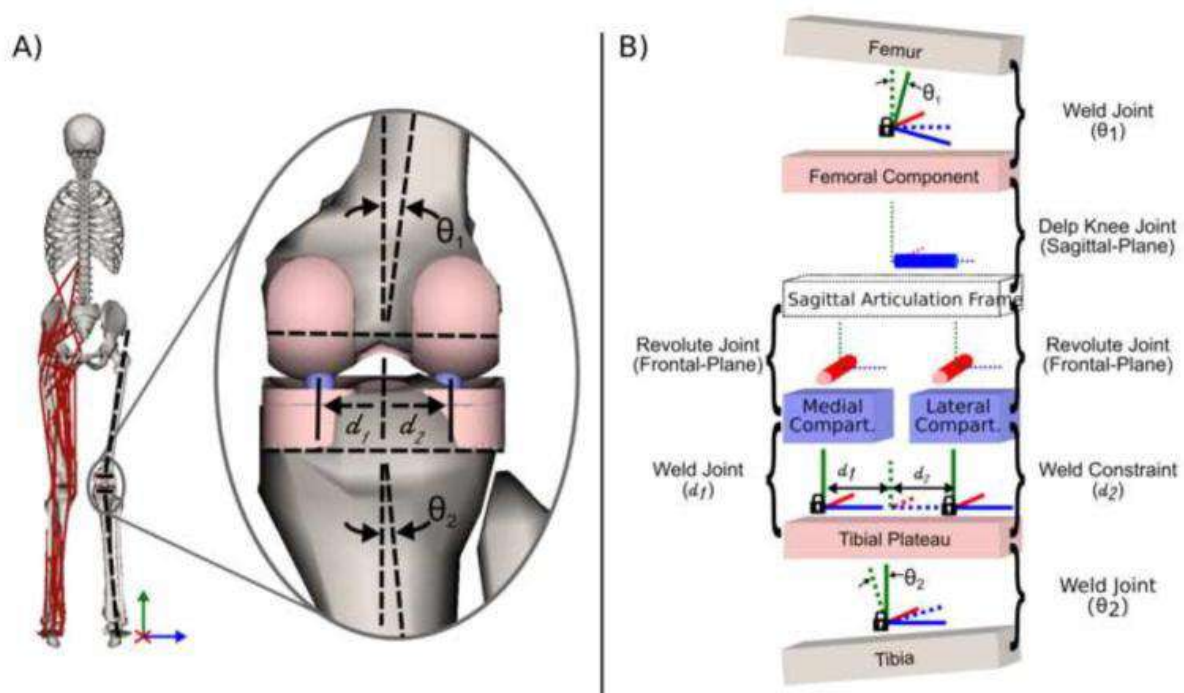


Figure 4 - Lerner knee model [25]

The tibiofemoral joint contains a series of components for manipulating the frontal plane alignment of the knee joint. A distal femoral component and tibial plateau body were added to allow for configuration of the frontal plane alignment, defined by two angles (θ_1 and θ_2). Below the femoral component, a sagittal articulation frame was added. The joint between these defines the movement of the knee in the same way as the knee joint in the Gait2392 model. This is connected via 2 revolute joints to a medial and lateral component, through which medial and lateral joint contact forces are distributed. These components are then connected to the tibial plateau via a series of weld joints that restrict their relative motion. This knee model can be seen in Figure 4.

The subject specific frontal alignment can be defined by altering θ_1 and θ_2 such that $\theta_1 + \theta_2$ equal the functional frontal plane alignment of the specific subject derived from MR data. The subject specific contact points of the medial and lateral condyles can also be altered by changing d_1 and d_2 in the model definition.

Simulations have confirmed that joint contact forces using this more complex knee joint are not significantly different to those using the standard Gait2392 model when not applying subject specific alignment. Because of this it is also possible to use this model without access to MR data, and simply allow the knee to remain in its default alignment. Medial and lateral joint contact forces can still be calculated, but may be erroneous due to the inaccurate joint alignment. In this case it may be best to report total joint contact forces at the knee which are not so reliant on having patient specific knee joint alignment.

3.2.4 Scaling of Generic Model

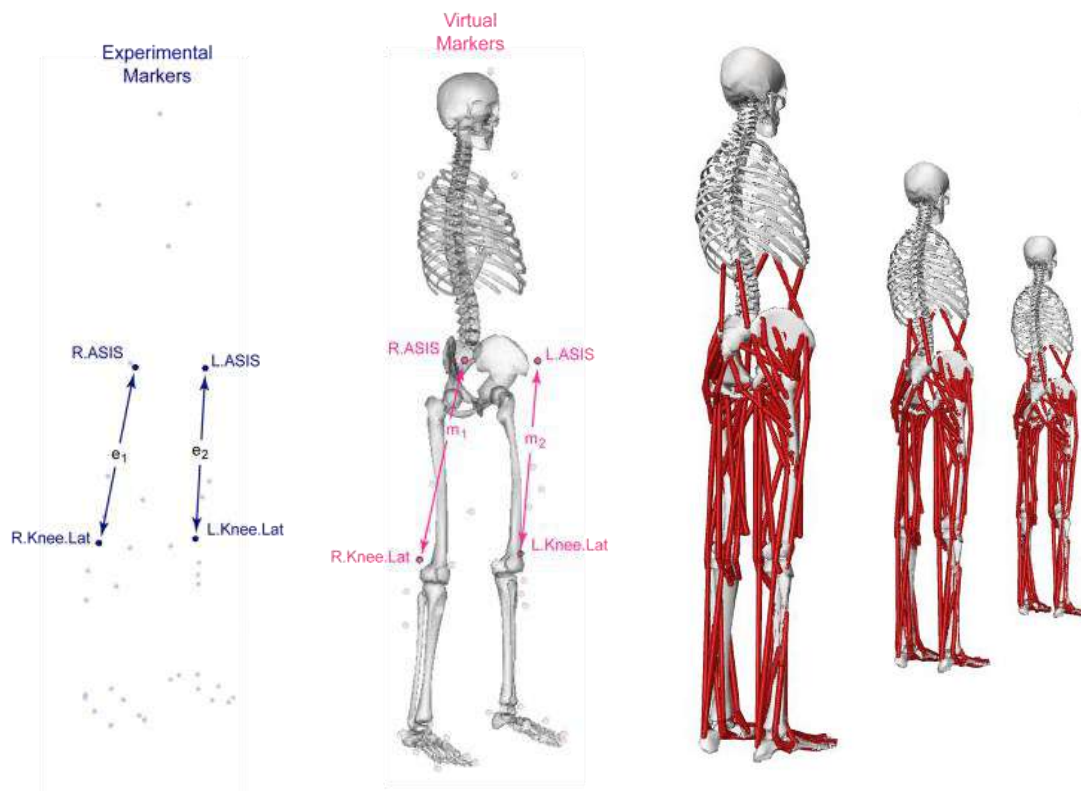


Figure 5 - Schematic showing how geometric scaling is achieved by comparing experimental markers with those on the model

Scaling of the generic model is required so that the segment lengths of the experimental subject match those of the model. There are a number of different methods to appropriately scale generic models to match that of a specific subject, but the most commonly used method is based on the relative positions of virtual and experimental markers.

The location of markers placed on the subjects anatomy at pre-defined locations are measured for a static

Segment	Marker Pair 1	Marker Pair 2
Hip	LASIS - RASIS	
Femur	RKNE - RASIS	LKNE - LASIS
Tibia	RANK - RKNE	LANK - LKNE
Talus	RHEE - RTOE	LHEE - LTOE
Calcaneus	RHEE - RTOE	LHEE - LTOE
Toe	RHEE - RTOE	LHEE - LTOE

trial (the subject stands upright and does not move). The length of segments in the anatomical model is then scaled based on the distances between sets of markers in the modelled and experimental dataset. For example, the tibia segment could be scaled based on the relative distance between the lateral knee and lateral ankle markers for experimental generic marker sets. Other components of the

model that depend on length, such as ligaments and muscles, are also scaled during this process using the same scale factors. The scale factors used in this study can be seen in the Table above. The segments comprising of the foot (talus, calcaneus and toe) are all scaled based on the distance between the heel and toe markers as these are the easiest to accurately locate on the subject.

3.2.5 Inverse Kinematics

Kinematics is the study of motion and in general is used to refer to the understanding of how joint angles change during this motion. The goal of inverse kinematics is to estimate the joint angles of a particular subject using only markers attached to specific locations on the skin. As its input it requires the 3D trajectories of these markers alongside constraints defining the range within which the joint angles can vary. This is undertaken without considering any of the forces involved in the motion.

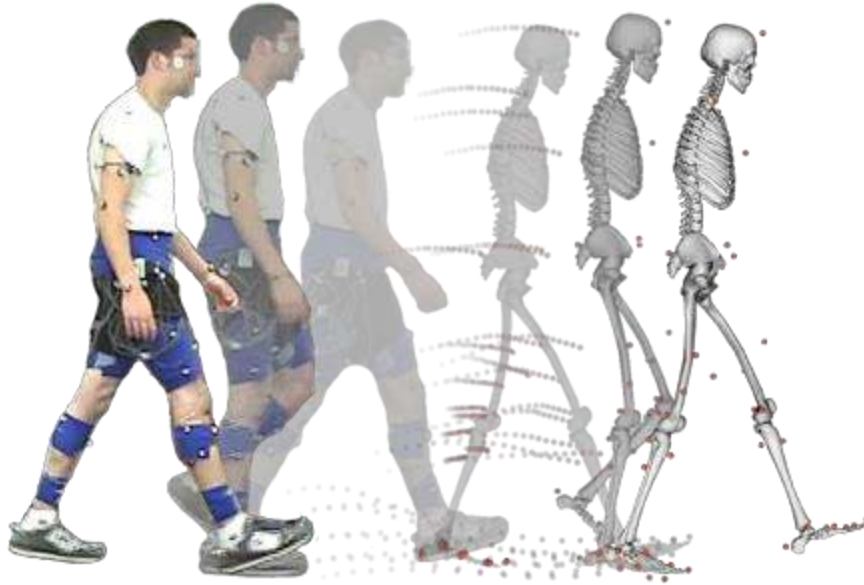
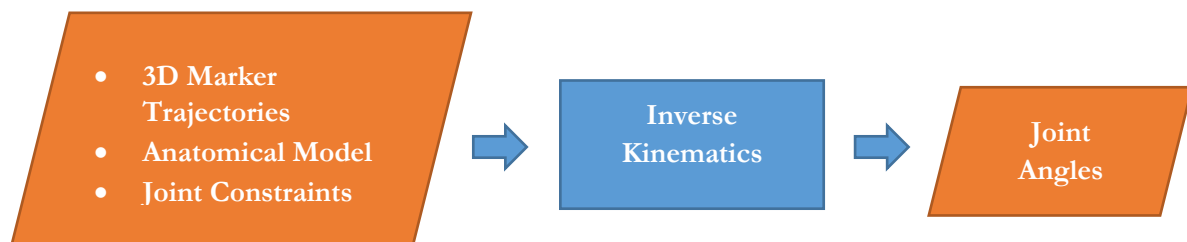


Figure 6 - Illustrative depiction of inverse kinematics process (<https://simtk-confluence.stanford.edu:8443/display/OpenSim/Getting+Started+with+Inverse+Kinematics>)

In OpenSim, the Inverse Kinematics tool cycles through each step one at a time and attempts to find a combination of joint angles for which the experimental 3D marker trajectories best fit the location of the 3D markers on the scaled anatomical model. For each marker a weighting can be given to indicate how important, or otherwise, it is that the experimental marker position matches the model based marker position. For some markers, like around the knee, skin movement causes the marker to move during gait making it less reliable. In this case it would be given a lower weighting than markers on the heel for instance that move significantly less relative to the underlying anatomy. Each joint angle is also constrained such that it can only vary within a certain range, which helps the optimizer find the best fit solution. These constraints are usually broad but can be narrowed if you know, for instance, that the range of motion of a particular joint is very low.



To find this best fit, a least squares optimization based approach is used which minimizes the following function:

$$\min \sum_i w_i \|x_i^{exp} - x_i^{model}\|^2$$

With i being the number of markers, w_i being the weighting allocated to the i^{th} marker, x_i^{exp} being the experimental co-ordinates of the marker, and x_i^{model} being the co-ordinates of the marker on the scaled anatomical model

In the model developed here this was subject to the constraints seen below:

Pelvis_tilt	-90:90	Hip_adduction	-120:120
Pelvis_list	-90:90	Hip_rotation	-120:120
Pelvis_rotation	-210:90	Knee_flexion	-120:10
Pelvis_tx	-5:5	Knee_abd	-10:10
Pelvis_ty	-1:2	Ankle_flexion	-90:90
Pelvis_tz	-3:3	Subtalar_angle	-90:90
Hip_flexion	-120:120	Mtp_angle	-90:90

Figure 7 shows an example of the outputs that can be obtained using inverse kinematics, the average knee flexion angle for one subject walking

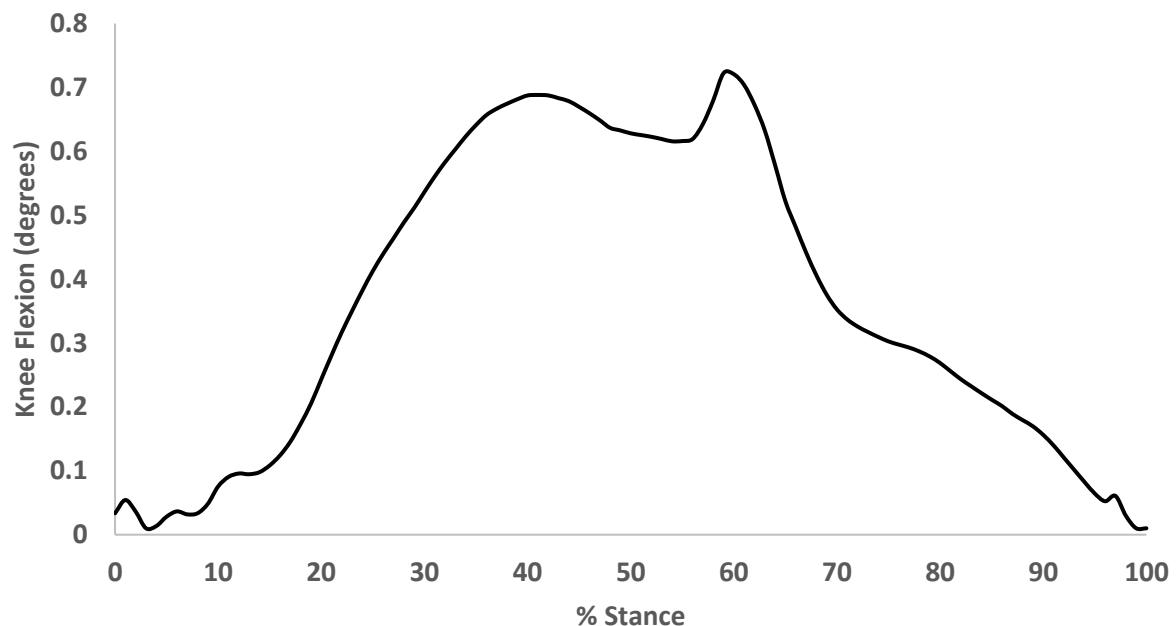


Figure 7-Average knee flexion angle for one subject walking

3.2.6 Inverse Dynamics

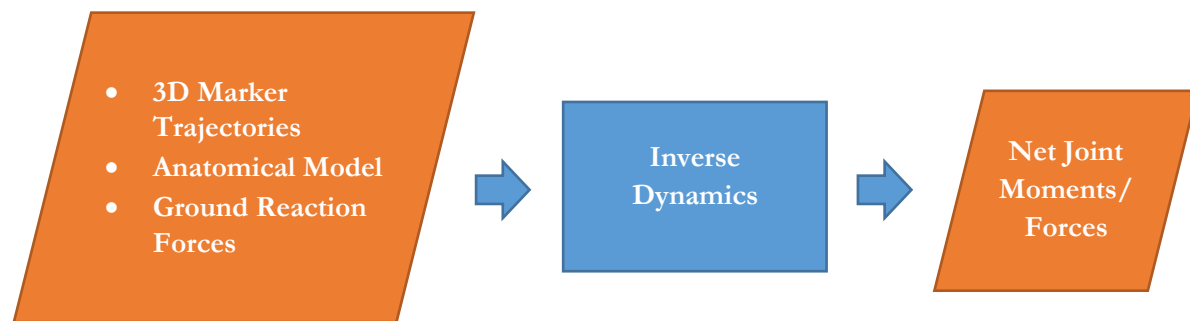
After the joint angles have been calculated using inverse kinematics, they can be combined with information about the forces applied to the body to determine loads and moments in the joints using a process called inverse dynamics. At each frame, forces and moments are resolved around each segment, which are treated as rigid bodies with a mass centre of mass and moment of inertia. The acceleration of each segment is calculated using the results of the inverse kinematic analysis. The equations of motion used to evaluate the forces and moments at each joint are:

$$\sum F = m(\ddot{r}A + \dot{\omega}r_c + \omega(\omega r_c))$$

$$\sum M = I\dot{\omega} + I\omega^2$$

Where $\sum F$ is the resultant force acting on the body segment; $\sum M$ is the resultant moment acting on the body segment; m is the mass of the body segment; rA is the position of the local coordinate system in the global frame; r_c is the position vector of the segment mass center relative to the local coordinate frame; ω is the angular velocity of the local coordinate system; I is the moment of inertia of the segment around the segment centre of mass.

Distal segments are analysed first and then progressively more distal segments are analysed. In this way the calcaneus is analysed first and the hip last.



This process allows the calculation of the net force and moments acting across the joints. This, however, should be differentiated from the joint contact force which is the actual force acting between the articulating surfaces of the joint. This will be calculated later using joint reaction analyses.

Figure 8 shows an example of the outputs from inverse dynamics, the knee adduction moment for one subject undergoing walking, sit to stand and step up and over trials.

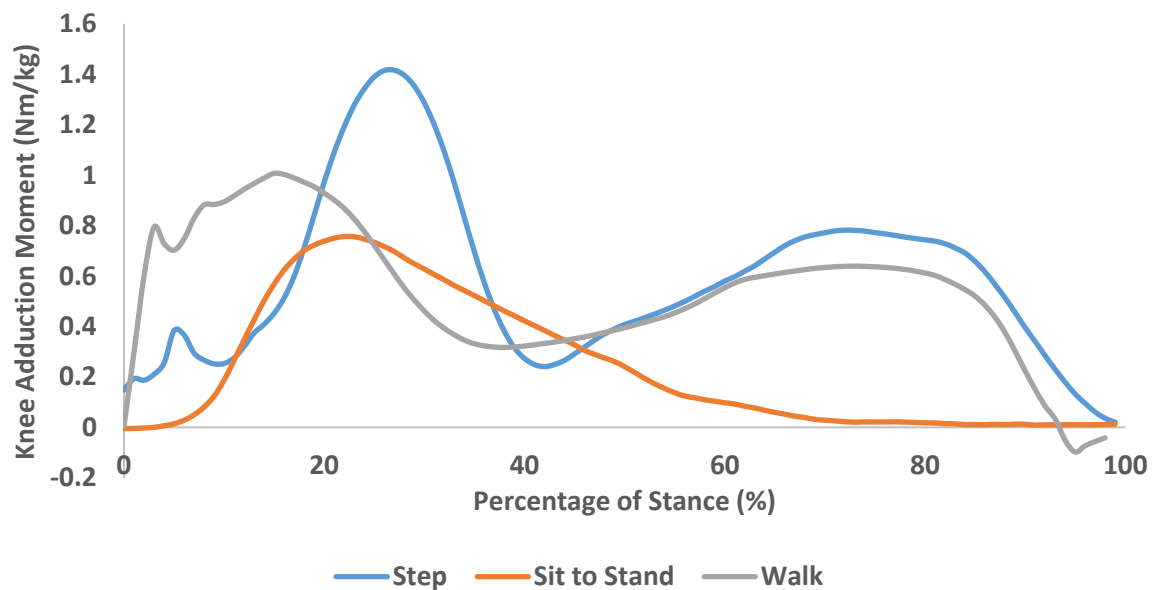
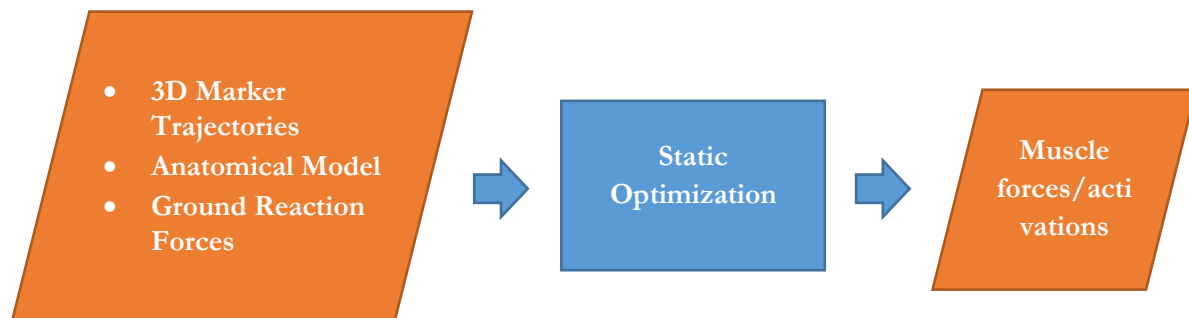


Figure 8 - Knee joint adduction moment for one subject undergoing walking, sit to stand and step up and over trials. Data is normalized to the patient's body mass

3.2.7 Static Optimization



Static optimization is a process which uses information about how a biological system is moving (kinematics) along with information about external loads applied to that system to derive muscle forces for muscles acting across joints for which kinematics are provided. Evaluating muscle forces is a non-trivial solution due to the number of muscles that cross a particular joint. The large number of muscles mean that it is an indeterminate system and as such there is no single mathematically accurate solution. Furthermore, co-contraction of muscles can lead to a plurality of solutions which are not mathematically distinguishable based solely on mechanical equations of motion.

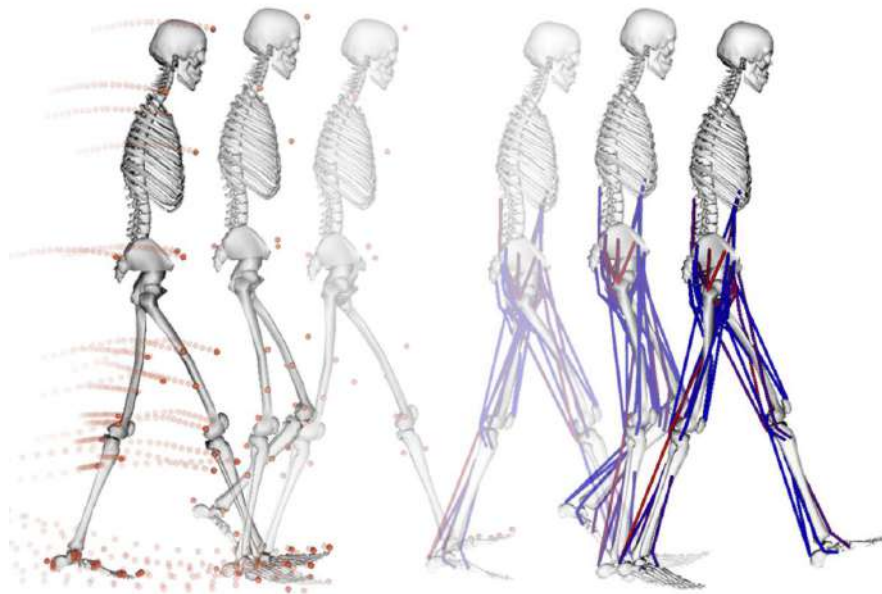


Figure 9 - Illustrative depiction of static optimization process (<https://simtk-confluence.stanford.edu:8443/display/OpenSim/Getting+Started+with+Static+Optimization>)

For these reasons, optimization is required to produce a solution which best minimizes a specific objective function, in this case the sum of the muscle activations squared:

$$E = \sum_{m=1}^n (a_m)^2$$

This optimization is subject to a number of criteria restricting the way in which muscles are allowed to generate force. Muscles can either be modelled as ideal force generators with:

$$\sum_{m=1}^n (a_m F_m^0) r_{m,j} = \tau_j$$

Or they can be constrained by the force-length-velocity properties of the muscle:

$$\sum_{m=1}^n [a_m f(F_m^0, l_m, v_m)] r_{m,j} = \tau_j$$

Where n is the number of muscles in the body; a_m is the activation of muscle m at a given time point; F_m^0 is the maximum isometric force of muscle m ; l_m is the muscle length; v_m is the muscle shortening velocity; $f(F_m^0, l_m, v_m)$ is the force-length-velocity relationship of the muscle; $r_{m,j}$ is its moment arm around the j^{th} joint axis; τ_j is the generalized force acting around the j^{th} joint axis;

Static optimization goes frame by frame and evaluates the optimal solution for each frame independently, regardless of the solution for the frame before or after.

Figure 11 shows an example of the outputs from static optimization, the average muscle activation for one subject walking.

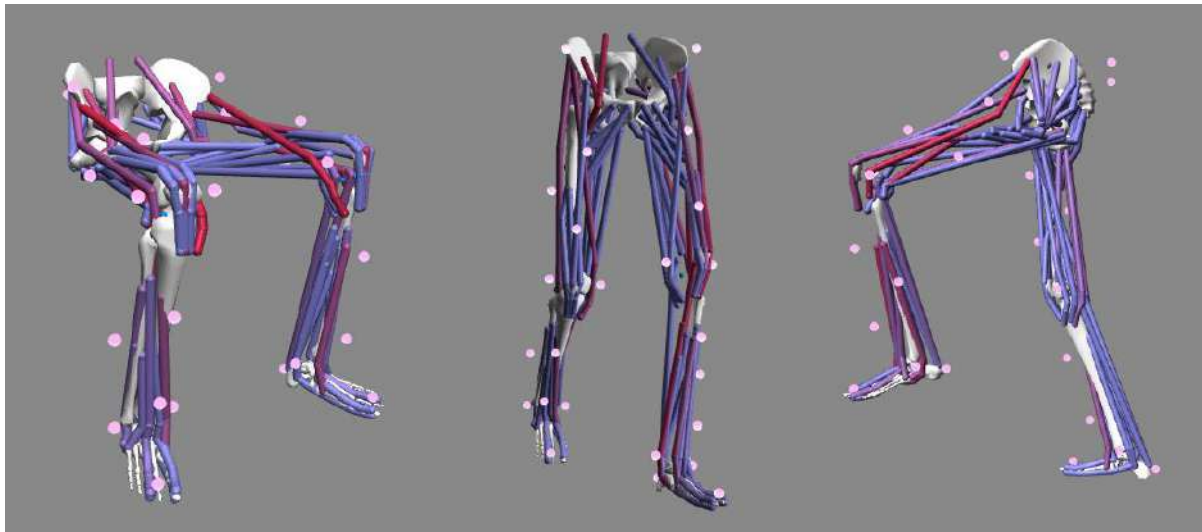


Figure 10 - Example frames from sit to stand, walking and step up trials showing marker locations as well as muscle activations. Muscles experiencing higher activations appear redder

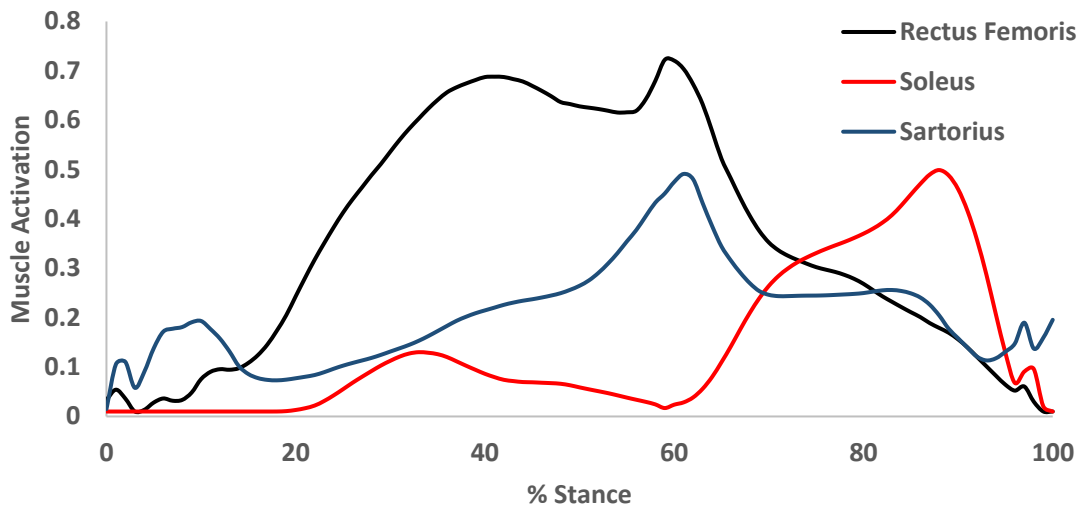


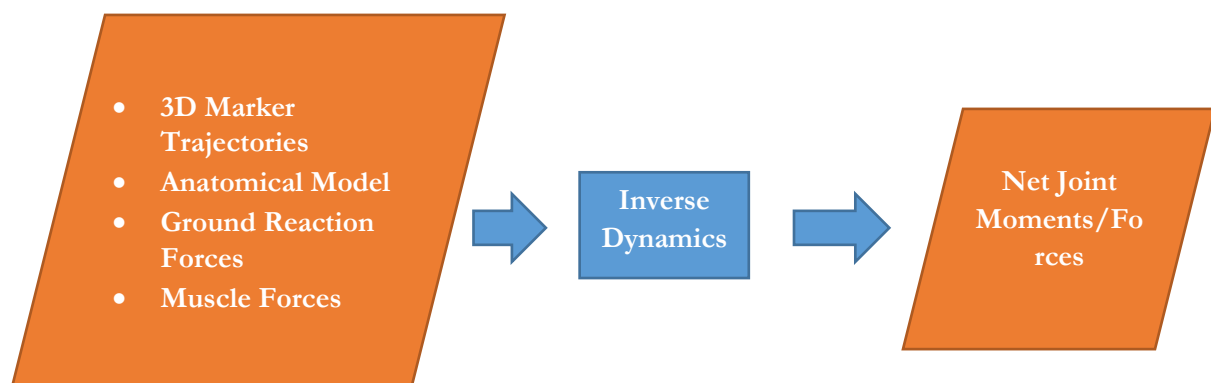
Figure 11 - Average muscle activations for one subject walking. Muscles displayed are the Rectus Femoris (acting around the hip); Soleus (acting around the ankle) and Sartorius (acting around the knee)

3.2.8 Joint Reaction Analysis

Joint reaction analysis is required to predict the internal joint forces that are produced when the muscles act across the joint in a way determined by static optimization. As its inputs it requires a scaled anatomical model, muscle forces for all muscles acting on the system, kinematics determining how each joint is positioned at each frame and ground reaction forces providing information as to the external loads acting on the system. At each frame the following equation is used to calculate the joint reaction at each joint:

$$R_i = M_i a_i - (\sum F_{External} + \sum F_{Muscles} + R_{i-1})$$

Where R_i is the joint contact force at the evaluated joint; M_i is the mass of the segment; a_i is the acceleration of the segment; $F_{External}$ are the external forces acting on the joint; $\sum F_{Muscles}$ is the sum of forces acting on the joints by the muscles (derived from static optimization output) and R_{i-1} is the joint contact force for the joint distal to the one being evaluated.



Setup files for running joint reaction analyses in OpenSim have been provided as part of this deliverable.

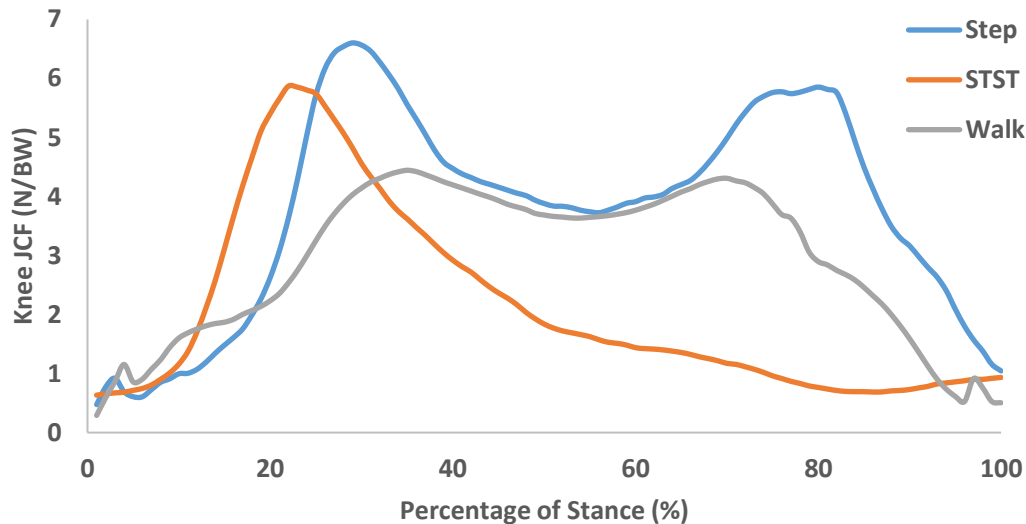


Figure 12 - Knee JCF for one subject undergoing sit to stand (STST), walking (Walk) and step up and over (Step) trials during stance phase

3.3 Automated Pipeline

Software was developed as part of this project to allow the musculoskeletal pipeline to be automated for each subject's data. This is important as it speeds up the processing of the data and also makes it easier for users who don't have knowledge of the OpenSim software environment to process data in a repeatable way. The automated software is developed on MATLAB and consists of 3 modules:

- **MskPreProcessing:** A pre-processing module used to convert the motion capture data into a format that can be used by OpenSim as well as rotating the marker co-ordinates and ground reaction forces from the lab co-ordinate system to the OpenSim co-ordinate system
- **MskProcessing:** A processing module to run the OpenSim pipeline on the pre-processed data, including scaling of a generic model; inverse kinematics; inverse dynamics; static optimization and joint reaction analysis
- **MskOutput:** An output module to visualize each patients processed data and output selected variables of interest including peak adduction moments and joint contact forces

These are all implemented as MATLAB functions which can be passed appropriate data for each patient and are provided as part of the deliverable. At this point the functions are suitable for processing data collected in this project from one centre, but have not been tested on alternative data sources so should be considered in beta status.

3.3.1 MskPreProcessing

The pre-processing module extracts marker trajectories and ground reaction force data from .c3d files produced in the motion capture laboratory and converts them to .trc (marker trajectory) and .mot (ground reaction force data) files. It also rotates the marker data from the original laboratory co-ordinate system into the co-ordinate system desired in OpenSim. This includes both alterations to the definitions of the x, y and z axis as well as rotations of the data to ensure the subject is walking forwards etc. The conversion between the laboratory and OpenSim co-ordinate system can be defined as an input to the MATLAB function by selecting the axis that correspond to the sagittal, coronal and transverse axis using the dialog box shown below.

Note: It is important this is defined correctly or the software will not be able to apply gravity correctly to the model

The function is defined as:

Function MskPreProcessing(root, subjects, offset1, offset2, stepheight, step_ID, units)

Inputs:

root: Root folder where subject folders are found with the input data. The data should be organized as follows: root -> subject -> InputData

subjects: List of subjects for which you would like data to be processed inputted as a cell array. (*Note: if wanting to process all subjects in the root folder then dir() can be used to create a cell array containing the names of all the subjects folders*)

offset1: Force plate 1 offset in mm (needs to have x, y and z offset)

offset2: Force plate 2 offset in mm (needs to have x, y and z offset)

stepheight: Height of step (in m) used in step up and over trials for adjusting the GRF location

step_ID: String containing identifier for step up and over trials (eg “STEP”)

units: String containing units that the forceplate COP is given in (must be either “m” or “mm”)

3.3.2 MskProcessing

MskProcessing is a function that processes the data that has been pre-processed using the MskPreprocessing function. This includes scaling of a generic model for each subject, inverse kinematics, inverse dynamics, static optimization and joint reaction analyses.

The function is defined as:

function []=MskProcessing(root, staticID, subjects, GRF_Table, grf_template, scale_setup, ik_template, id_template, model_file, subjectData_fname)

Inputs

root: Root folder where subject folders are found

subjects: Array of strings containing Subject ID(s) to be analysed

GRF_Table: File name linking to csv file containing data types needing to be analysed along with the feet that make contact with each of the force plates (Example shown in Figure 13). Must contain the following:

- TrialID: Label identifying the type of trial being used to output the variable (for example WALK_L, WALK_R etc)
- TrialTypes: Label identifying the type of trial. Must be either WALK, STST or STEP
- GRF_1: Name of foot making contact with FP1. Must be either right or left (no caps)
- GRF_2: Name of foot making contact with FP2. Must be either right or left (no caps)

	A	B	C	D	E	F	G	H	I	J
1	TrialID	TrialType	GRF_1	GRF_2						
2	WALK_L	WALK	left	right						
3	WALK_R	WALK	right	left						
4	STST	STST	right	left						
5	STEP_L	STEP	left	right						
6	STEP_R	STEP	right	left						
7										
8										
9										
10										
11										
12										

Figure 13 - Example input file for MskProcessing function

grf_template: Filename of xml file that contains the OpenSim grf template

scale_setup: Filename of xml file that contains the OpenSim scale setup

ik_template: Filename of xml file that contains the OpenSim setup file for inverse kinematics

id_template: Filename of xml file that contains the OpenSim setup file for inverse dynamics

model_file: Filename of .osim file containing the OpenSim model template needing to be scaled. This must already have markers attached whose names correspond to the marker names in the trc file

subjectData_fname: String containing file name of csv file containing subject names and masses. Subject names must correspond to those used to store the data (Figure 15)

3.3.3 MskOutput

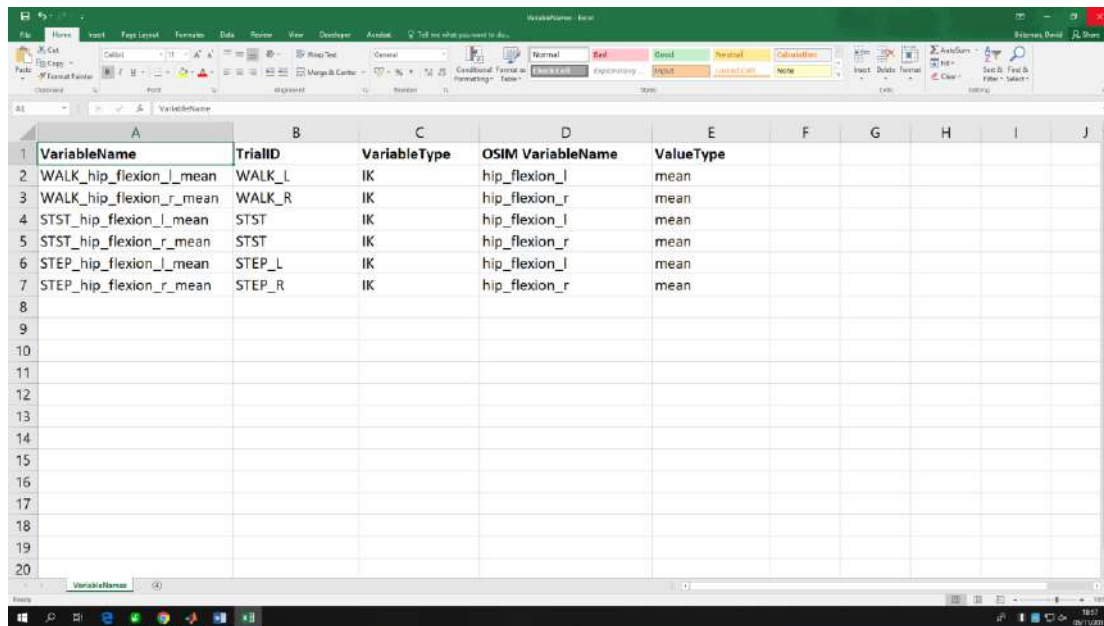
MskOutput is a function designed to enable easy outputting of identified patient parameters from the musculoskeletal model for inputting into the OActive database. The .m file required to run the function is provided as part of this deliverable. The function takes the form:

```
Function [outputTable] = MskOutput(root, subjects, variables_fname...
table_fname, subjectData_fname, plotGraphs)
```

Inputs

root: Root folder where subject folders are found. This should have been created as part of the processing and pre-processing functions that have already be run (*note: if the OpenSim data was created in a way other than using the automated software provided then the data will have to be put into the same file structure created in the automated pipeline for the MskOutput function to work*)

subjects: List of subjects for which you would like data to be processed inputted as a cell array. (Note: if wanting to process all subjects in the root folder then `dir()` can be used to create a cell array containing the names of all the subjects folders)



VariableName	TrialID	VariableType	OSIM VariableName	ValueType
WALK_hip_flexion_l_mean	WALK_L	IK	hip_flexion_l	mean
WALK_hip_flexion_r_mean	WALK_R	IK	hip_flexion_r	mean
STST_hip_flexion_l_mean	STST	IK	hip_flexion_l	mean
STST_hip_flexion_r_mean	STST	IK	hip_flexion_r	mean
STEP_hip_flexion_l_mean	STEP_L	IK	hip_flexion_l	mean
STEP_hip_flexion_r_mean	STEP_R	IK	hip_flexion_r	mean

Figure 14 - Screenshot showing csv file used as input to MskOutput function. This example would output mean left and right hip flexion for walking, sit to stand and step up trials

variables_fname: String containing file name linking to csv file containing variables needing to be visualized with one row per variable. An example is provided as part of this deliverable and can be seen in Figure 14. This file must contain the following headings and be populated accordingly:

- *Variable Name*: Name of variable you want to create
- *TrialID*: String identifying type of trial used to output the variable (for example WALK_L trials might be used to derive the left JCF and WALK_R trials used to derive the right JCF)
- *VariableType*: String identifying the type of data that this variable is: can be IK, ID, SO or JR
- *OSimVariableName*: Name of variable in OpenSim system (for example left knee flexion is knee_angle_r)
- *ValueType*: Type of value needing to be output. Can be max, min, mean or max_abs

table_fname: String containing file name where you want the data table to be outputted.

subjectData_fname: String containing file name of csv file containing subject names and masses. Subject names must correspond to those used to store the data

Subject	Weight (Kg)
Pilot	73.6
Sujeto_01	71
Sujeto_02	67
Sujeto_03	85.6
Sujeto_04	84
Sujeto_05	79
Sujeto_06	73
Sujeto_07	74
Sujeto_08	62
Sujeto_09	108
Sujeto_10	70
Sujeto_11	92
Sujeto_12	72.5
Sujeto_13	65

Figure 15 - Example of subject data file for inputting masses to the function

plotGraphs: Boolean determining if graphs need to be plotted. If true, this will output graphs for each variable comparing between subjects. An example of these graphs can be seen in Figure 17

Outputs

The function will produce a MATLAB table containing all the variables requested. This can be saved for future use or used for further analysis if required. It will create a .csv file from this table using the filename specified. This can be opened in other software where the data can be easily manipulated or transferred to another system such as the OActive database. Figure 16 shows an example of such an output file:

subjectID	WALK_hip_flexion_l_mean	WALK_hip_flexion_r_mean	STST_hip_flexion_l_mean	STST_hip_flexion_r_mean	STEP_hip_flexion_l_mean	STEP_hip_flexion_r_mean
Sujeto_01	-1.713660439	4.251093769	38.64297864	37.5281217	40.86980795	36.58192021
Sujeto_03	-10.78823716	-8.021241267	38.99136858	41.57716004	24.35421845	28.55021988
Sujeto_04	-12.08781018	-15.09684348	43.20165083	41.90567803	28.81717223	29.2018163
Sujeto_05	-4.717849773	0.653187027	48.13608569	51.72532945	30.50701349	39.64076382
Sujeto_06	-12.29012362	-8.707362713	33.59981688	32.63295004	47.32256378	43.16816752

Figure 16 - Screenshot showing an output file that was created using the function. Mean left and right hip flexion across 3 types of activity are reported for 5 subjects. This data can either be stored or can be transferred into the OActive database to be used in the neural networking module

If the graph output is selected then the function will also output a series of graphs that compare each of the variables for all the subjects selected. These can be seen in Figure 17.

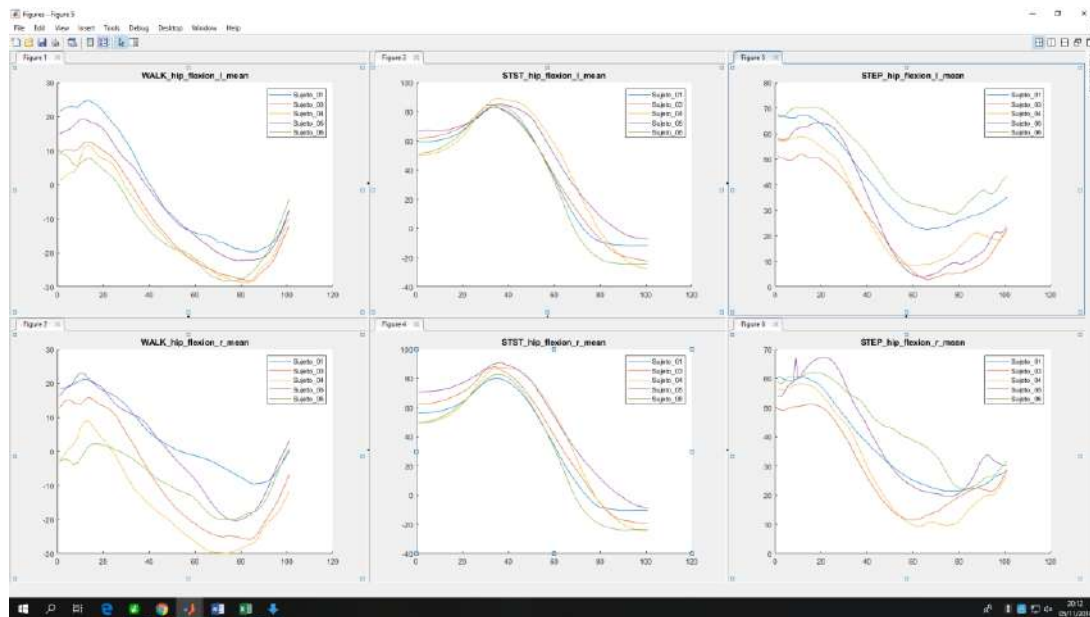


Figure 17 - Screenshot showing graphical output from the function (if requested). The hip flexion angles can be seen for all 5 subjects for right and left legs for each of the 3 activities

4 Finite Element Model

4.1 Introduction to FEA

The finite element analysis is a numerical solution which breaks complex problems, which either have no analytical solution or the analytical solution isn't feasible, into manageable parts called finite elements. Usual applications are the computation of heat transfer, mechanics, electric currents and fluid dynamics. The focus here is on mechanics, calculating resulting displacements, strain and stresses on the mesh composed of elements.

The mesh generation is a key part of any finite element work, as the dimension, type, order and the number of elements affect the outcome of the simulation. Due to the complex geometry of biomechanical problems such as the knee joint, a complete mesh with 3D elements is required to accurately simulate stresses. There are various commercial and open-source meshing algorithms available, such as Tetgen [1], which is used in the following models. Surface geometries are required as input for meshing algorithms which can be sourced from segmenting cross-section images from medical imaging techniques like CT or MRI to create a personalized model.

On the generated mesh, boundary conditions, sliding contact interfaces, material properties and joints need to be defined and assigned. The boundary conditions are the fixed displacements and applied forces, i.e. the natural boundary conditions, which can be either created artificially or sourced from movement trails (see chapter 3). Contact of two meshes requires the definition of the type and properties of the sliding interface and a pair of master and slave surfaces which are meant to be interacting with each other. Each part of the geometry requires to be assigned the appropriate material properties, which are explained in detail in section 4.4.2.

Once the pre-processing is complete, the weak form of the partial differential equations are established and can be solved by the solver. As biomechanical problems are often non-linear an iterative solver is required, which linearizes the problem to define the stiffness matrix. The results can be viewed in the post-processor.

4.1.1 How does FEA work?

The aim of the finite element analysis to produce a stiffness matrix which represents the system of linear equations that produces an approximate solution to the differential equations. The stress-strain relationship in a linear 3D case is defined by

$$\sigma_{ij} = C_{ijkl}\epsilon_{kl}$$

In which C_{ijkl} , the fourth-order constitutive tensor, represents the elastic properties of the material, connecting the strain tensor ϵ_{kl} with the stress tensor σ_{ij} . This produces six independent equations with no unknowns due to symmetry along the trace of C_{ijkl} . The stress equilibrium inside the body can be written as

$$\sigma_{ij,j} + f_i = 0$$

In which f_i describes the body force, i.e. gravitation or inertial forces, here assumed to be zero, resulting in three equations with six unknowns. The relationship between strains and displacements can be expressed as

$$e_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$$

Which provides additional six independent equations with three unknown displacements and six unknown strains, resulting in fifteen equations with fifteen unknown's total. This system of equations can be solved under the defined essential boundary conditions

$$u_i = \bar{u}_i$$

Which represents a fixed part of the model to prevent rigid body motions and natural boundary conditions

$$t_i = \bar{t}_i$$

In which t_i represents the traction $t_i = \sigma_{ij}n_j$ providing the applied forces.

The linear relation of the finite element formulation is expressed by

$$KU = F$$

Where K represents the stiffness constants in the body. The calculated nodal displacements d are approximated across an element through the interpolation shape function matrix N .

$$U = Nd$$

Strains can be calculated through

$$\epsilon = Bd$$

With the strain-displacement matrix B which is a derivative of the shape function N. The element stiffness matrix k is calculated through

$$k = \int_V B^T E B dV$$

For every element.

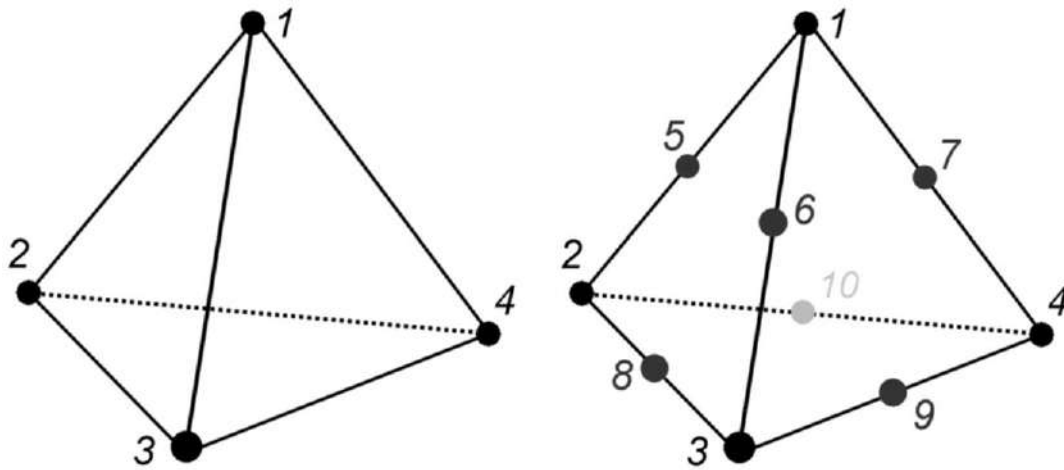


Figure 18 - Linear 4-node and quadratic 10-node tetrahedral element

The shape functions interpolate the displacements calculated only at specific points across whole the element. The order of the shape functions equals the order of the element. Linear elements, featuring one node at each corner, produce linear shape functions while quadratic elements, with an extra node between each corner, therefore produce quadratic shape functions. As strains are calculated as a derivative of the shape functions, quadratic and linear elements produce linear and constant strains across the element respectively.

P-refinement is the corresponding method of choosing higher order elements in cases with high gradients to improve convergence. Alternatively, h-refinement is the method of increasing the number and subsequently reducing the size of the elements to help convergence. Mesh convergence is reached once the simulation result is not determined by changing the mesh size.

In nonlinear cases such as soft tissues and rubber, the stresses are expressed by the strain energy function, with its general form

$$W = W[I_1(C_{ij}), I_2(C_{ij}), I_3(C_{ij})]$$

Where I_1 , I_2 and I_3 are the invariants of the right Cauchy-Green tensor C . The specific nonlinear models used are discussed in more detail in the model definition section (Section 4.4.2).

4.2 Manual Geometry Generation

4.2.1 Segmentation

In order to create the geometries used in the finite element modelling, greyscale cross-section images from different anatomical perspectives produced by medical imaging techniques like MRI and CT need to be segmented. Different types of tissues become discriminable in MRI due to the differences in water content and tissue composition while contrast in CT is based on differences of attenuation of x-rays in various materials. The greyscale images can be segmented manually using open-source software like MITK (<http://mitk.org>). Each significant part of the anatomical structure (i.e. femur, femoral cartilage etc.) is segmented on their own as it is necessary for the mesh generation and the assignment of material properties later in the process.

Manual segmentation requires the user to mark the voxels which belong to the same structure. The larger and easily distinguishable bones can be selected through thresholding or region growing algorithms, which select voxels of similar greyscale on the whole image or around the selected region. Thinner structures like

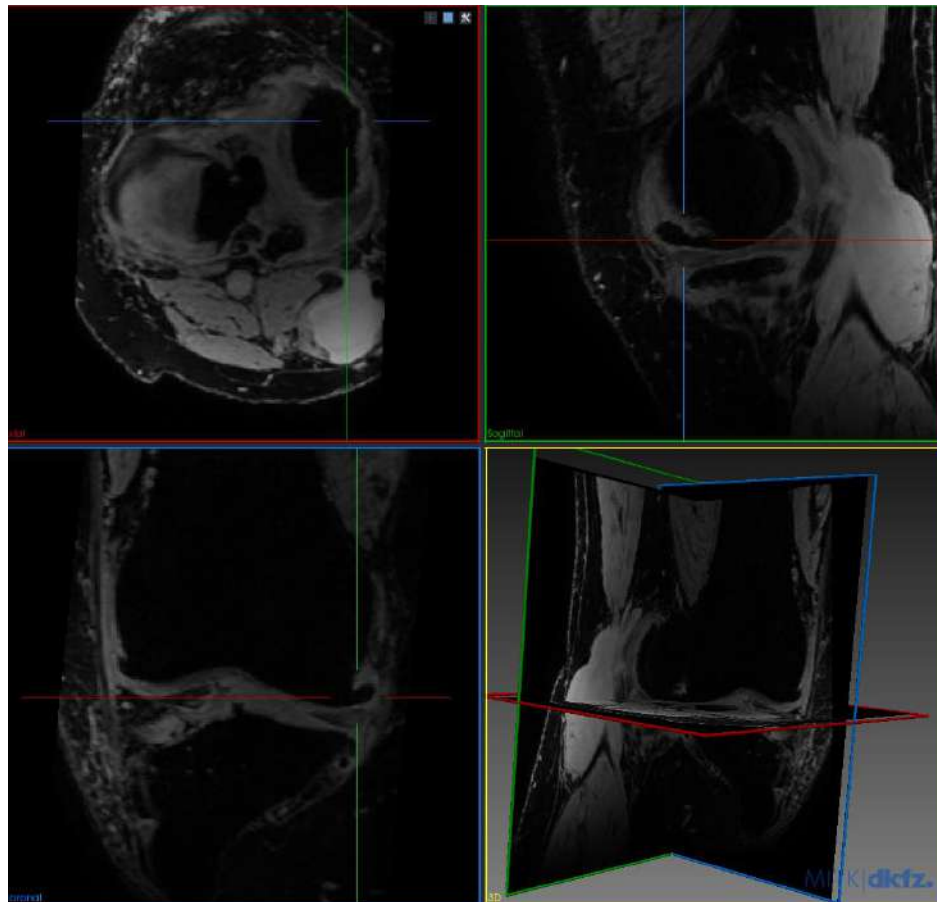


Figure 19 MRI of advanced medial osteoarthritis with visible osteophytes. Cross-section images from the axial, sagittal and frontal plane.

The cartilage and menisci are selected by hand. In both cases, once significant sections of the geometry are mapped, ideally from multiple perspectives, the missing slices can be segmented through interpolation. Errors in difficult regions such as the joint space can be corrected by hand. To prevent a gap between the bone and cartilage after smoothing, the cartilage segmentation is intersected with the corresponding bone. The overlapping geometry is then later removed by Boolean operations.

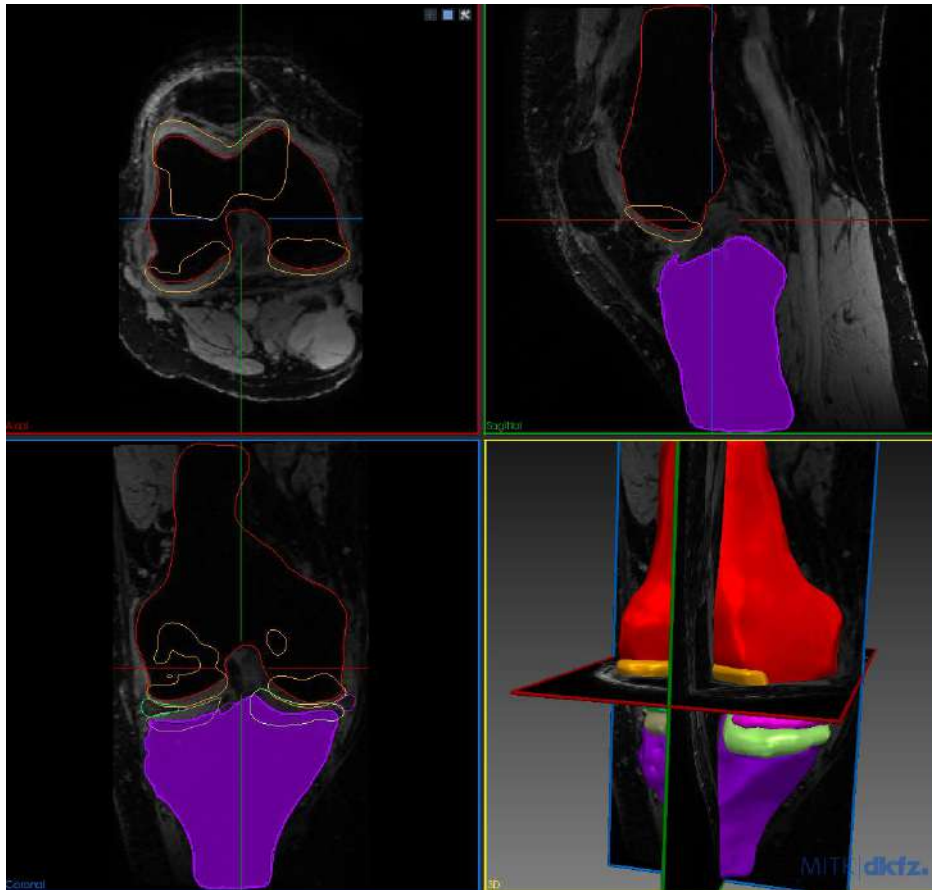


Figure 20 Completed segmentation and created surface geometries

A smoothed polygon model is created by a marching cube algorithm and smoothed from the segmented geometry and exported as a *.stl* surface.

4.2.2 Smoothing

Smoothing is necessary as the slice thickness of the MRI is usually around 1-3mm, creating a geometry featuring sharp edges on slice borders which are difficult to mesh and may falsify the simulation results

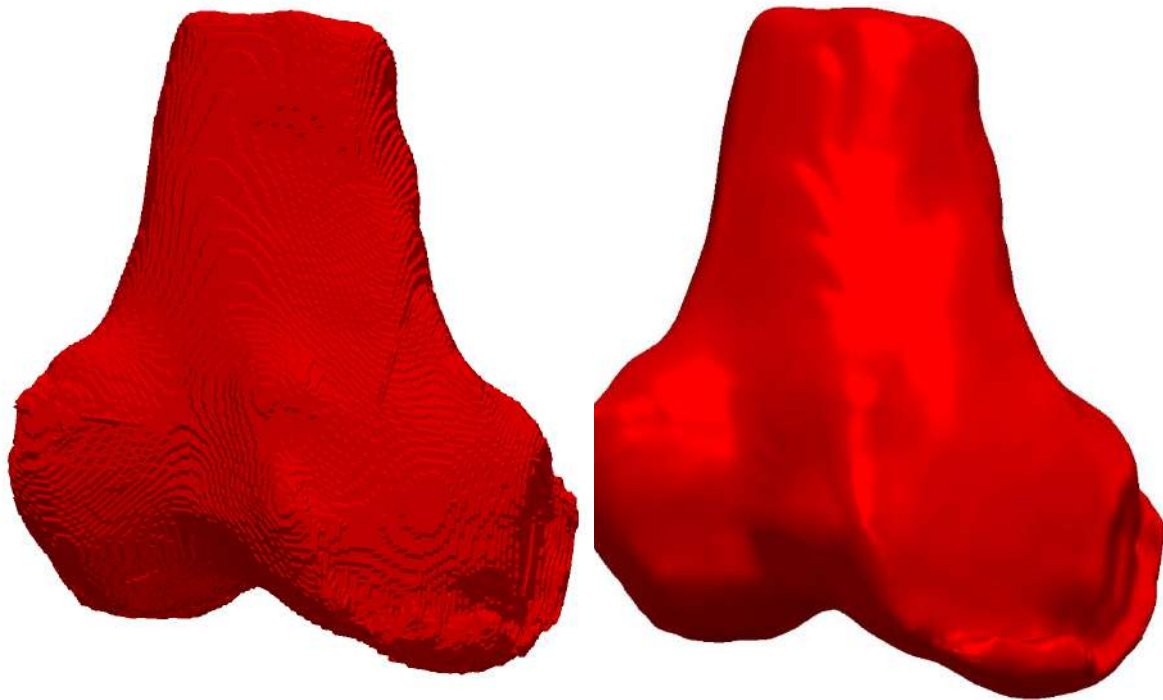


Figure 21 Original and smoothed surface geometry of the femur

due to peak stresses at edges. MITK was used to create a smoothed polygon model using its automated techniques which preserve detailed geometry such as osteophytes.

4.2.3 Meshing

The surface geometries are imported into FEBio's pre-processor PreView [2] which features basic mesh manipulations and integration of TetGen [1], an open-source tetrahedral mesh generator. As an input and boundary for the TetGen program, the surface mesh needs to be decimated to a more coarse state to prevent overly large meshes.

Decimation can lead to mesh errors, i.e. holes, duplicated edges and faces, which need to be resolved prior to meshing through editing the surface geometry. The prepared surface is converted to an editable mesh and the parameters for element size and minimal radius-edge ratio in the TetGen program can be set. The number of elements required to achieve mesh convergence needs to be tested via a convergence study with at least 3 different mesh sizes. Once the results are no longer influenced by mesh size in a set margin, convergence is achieved. The resulting mesh is made out linear tetrahedral (Tet4) elements which can be converted into quadratic (Tet10) ones. In the end, the mesh quality should be checked to prevent errors which will only become apparent after the model definition is complete. Element volume, Jacobians, minimal and maximal dihedral angles can be inspected via PreViews build-in mesh inspector tool.

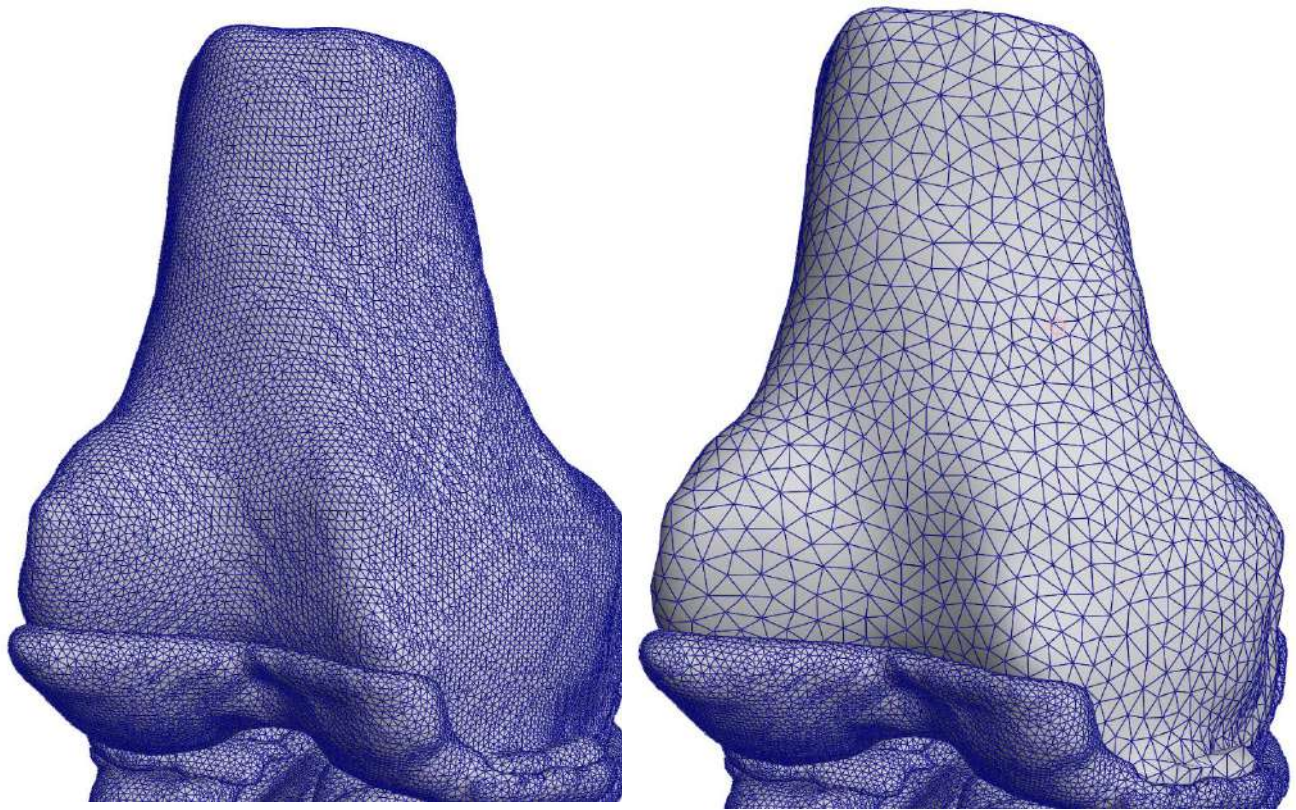


Figure 22 Femur surface geometry before and after decimation

4.3 Automatic Geometry Generation

As well as traditional methods for geometry generation that were explored in the previous section, work has been undertaken to develop approaches to automatically segment MR images. The benefits of this would be a reduction in time required to run the pipeline and the ability for it to be used by less expert users who may not have the technical expertise to develop the geometry manually. Although the implementation is not provided as part of this deliverable, it is the intention that this will prove a key part of the biomechanics pipeline in the future and so the approach is outlined here

4.3.1 Segmentation

The method which was employed for automatic segmentation of the MR images was based on the work of [1], which is a multi-atlas, automatic segmentation scheme. We opt for automatic segmentation methods since they are scalable to large datasets, require less time effort and reproducibility. The dataset of atlases (MR images with the corresponding segmentations) is registered with deformable transformations upon the target unlabelled image, in order to achieve spatial correspondence for the propagation of the atlas label maps to the unlabelled image. However, due to registration inaccuracies and inter-subject variability, the image is not properly segmented. Therefore, a label fusion approach is utilized so that each atlas vote at each location of the unlabelled image for a candidate label. Hence, a weighted voting approach that takes into account the probability of more atlases producing the same label error, is used to decide for the consensus label for a voxel of the target image. Additionally, the label fusion strategy is based on the notion that when two areas in two different images look similar in appearance, might also have similar segmentations. Therefore, the weights are spatially adapted based on the image similarity between each atlas and the novel image, such that the registered atlases that “look” more similar to the novel unlabelled image,

have higher influence during the label fusion step. A key difference to other general multi-atlas segmentation methods is that this method makes use of a local patch search strategy. A specific anatomical image patch on the new image might not be accurately aligned with the corresponding anatomical patch on the registered atlas image. So, when the segmentation algorithm tries to resolve which should be the segmentation of an anatomical image patch, instead of relying solely on the image content of the registered atlases that overlap that patch, the algorithm searches the nearby image content of the registered atlases to find whether a more similar patch exists. Consequently, the resulting segmentation is more robust to registration errors. Finally, the method proposed by [1] utilizes a corrective learning segmentation scheme that tries to correct the systematic errors produced by the automatic segmentation method, that is the similar labelling errors that are produced by different atlases.

4.3.2 Meshing

The segmentation method that is used produces geometries that, due to several segmentation inaccuracies, are noisy with rough surfaces and unconnected or irregular components. Thus, these components should be removed and the extracted triangular geometries should be filtered so that they are appropriate for volumetric meshing and finite element analysis, without any significant loss of the geometric structure. Subsequently, the triangular surface geometries/meshes of the cartilages and menisci are meshed to hexahedra using the method presented by [2]. This method produces hexahedral volumetric meshes for the cartilages and the menisci that are not only adequately smooth for the FE analysis, but also structured (the cartilages only) and subdivided into layers, so that one can give different attributes to different layers, for the purpose of modelling different pathological conditions of the cartilages. Apart from that, hexahedral elements are opted, since they are quite accurate in predicting stresses. Besides, the computational cost is less in order to achieve the same solution with the tetrahedral elements. The way that the geometries are extracted is based on a geometry specific sweeping algorithm, that produces an initial coarse mesh that subsequently is iteratively smoothed, expanded (to fit the original triangular geometry) and refined so that the quality of the hexahedral elements is suitable for FE analysis. An illustrative example of the entire methodology concerning the femoral articular cartilage is presented in Figure 23. Finally, the resulting volumetric meshes concerning the bones and tissues are depicted in Figure 24 and Figure 25, respectively.

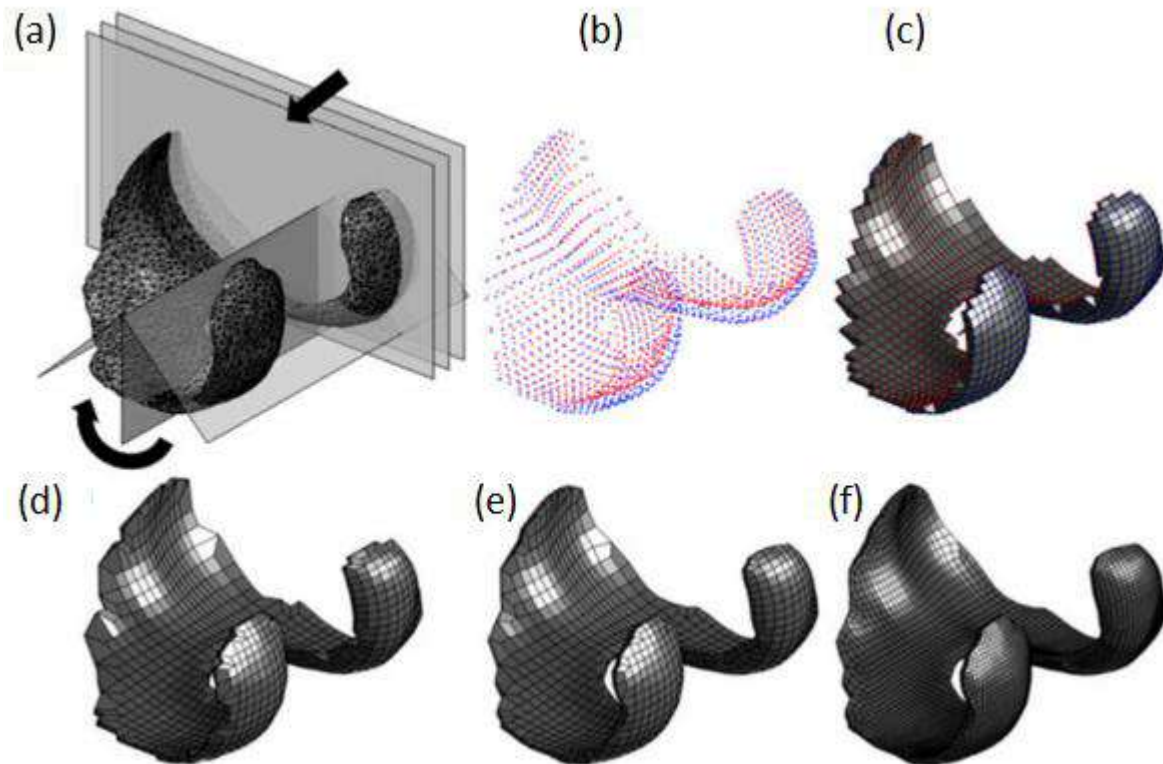
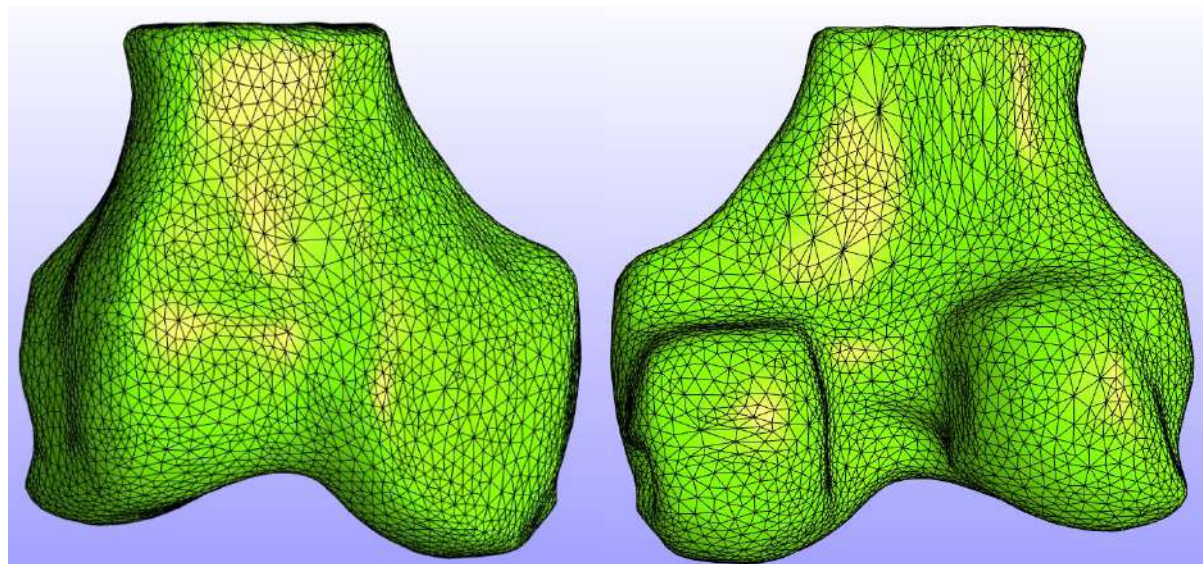
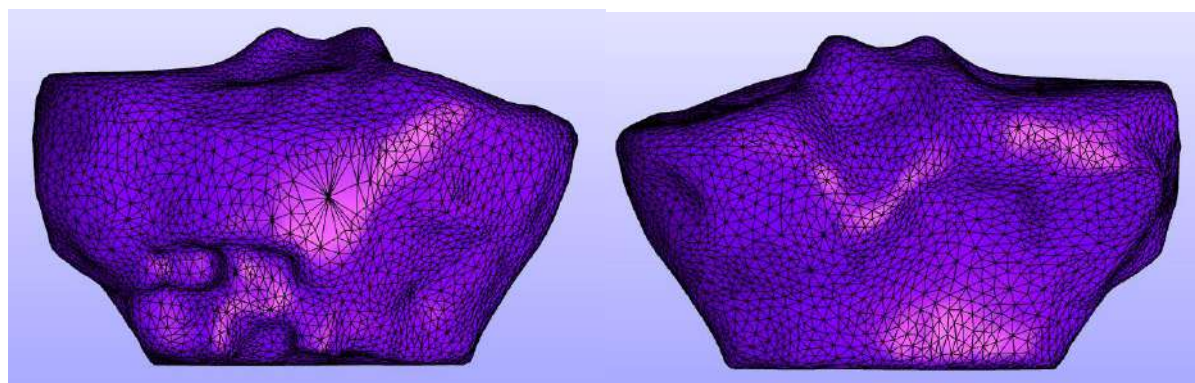


Figure 23 - Methodology of creating the volumetric mesh of the femoral articular cartilage: (a) Scanning of the initial geometry in radial and angular directions utilizing cylindrical coordinations, (b) Definition of initial nodes, (c) Creation of initial volumet



(a)

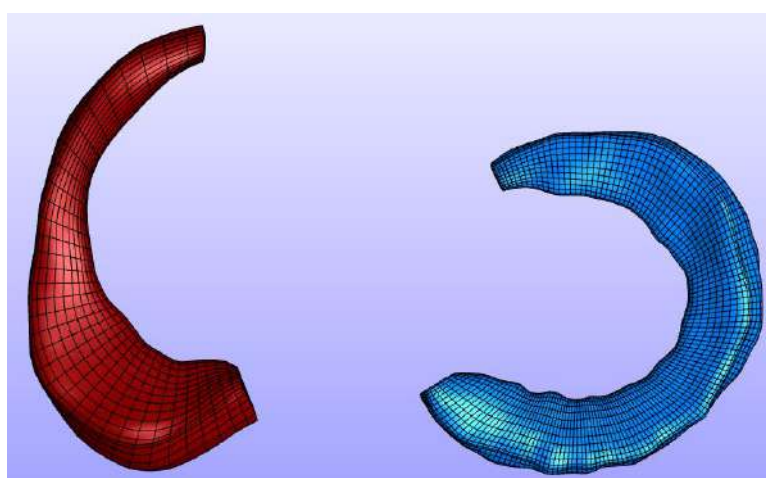
(b)



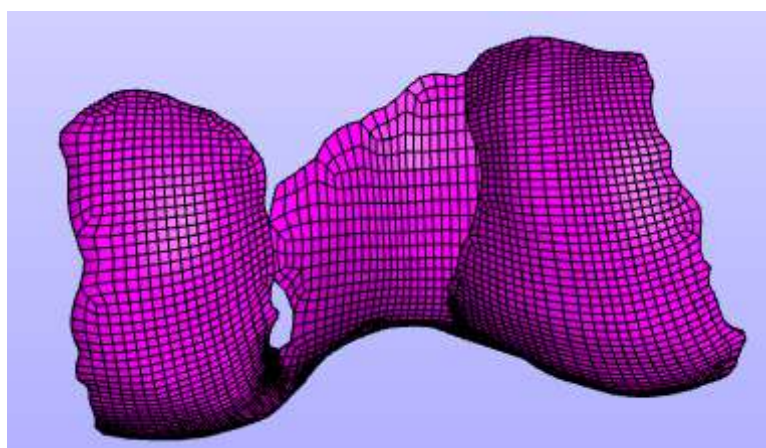
(c)

(d)

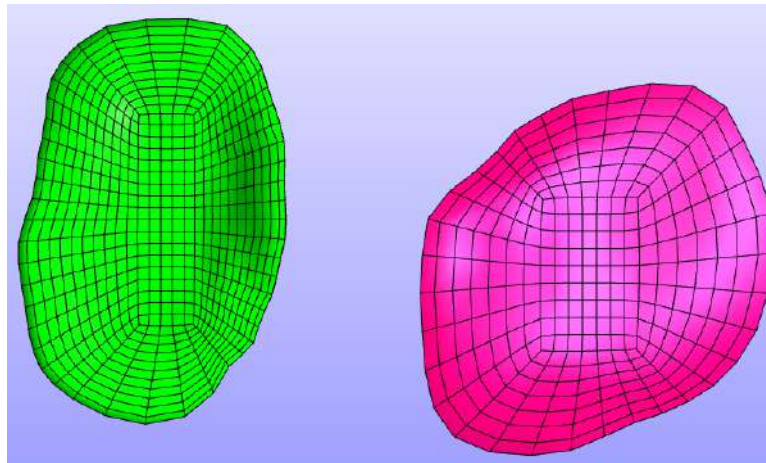
Figure 24 - Anterior (left) and posterior (right) view of the 3D volumetric mesh of: (a,b) Femur and (c,d) Tibia



(a)



(b)



(c)

Figure 25 - 3D volumetric mesh of: (a) lateral (red) and medial (blue) menisci, (b) outer layer of the femoral articular cartilage and (c) outer layer of the tibial articular cartilage; medial (green) and lateral (pink) part

4.4 Model Definition

4.4.1 Knee Anatomy

The more fundamental step for the purpose of modeling any biomechanical process is the understanding of the anatomy and the operation of each component. As a consequence, the anatomy of the knee joint is briefly given below along with a schematic representation of it, Figure 26

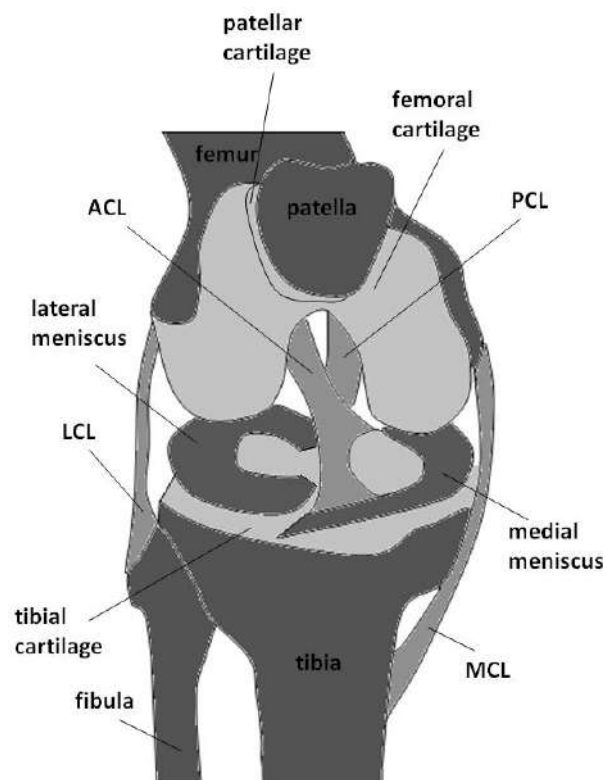


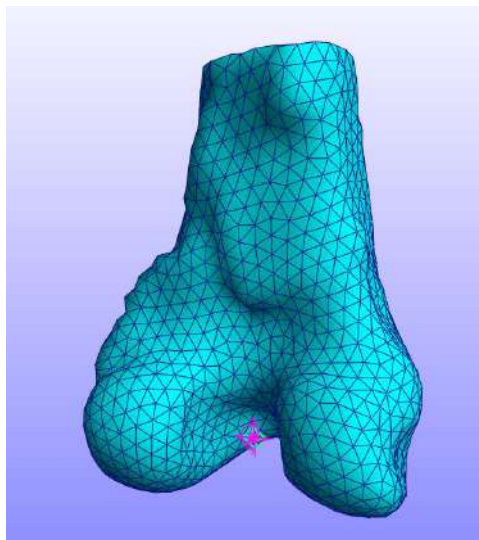
Figure 26 - Schematic illustration of knee joint anatomy

In short, the knee joint is composed of three primary bones, namely femur, tibia and patella. In addition to the main bones, fibula is located parallel to tibia and has similar length to it but it is much thinner. Articular cartilages cover each end of the bones having sponge-like tissues that enable bones to slide along each other with small friction [4]. Besides, protection of the bones is guaranteed by two crescent-shaped cartilaginous tissues that are called menisci and sited between the tibial plateau and the femoral condyle [5]. As in all synovial joints, a synovial fluid fills the gap between the bones ensuring lubrication in order to decrease wear and friction [6]. Furthermore, there are four main ligaments that stabilize the knee via resisting forces and moments [7]. These ligaments are the medial and lateral collateral ligaments (MCL and LCL, respectively), anterior and posterior cruciate ligaments (ACL and PCL, respectively) as well as patellar ligaments. The ligaments are very stiff in tension in relation to menisci and cartilages. The collateral ligaments are located on the inside (MCL) and on the outside (LCL) of the knee joint and control the sideways knee motion. The cruciate ligaments, on the other hand, are considered the most significant stabilizers concerning anteroposterior translation of the femur with respect to tibia [8]. Finally, tendons, muscles and synovial fluid are usually not included in the finite element models for simplicity.

4.4.2 Material Properties

The finite element modeling of the knee joint is a very challenging task. Out of the essential requirements, the realistic representation of the parts comprising the knee joint is critical so as to develop a reliable numerical tool which can simulate important aspects of knee function. Next, a brief description of the material models and properties concerning the bones and soft tissues that were used in the numerical model developed in FEBio software suite is given.

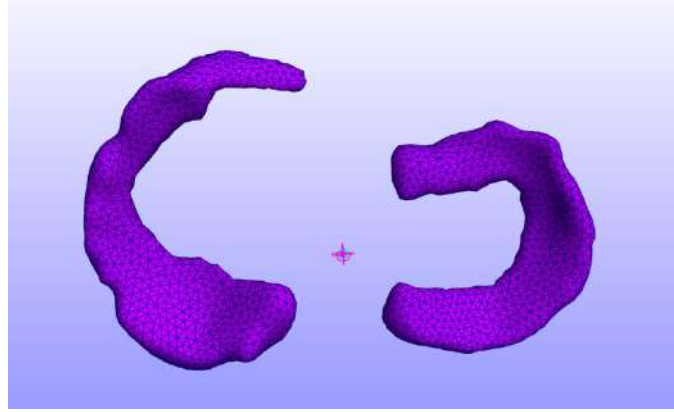
Bones



In the majority of the finite element models pertaining to the knee joint, bones are considered as rigid bodies [9,10]. This simplification is originated from their large values of density and Young's modulus as compared to soft tissues, such as menisci and cartilages. Other studies, however, define the bony structures as deformable bodies so as to estimate the stresses on them [11,12]. Finally, as it was proven by Donahue et al. [13], considering the bones with either rigid or deformable behavior has no important difference on the contact responses within the knee joint. As far as the present numerical model is concerned, only the femur and tibia were taking into account, thus, greatly reducing the number of equations in the system and as a result the computational time.

Menisci

Modeling menisci as poroelastic materials seems to be the most reasonable approach [5,9]. However, also simpler models are implemented in the literature for the description of meniscal behavior such as the consideration of a linear elastic and isotropic material [14]. Besides, since fibers run in the circumferential direction [15] anisotropy is observed that can be modeled by incorporating a linearly elastic transversely isotropic material with different Young's modulus and Poisson's ratio in the radial and axial directions [13,16]. Finally, meniscal horns attachments are commonly utilized via springs attaching each node of the meniscal horn faces to a node on the tibia [10,13,17].



In the present model, the menisci were considered as Fung orthotropic hyperelastic materials with linear springs as horn attachments. The strain energy function is given by:

$$W = \frac{1}{2} c (e^Q - 1)^2 \quad (1a)$$

with

$$Q = c^{-1} \sum_{a=1}^3 \left[2\mu_a A_a^0 : E^2 + \sum_{b=1}^3 \lambda_{ab} (A_b^0 : E) \right] \quad (1b)$$

$$\begin{bmatrix} \lambda_{11} + 2\mu_1 & \lambda_{12} & \lambda_{13} \\ \lambda_{12} & \lambda_{22} + 2\mu_2 & \lambda_{23} \\ \lambda_{13} & \lambda_{23} & \lambda_{33} + 2\mu_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{E_1} & \frac{-\nu_{12}}{E_1} & \frac{-\nu_{31}}{E_3} \\ \frac{-\nu_{12}}{E_1} & \frac{1}{E_2} & \frac{-\nu_{23}}{E_2} \\ \frac{-\nu_{31}}{E_3} & \frac{-\nu_{23}}{E_2} & \frac{1}{E_3} \end{bmatrix} \quad (1c)$$

$$\mu_1 = G_{12} + G_{31} - G_{23} \quad (1d)$$

$$\mu_2 = G_{12} - G_{31} + G_{23}$$

$$\mu_3 = -G_{12} + G_{31} + G_{23}$$

where A^0 corresponds to initial direction of material axes, E is the Green-Lagrange strain tensor, λ, μ are the Lamé's first and second parameters, respectively, E is the Young's modulus (MPa), G is the shear modulus (MPa), K is the bulk modulus like penalty parameter (MPa) and ν is the Poisson's ratio. The parameters for the representation of menisci material were adapted from [18] and reported in **Table 1**.
Deliverable D3.1

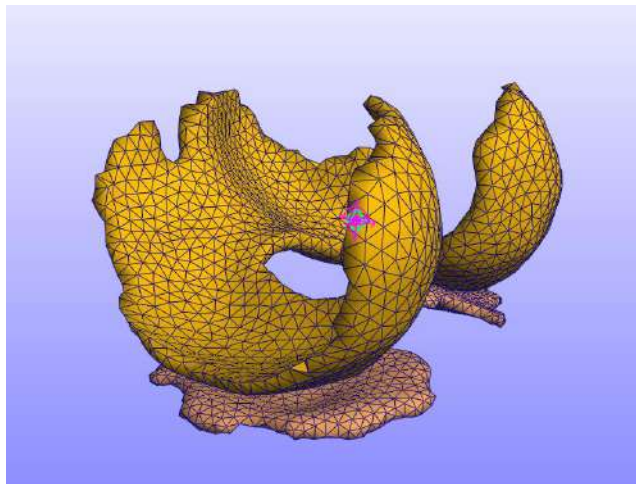
Table 1 Properties used for the menisci [18]

E_1	E_2	E_3	ν_{12}	ν_{23}	ν_{31}	G_{12}	G_{23}	G_{31}	ϵ	K
125	27.5	27.5	0.1	0.33	0.1	2	12.5	2	1	10

Ligaments

A number of element types, material models and properties have been implemented to describe ligaments, ranging from unidimensional and elastic elements to elaborate three-dimensional and hyperelastic anatomically realistic structures. A systematic review on the material models along with material properties regarding finite element analysis of the ligaments of the knee joint has been conducted by Galbusera et al. [19]. The simpler approach to model the ligaments is via spring elements [13]. Moreover, nonlinear spring elements have been implemented in the literature [14]. In the present model, the ligaments were modeled as linear springs attaching each node of one bone to another one. The spring constants were calculated similar to [10].

Articular cartilages



In the same manner as meniscal description, cartilage is composed of an interstitial fluid phase and a solid matrix. Its primary function is to reduce the contact stresses and permit motion of the surfaces by means of minimum friction. The existent models range from the isotropic hyperelastic [20] to the depth-dependent isotropic ones [21]. Additionally, a fibril-reinforced poroviscoelastic material has been used by other researchers [9] in an effort to incorporate the influence of the permeability and stiffness of the collagen fibril network.

Taking into account that the loading time is much smaller comparing to the viscoelastic time constant of cartilage [15], cartilage has also been modeled as a linear elastic isotropic material [13]. Finally, owing to the non-linearity of the cartilage during large deformations, the Mooney-Rivlin model and the Neo-Hookean hyperelastic model (a subcase of Mooney-Rivlin model) are widely used to describe cartilage behavior [10].

In the present finite element analysis, two approaches have been taken to modelling the femoral and tibial cartilages. In the first the cartilage is modelled as a simple structure with one homogeneous layer. In the second, both femoral and tibial articular cartilages have been divided into three layers in the same manner as [10]. The benefit of this is that it can better model in-vivo cartilage which is often thought of as being multi-layered. The downside is that it adds significantly to model complexity and computation time. Analyses are ongoing to evaluate the effect of this decision. In both cases, the cartilage is defined as nearly incompressible Neo-Hookean material with the strain energy function given as:

$$W = C_1 (\tilde{I}_1 - 3) + \frac{K}{2} (\ln J)^2 \quad (2)$$

Where, according to [13], $C_1 = 0.856$ MPa and $K = 8$ MPa is the bulk modulus like penalty parameter, \tilde{I}_1 represents the first invariant of the deviatoric right Cauchy-Green deformation tensor, respectively, while J is the determinant of the deformation gradient tensor (volume ratio).

In this fashion, it should be pointed out that in experimental studies such as [22] it has been demonstrated that the permeability of cartilage depends on strain. In particular, when contact takes place the region underlying the articular surface is compressed and thus permeability decreases. Given that osteoarthritis is associated with variations in permeability, a realistic description of the cartilage with a strain-dependent permeability may provide novel information on the adaptive and degenerative processes leading to osteoarthritis.

As an alternative approach for modeling cartilage, a poroelastic (biphasic) material was also utilized. In brief, in biphasic theory two immiscible, intrinsically incompressible phases are considered: a solid phase and a fluid phase [23]. Besides, the solid matrix is isotropic linear elastic and the permeability is constant. Linear biphasic theory predicts compressive creep and stress relaxation behaviors under infinitesimal strain [23]. The governing equations are the conservation of linear momentum and the conservation of mass for the mixture. As it is described in [24], under quasi-static conditions, the conservation of momentum reduces to:

$$\text{div} \boldsymbol{\sigma} + \rho \mathbf{b} = \mathbf{0} \quad (3)$$

Where $\boldsymbol{\sigma}$ is the Cauchy stress for the mixture, ρ is the mixture density and \mathbf{b} is the external mixture body force per mass. Since the mixture is porous, this stress may also be written as:

$$\boldsymbol{\sigma} = -p \mathbf{I} + \boldsymbol{\sigma}^e \quad (4)$$

Where p is the fluid pressure and $\boldsymbol{\sigma}^e$ is the effective or extra stress, resulting from the deformation of the solid matrix. Conservation of mass for the mixture requires that:

$$\text{div} (\mathbf{v}^s + \mathbf{w}) = 0 \quad (5)$$

Where \mathbf{v}^s is the solid matrix velocity and \mathbf{w} is the flux of the fluid relative to the solid matrix.

In view of relating the relative fluid flux, \mathbf{w} , to the fluid pressure and solid deformation, it is necessary to employ the equation of conservation of linear momentum for the fluid:

$$-\varphi^w \text{grad} p + \rho^w \mathbf{b}^w + \mathbf{p}_d^w = \mathbf{0} \quad (6)$$

Where φ^w is the solid matrix porosity, $\rho^w = \varphi^w \rho_t^w$ is the apparent fluid density and ρ_t^w is the true fluid density. Moreover, \mathbf{b}^w is the external body force per mass acting on the fluid, and \mathbf{p}_d^w is the momentum exchange between the solid and fluid constituents. The latter typically represents the frictional interaction

between these constituents. This equation neglects the viscous stress of the fluid in comparison to \mathbf{p}_d^w . The most common constitutive relation is:

$$\mathbf{p}_d^w = -\varphi^w \mathbf{k}^{-1} \cdot \mathbf{w} \quad (7)$$

Where the second order, symmetric tensor \mathbf{k} is the hydraulic permeability of the mixture. When combined with Eq. 6 it produces:

$$\mathbf{w} = -\mathbf{k} \cdot (\text{grad} p - \rho_T^w \mathbf{b}) \quad (8)$$

Which is equivalent to Darcy's law. In general, \mathbf{k} may be a function of the deformation.

As far as the present analysis is concerned, the constitutive relations proposed by Holmes and Mow [25] are implemented in order to describe the solid matrix elasticity and permeability under finite deformation. The hyperelastic strain-energy function for this material is given by:

$$W(I_1, I_2, J) = \frac{1}{2} c (e^Q - 1) \quad (9)$$

Hence, for the isotropic solid matrix the corresponding Cauchy stress tensor is given by:

$$\mathbf{T}^e = \frac{1}{2J} e^Q \left[(2\mu_s + \lambda_s (I_1 - 1)) \mathbf{B} - \lambda_s \mathbf{B}^2 - (\lambda_s + 2\mu_s) \mathbf{I} \right] \quad (10)$$

Where:

$$Q = \frac{\beta}{\lambda_s + 2\mu_s} \left[(2\mu_s - \lambda_s)(I_1 - 3) + \lambda_s (I_2 - 3) - (\lambda_s + 2\mu_s) \ln J^2 \right],$$

$$c = \frac{\lambda_s + 2\mu_s}{2\beta} \quad (11)$$

$\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T$ is the left Cauchy-Green tensor, I_1, I_2, I_3 its invariants, $J = \det \mathbf{F}$, λ_s and μ_s the Lamé coefficients and β the dimensionless nonlinear stiffening coefficient.

In general when the permeability tensor is isotropic:

$$\mathbf{K} = k \mathbf{I} \quad (12)$$

With k being a constant parameter. However, this assumption is only reasonable when strains are small.

The Holmes-Mow isotropic material is similar to the constant isotropic permeability material described above, except that it uses a strain-dependent permeability tensor:

$$\mathbf{K} = k(J) \mathbf{I} \quad (13)$$

Where

Deliverable D3.1

$$k(J) = k_0 \left(\frac{J - \varphi_0^s}{1 - \varphi_0^s} \right)^a e^{\frac{M}{2}(J^2 - 1)} \quad (14)$$

With k_0 corresponding to the permeability in the limit of no change in volume, $J = 1$, φ_0^s is the volume fraction of the porous solid matrix when $J = 1$ ($0 < \varphi_0^s < 1$) while a, M are nondimensional parameters. In the present analysis, the following values were utilized similar to [21,25]: $\lambda_s = 0$ MPa, $\mu_s = 0.2$ MPa, $\beta = 0.35$, $k_0 = 2.7 \times 10^{-3}$ mm⁴/Ns, $M = 2.2$, $a = 2$ and $\varphi_0^s = 0.2$

4.4.3 Definition of articulating surfaces

Contact pairs were set between tibial and femoral cartilage surfaces and between cartilages and menisci. In particular, contact pairs represent mechanical interactions between the tissue structures during simulations of joint loading. Moreover, contact surfaces were simulated by zero-friction sliding contact elements based on the low friction in synovial joints.

Figure 27 and Figure 28 show the definition of articulating surface for the medial and lateral side of the tibial and femoral cartilage.

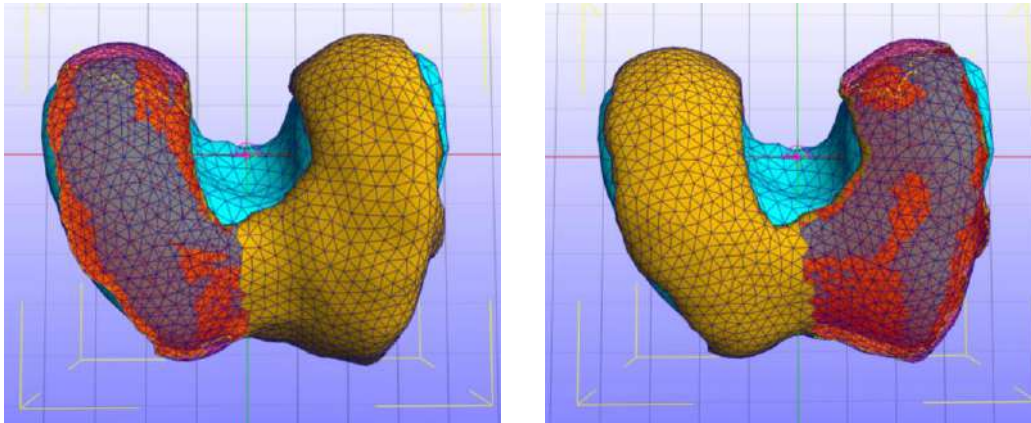


Figure 27 - Definition of medial and lateral articulating surfaces on the femur

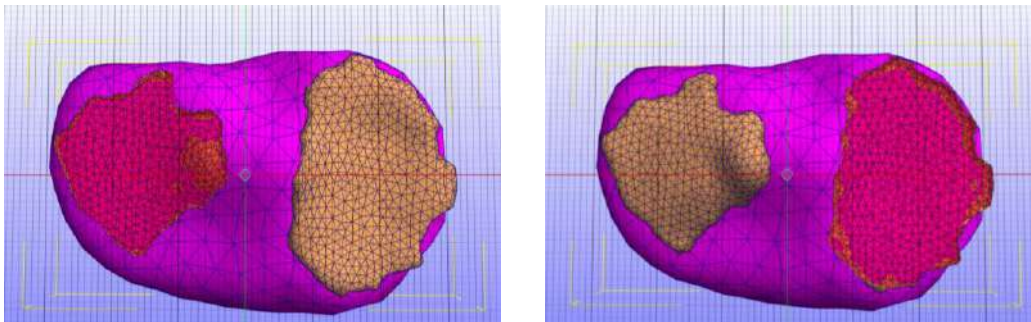


Figure 28 - Definition of medial and lateral articulating surfaces on the tibia

4.4.4 Load Application and Boundary Conditions

Part of the model requires fixed displacements to prevent rigid body motions. These essential boundary conditions are fulfilled by fixing all three components of translation and three components of rotation of the tibia. To implement the joint forces the femur and tibia are linked through three connectors which represent rigid cylindrical joints. Each joint's origin is placed in the actual tibiofemoral joint centre but represents a different joint axis. To define the connectors, two rigid materials need to be selected. In order to prescribe rotations, displacement and forces on rigid bodies, a pair of dummy femur and tibia rigid bodies are created that are used in the definition of the connectors. Using the connectors, the dummy bodies are each fixed to one of the actual bone geometries and then to each other, attaching them to the rest of the model without having a defined geometry themselves.

Forces, translations and rotations can be assigned through the connectors themselves and are applied along the defined joint axis. These natural boundary conditions can be defined at specific time steps through the curve editor. For improved convergence, the initial gradient of applied forces and rotations should be low as parts of the model require to make contact with each other. Large initial gradients would need to be compensated by small iterative time steps, slowing down simulation time.

4.4.5 Validation of FeBio contact algorithm

Validity of the frictionless contact algorithm

The validity of the contact algorithm of FEBio was firstly assessed via reproducing the results of Ateshian et al. [24]. The numerical method along with the algorithm for the frictionless contact of porous media is

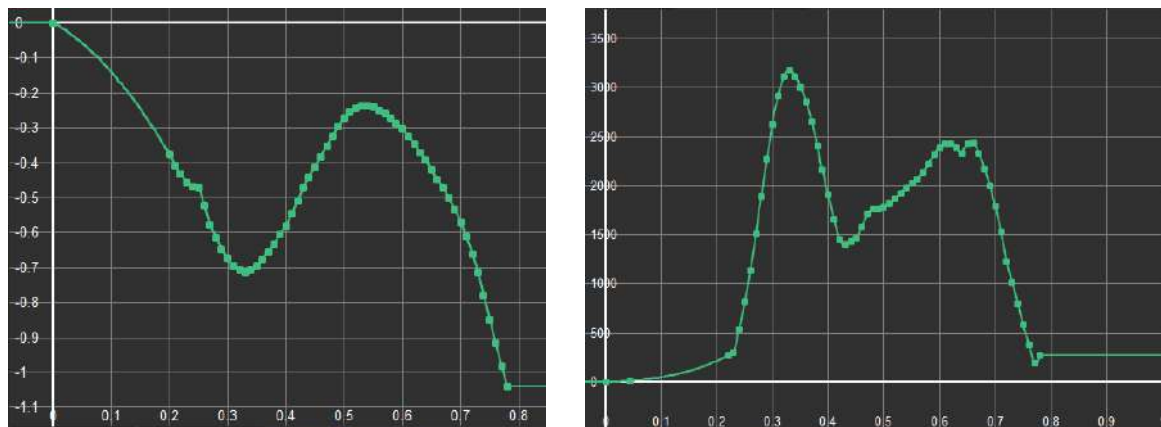


Figure 29 Prescribed rotations and forces along the simulation time steps

described in detail in [24]. Two specialized plane strain analyses were performed to verify that the contact algorithm functioned as expected and passed the patch test. In each analysis, two models were created: one, where two bodies come into contact, and the other, where a single body of the same overall geometry is loaded under the same conditions.

Case 1: Confined compression

The model consists of a slab with dimensions 12 mm x 4 mm. The sides of the slab are constrained from moving in x-direction, while the bottom surface is constrained from moving in y-direction and the fluid is free to exit from the bottom surface.

As it will be elaborated next, two cases are considered, namely the contact model, consisting of two identical slabs with dimensions 12 mm x 2 mm, Figure 30a, and the no-contact model consisting of a single slab with dimensions 12 mm x 4 mm, Figure 30b. Regarding the contact model, the top slab has 5 x 12 uniformly spaced elements while the bottom slab has 3 x 28 elements. On the other hand, the finite element mesh of the contact model consists of 3 x 40 elements. Moreover, in both cases, a mesh bias in the vertical direction is applied with the aim of accommodating the boundary layer anticipated at the free-draining bottom surface. Finally, a prescribed displacement, u_y , on the top surface is imposed having a ramp-and-hold profile, with a ramp rate of -10^{-4} mm/s and a final displacement of -0.5 mm, Figure 31.

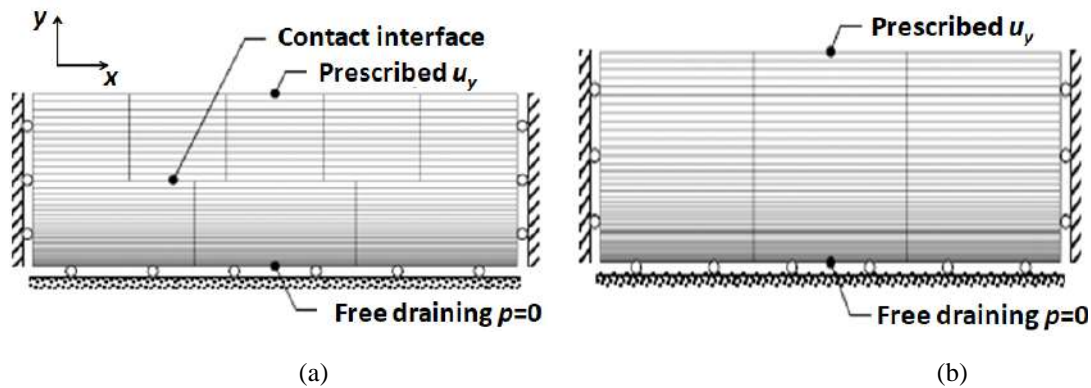


Figure 30 - Confined compression stress-relaxation analysis in plane strain: (a) The contact model consists of two slabs constrained as depicted (slab width = 12 mm, height = 2 mm) and (b) The no-contact model consists of a single slab (width = 12 mm, height = 4 mm) [24]

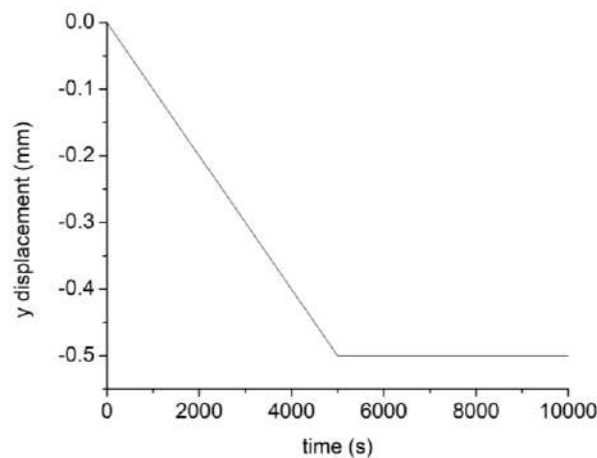


Figure 31 - The prescribed displacement, u_y , on the top surface

(a) Contact model

The two slabs were loaded in contact under a prescribed displacement on the upper body for the purpose of producing a stress-relaxation response. The time-dependent fluid pressure and vertical displacement at the nodes belonging to either contact surface were compared in Figure 32. As it can be gleaned from **Fig. 4**, the fluid pressures and vertical nodal displacements were identical over the entire width of the contact interface.

(b) No-contact model

In addition, a single slab having the same dimensions as the above combined slabs was examined under the same boundary conditions. In Figure 32, also the fluid pressure and the vertical displacement are plotted for the nodes coinciding with the contact interface of the contact model. The results for both fluid pressure and vertical displacement were identical to the single slab analysis. As a consequence, the contact algorithm of FEBio was verified. Moreover, for the sake of completeness, also indicative values of the Ateshian et al. [24] results were included in Figure 32 which confirm the correctness of the present analysis. Finally, the relaxation is evident after $t = 5000$ sec, since the fluid pressure after that time starts to decrease while, at the same time, the vertical position of the nodes corresponding in the contact region changes.

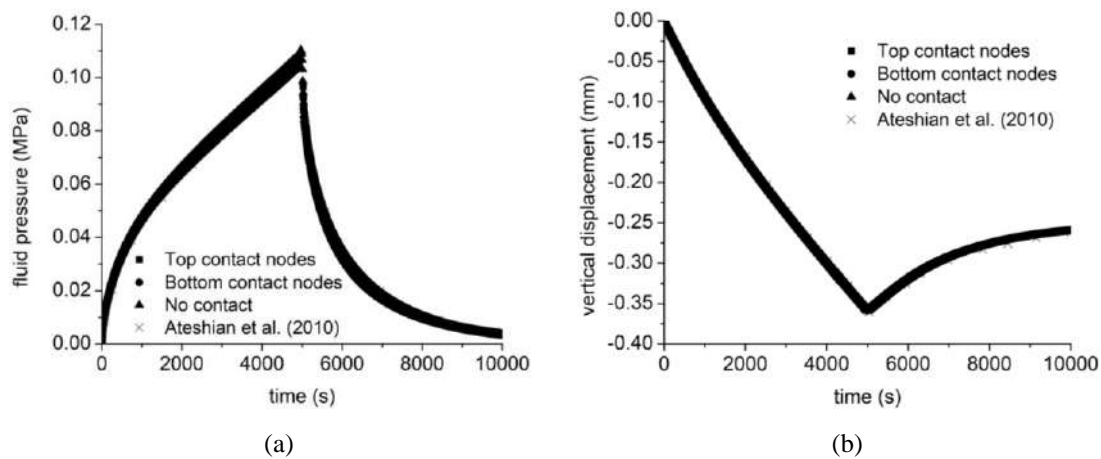


Figure 32 - Transient response of the confined compression at all the nodes of the contact surfaces in the contact model and the corresponding nodes with the same y coordinate of the no-contact model: (a) Nodal fluid pressure and (b) Nodal vertical displacement

Case 2: Unconfined compression

In the second analysis, two rectangular slabs of porous media were loaded in unconfined compression. The prescribed displacement, u_y , on the top surface has a ramp-and-hold profile, with a ramp rate of -0.4 mm/s and a final displacement of -0.4 mm. For the purpose of evaluating the patch test, a different number of elements were used in the top and bottom slabs, guaranteeing that nodes on opposing contact surfaces did not face each other directly, in the same manner as in [24], Figure 33. The two slabs have dimensions 12 mm \times 2 mm. Besides, the top slab has 5×12 uniformly spaced elements, while the bottom one has 3×28 elements, with a mesh bias in the vertical direction to accommodate the boundary layer anticipated at the free-draining bottom surface, Figure 33a. The deformed mesh is displayed at the end of the displacement ramp, $t = 1$ s, Figure 33a, and after reaching equilibrium, i.e. at $t = 10^5$ s, Figure 33a. The lateral edges of the two slabs are imposed to line up together similarly to [24]. A plot of the nodal fluid pressures at the contact interface demonstrated that the pressure distribution is identical on the two contact surfaces, Figure 33a. As a consequence, the results depicted in Figure 34 indicate that the contact algorithm successfully enforces the contact conditions, while also passing the patch test.

Similar to case 2 consideration, also the equivalent model utilizing a single slab with the same overall dimensions and subject to the same boundary conditions was also analyzed (3×28 elements, with a mesh bias in the vertical direction to accommodate the boundary layer anticipated at the free-draining bottom surface, Figure 33b). The fluid pressure at nodes coinciding with the contact interface of the corresponding contact model were plotted, Figure 34, demonstrating excellent agreement with the contact analysis and verifying the contact algorithm for this unconfined compression configuration. Finally, similarly to case 1

consideration, also indicative values of the Ateshian et al. [24] results were incorporated in Figure 34, which confirm the correctness of the present analysis.

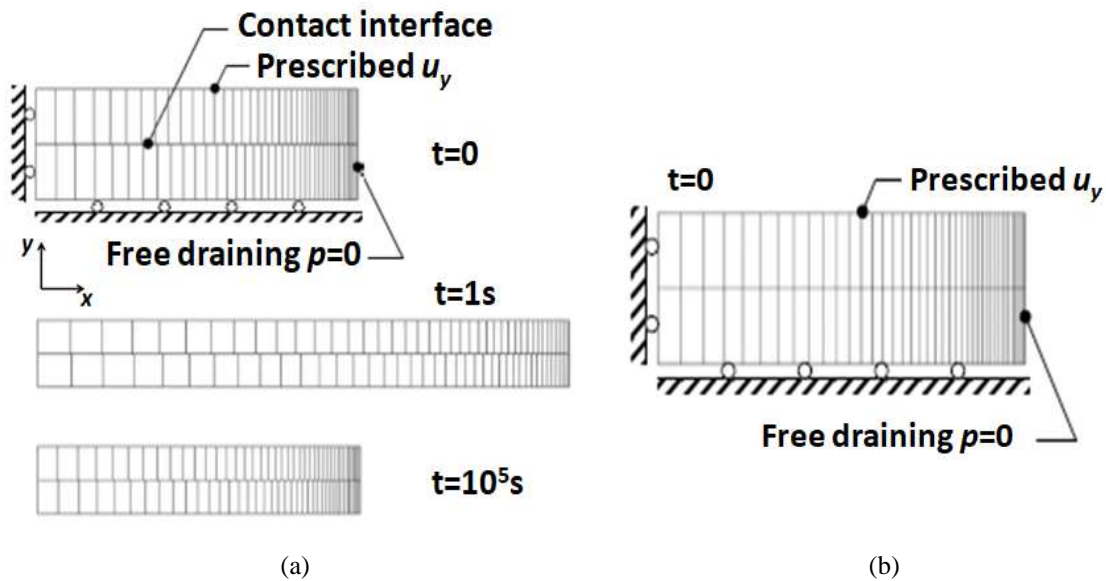


Figure 33 - Confined compression stress-relaxation analysis in plane strain: (a) The contact model consists of two slabs constrained; Mesh at the beginning of the displacement ($t = 0$ s) and deformed mesh at the end of the displacement ramp ($t = 1$ s) and after reaching equilibrium ($t = 10^5$ s), (b) The no-contact model consists of a single slab; Mesh at the beginning of the displacement ($t = 0$ s), [24]

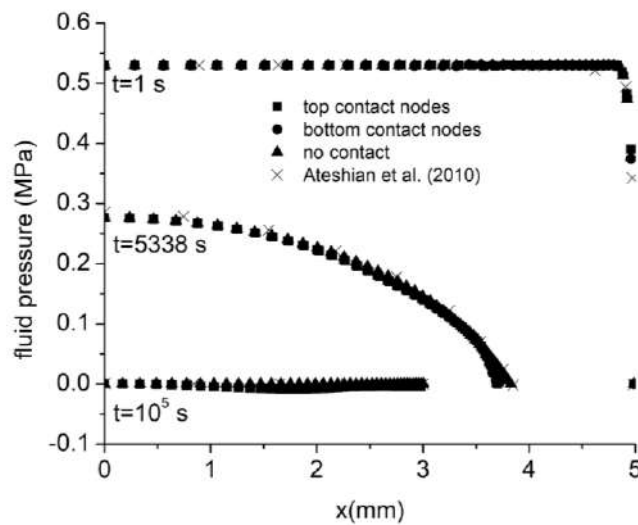


Figure 34 - Nodal fluid pressure across the nodes of the contact surfaces in the contact model and the corresponding nodes of the no-contact model

5 Digital Media

Included as part of this deliverable are a series of digital media intended to allow others to process data using the pipeline described in this deliverable. These include:

- *ModelTemplate.osim*: Generic OpenSim model file with markers attached
- *MskPreProcessing.m*: MATLAB file for pre-processing .c3d files and producing .trc and .mot files that can be interpreted by OpenSim
- *MskProcessing.m*: MATLAB file for processing the mocap and GRF data using OpenSim processes, including scaling, inverse kinematics, inverse dynamics, static optimization and joint reaction analysis
- *MATLAB additional functions*: Additional functions to allow the MskPreProcessing, MskProcessing and MskOutput to be able to run
- *ExampleModel.prv/ExampleModel.feb*: Example finite element model based on subject specific patient geometry
- *Sample Data*: Sample data that can be used when running the pipeline

6 References

- [1] H. Wang and P. Yushkevich, Multi-atlas segmentation with joint label fusion and corrective learning—an open source implementation, *Frontiers in Neuroinformatics* 7 (2013) doi:10.3389/fninf.2013.00027
- [2] V.B. Rodriguez, G.P. Sanchez, I. Oropese, E.G. Gomez and D.M. Pierce, Automated hexahedral meshing of knee cartilage structures-application to data from the osteoarthritis initiative, *Computer methods in biomechanics and biomedical engineering* 20(14) (2017) 1543-1553.
- [3] Saarakkala, S., Julkunen, P., Kiviranta, P., Mäkitalo, J., Jurvelin, J. S., & Korhonen, R. K. (2010). Depth-wise progression of osteoarthritis in human articular cartilage: investigation of composition, structure and biomechanics. *Osteoarthritis and Cartilage*, 18(1), 73–81. doi:10.1016/j.joca.2009.08.003.
- [4] Schmidt, T. A., & Sah, R. L. (2007). Effect of synovial fluid on boundary lubrication of articular cartilage. *Osteoarthritis and Cartilage*, 15(1), 35–47. doi:10.1016/j.joca.2006.06.005
- [5] Makris, E. A., Hadidi, P., & Athanasiou, K. A. (2011). The knee meniscus: Structure–function, pathophysiology, current repair techniques, and prospects for regeneration. *Biomaterials*, 32(30), 7411–7431. doi:10.1016/j.biomaterials.2011.06.037.
- [6] Claes, S., Vereecke, E., Maes, M., Victor, J., Verdonk, P., & Bellemans, J. (2013). Anatomy of the anterolateral ligament of the knee. *Journal of Anatomy*, 223(4), 321–328. doi:10.1111/joa.12087.
- [7] Scuderi G.R., & Tria A.J., *The Knee: A Comprehensive Review* (World Scientific Publishing Co. Pte. Ltd., Singapore, 2010), pp. 1-17.
- [8] Halonen, K. S., Dzialo, C. M., Mannisi, M., Venäläinen, M. S., de Zee, M., & Andersen, M. S. (2017). Workflow assessing the effect of gait alterations on stresses in the medial tibial

- cartilage - combined musculoskeletal modelling and finite element analysis. *Scientific Reports*, 7(1). doi:10.1038/s41598-017-17228-x.
- [9] Erdemir, A. (2015). Open Knee: Open Source Modeling and Simulation in Knee Biomechanics. *Journal of Knee Surgery*, 29(02), 107–116. doi:10.1055/s-0035-1564600.
 - [10] Abdullah, A. H., Rashid, H., Mahmud, J., Othman, M. F., & Ibrahim, M. W. A.-J. (2012). Effects of Screw Materials in Anterior Cruciate Ligament Reconstruction using Finite Element Analysis. *Procedia Engineering*, 41, 1614–1619. doi:10.1016/j.proeng.2012.07.358.
 - [11] Harris, M. D., Anderson, A. E., Henak, C. R., Ellis, B. J., Peters, C. L., & Weiss, J. A. (2011). Finite element prediction of cartilage contact stresses in normal human hips. *Journal of Orthopaedic Research*, 30(7), 1133–1139. doi:10.1002/jor.22040.
 - [12] Donahue, T. L.H, Hull, M. L., Rashid, M. M., & Jacobs, C. R. (2003). How the stiffness of meniscal attachments and meniscal material properties affect tibio-femoral contact pressure computed using a validated finite element model of the human knee joint. *Journal of Biomechanics*, 36(1), 19–34. doi:10.1016/s0021-9290(02)00305-6.
 - [13] Peña, E., Calvo, B., Martínez, M. A., & Doblaré, M. (2006). A three-dimensional finite element analysis of the combined behavior of ligaments and menisci in the healthy human knee joint. *Journal of Biomechanics*, 39(9), 1686–1701. doi:10.1016/j.jbiomech.2005.04.030.
 - [14] Trad, Z., Barkaoui, A., Chafra, M., & Tavares, J.M.R.S. (2018), FEM analysis of the human knee joint, Springer International Publishing.
 - [15] Mononen, M. E., Mikkola, M. T., Julkunen, P., Ojala, R., Nieminen, M. T., Jurvelin, J. S., & Korhonen, R. K. (2012). Effect of superficial collagen patterns and fibrillation of femoral articular cartilage on knee joint mechanics - A 3D finite element analysis. *Journal of Biomechanics*, 45(3), 579–587. doi:10.1016/j.jbiomech.2011.11.003.
 - [16] Mootanah, R., Imhauser, C. W., Reisse, F., Carpanen, D., Walker, R. W., Koff, M. F., et al. (2014). Development and validation of a computational model of the knee joint for the evaluation of surgical treatments for osteoarthritis. *Computer Methods in Biomechanics and Biomedical Engineering*, 17(13), 1502–1517. doi:10.1080/10255842.2014.899588.
 - [17] Gardiner, J. C., Weiss, J. A., & Rosenberg, T. D. (2001). Strain in the Human Medial Collateral Ligament During Valgus Loading of the Knee. *Clinical Orthopaedics and Related Research*, 391, 266–274. doi:10.1097/00003086-200110000-00031.
 - [18] Galbusera F., Freutel M., Dürselen L, D’Aiuto M., Croce D., Villa T., Sansone V., & Innocenti B., (2014). Material Models and Properties in the Finite Element Analysis of Knee Ligaments: A Literature Review. *Frontiers in Bioengineering and Biotechnology*, 2. doi:10.3389/fbioe.2014.00054.
 - [19] Zhang, H., Vrahas, M. S., Baratta, R. V., & Rosler, D. M. (1999). Damage to rabbit femoral articular cartilage following direct impacts of uniform stresses: an in vitro study. *Clinical Biomechanics*, 14(8), 543–548. doi:10.1016/s0268-0033(99)00010-8.
 - [20] Adouni, M., Shirazi-Adl, A., & Shirazi, R. (2012). Computational biodynamics of human knee joint in gait: From muscle forces to cartilage stresses. *Journal of Biomechanics*, 45(12), 2149–2156. doi:10.1016/j.jbiomech.2012.05.040.

- [21] Ateshian, G. A., Warden, W. H., Kim, J. J., Grelsamer, R. P., & Mow, V. C. (1997). Finite deformation biphasic material properties of bovine articular cartilage from confined compression experiments. *Journal of Biomechanics*, 30(11-12), 1157–1164. doi:10.1016/s0021-9290(97)85606-0.
- [22] Maas, S. A., Ellis, B. J., Ateshian, G. A., & Weiss, J. A. (2012). FEBio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering*, 134(1), 011005. doi:10.1115/1.4005694.
- [23] Ateshian G., Maas, S., & J. Weiss, Finite Element Algorithm for Frictionless Contact of Porous Permeable Media Under Finite Deformation and Sliding, *Journal of Biomechanical Engineering* 132(6) (2010) 061006 (13 pages).
- [24] Federico, S., Rosa, G., Herzog, W., & Wu, J., Effect of Fluid Boundary Conditions on Joint Contact Mechanics and Applications to the Modeling of Osteoarthritic Joints, *Journal of Biomechanical Engineering* 126(2) (2004) 220-225.
- [25] Zachary F. Lerner, Matthew S. DeMers, Scott L. Delp, Raymond C. Browning, How tibiofemoral alignment and contact locations affect predictions of medial and lateral tibiofemoral contact forces, *Journal of Biomechanics*, 48(16), 2015644-650,