

Model Benchmarking Phase Specifications ABI - DU02

1. Metadata

Authors:

Thor Besier

Nynke Rooks

Marco Schneider

Date: 28/02/2021

Contact: t.besier@auckland.ac.nz

2. Summary of data utilized with the M&S processes

Calibration data needed:

- AP forces and corresponding positions at all FE angles.
- IE torques and corresponding angles at all FE angles.
- VV torques and corresponding angles at all FE angles.

Selected DU02 robot data

Load - displacement data

File 1: RECALIBRATION-NaturalKneeData_StandardizedData.xlsx

Table 1. Selected robot data from File 1.

Sheet	Variable	Cells
IE Laxity	TF FE (deg)	B18:B439, O18:O458, AB18:AB435, AO18:AO388
	TF IE (deg)	D18:D439, Q18:Q458, AD18:AD435, AQ18:AQ388
	Torque TF IE (Nmm)	J18:J439, W18:W458, AJ18:AJ435, AW18:AW388
VV Laxity	TF FE (deg)	B18:B441, O18:O388, AB18:AB416, AO18:AO601
	TF VV (deg)	C18:C441, P18:P388, AC18:AC416, AP18:AP601
	Torque TF VV (Nmm)	I18:I441, V18:V388, AI18:AI416, AV18:AV601

File 2: DU02_INTACT_KE_AP_CORRECTED.xlsx

Table 2. Selected robot data from File 2.

Variable	Cells
TF FE (deg)	A2:A8981
TF AP (mm)	E2:E8981
Force TF AP (N)	W2:W8981

Table 3. Selected robot data from File 3.

Sheet	Variable	Cells
Passive Flexion	TF FE (deg)	P52:P65
	TF VV (deg)	Q52:Q65
	TF IE (deg)	R52:R65
	TF ML (deg)	S52:S65
	TF AP (deg)	T52:T65
	TF SI (deg)	U52:U65
	Torque TF FE (Nmm)	V52:V65
	Torque TF VV (Nmm)	W52:W65
	Torque TF IE (Nmm)	X52:X65
	Force TF ML (N)	Y52:Y65
	Force TF AP (N)	Z52:Z65
	Force TF SI (N)	AA52:AA65
	AP	TF FE (deg)
TF VV (deg)		AD8:AD12, AD16:AD20
TF IE (deg)		AE8:AE12, AE16:AE20
TF ML (deg)		AF8:AF12, AF16:AF20
TF AP (deg)		AG8:AG12, AG16:AG20
TF SI (deg)		AH8:AH12, AH16:AH20
Torque TF FE (Nmm)		AI8:AI12, AI16:AI20
Torque TF VV (Nmm)		AJ8:AJ12, AJ16:AJ20

	Torque TF IE (Nmm)	AK8:AK12, AK16:AK20
	Force TF ML (N)	AL8:AL12, AL16:AL20
	Force TF AP (N)	AM8:AM12, AM16:AM20
	Force TF SI (N)	AN8:AN12, AN16:AN20
VV	TF FE (deg)	AB8:AB12, AB16:AB20
	TF VV (deg)	AC8:AC12, AC16:AC20
	TF IE (deg)	AD8:AD12, AD16:AD20
	TF ML (deg)	AE8:AE12, AE16:AE20
	TF AP (deg)	AF8:AF12, AF16:AF20
	TF SI (deg)	AG8:AG12, AG16:AG20
	Torque TF FE (Nmm)	AH8:AH12, AH16:AH20
	Torque TF VV (Nmm)	AI8:AI12, AI16:AI20
	Torque TF IE (Nmm)	AJ8:AJ12, AJ16:AJ20
	Force TF ML (N)	AK8:AK12, AK16:AK20
	Force TF AP (N)	AL8:AL12, AL16:AL20
	Force TF SI (N)	AM8:AM12, AM16:AM20
	IE	TF FE (deg)
TF VV (deg)		AC7:AC11, AC15:AC19
TF IE (deg)		AD7:AD11, AD15:AD19
TF ML (deg)		AE7:AE11, AE15:AE19
TF AP (deg)		AF7:AF11, AF15:AF19

	TF SI (deg)	AG7:AG11, AG15:AG19
	Torque TF FE (Nmm)	AH7:AH11, AH15:AH19
	Torque TF VV (Nmm)	AI7:AI11, AI15:AI19
	Torque TF IE (Nmm)	AJ7:AJ12, AJ15:AJ19
	Force TF ML (N)	AK7:AK11, AK15:AK19
	Force TF AP (N)	AL7:AL11, AL15:AL19
	Force TF SI (N)	AM7:AM11, AM15:AM19

Anatomical landmarks data

Files:

- DU02_HipBall.txt
- DU02_fem_GSPts.txt
- DU02_tib_GSPts.txt

Registration data (probed bone points)

Files:

- DU02_fem_Bone.txt
- DU_fem_perimeter.txt
- DU02_tib_bone.txt

3. Overview of target M&S outputs and M&S processes

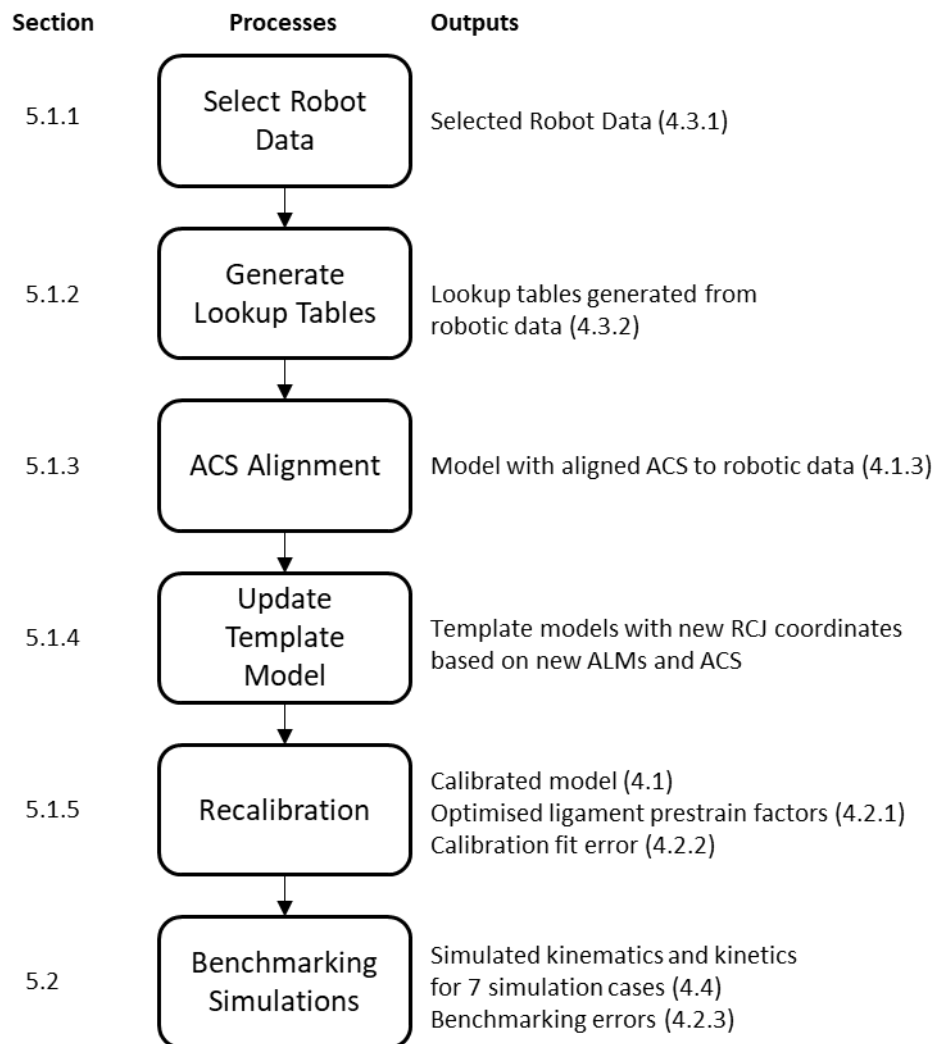


Figure 1. Overview of the model benchmarking processes and the target outputs. In summary, the robot data for recalibration and benchmarking will be selected and processed into lookup tables. Anatomical coordinate systems (ACS) will be calculated and implemented in the knee model after which the model will be calibrated against the robot data. Once calibrated, benchmarking simulations will be performed.

4. Detailed descriptions of target M&S outputs

4.1 Complete re-calibrated model and benchmarked model

To ensure consistency of Model Benchmarking (MB) with the Model Calibration (MC) phase, we will use the same geometries as described in section 4 of the ABI MC Protocol Deviations document. These geometries were updated during MC and consist of triangulated surface meshes of the bones (femur, tibia, fibula, patella) and tetrahedral meshes of the soft-tissues, including the femoral cartilage, tibial cartilages, patellar cartilage, the medial meniscus (mMen), lateral meniscus (lMen), medial collateral ligament (MCL), lateral collateral ligament (LCL), posterior cruciate ligament (PCL), and anterior cruciate ligament (ACL). For more information on how this model was built, please refer to the ABI MC phase protocol deviations document.

The complete re-calibrated model and benchmarking models will be provided as *.feb files together with their corresponding *.log and *.xplt files.

4.1.1 Anatomy

The model will consist of the tibiofemoral joint. The patella will also be present in the model as a rigid body, but will be fixed in all degrees of freedom and will not have any interactions with the other structures in the model.

The tibiofemoral joint structures present in the re-calibrated model will be the femur, tibia, fibula, femoral cartilage, medial tibial cartilage, lateral tibial cartilage, mMen, lMen, ACL, PCL, MCL, and LCL.

In the model, the bones will consist of triangulated surface meshes. The soft tissues, including the cartilage, ligaments, and menisci will consist of tetrahedral meshes. The menisci will be attached to the tibia with 36 linear springs each that have a stiffness of 1 N/mm.

The anatomy of the individual components of the re-calibrated knee model will be provided as meshes in *.feb, *.ele, *.node and *.fsprj format.

4.1.2 Mechanical properties

As in the MC phase, the bones and cartilages will be modelled as rigid bodies.

The ligaments will be modelled using a prestrain elastic Neo-Hookean material model. Each ligament will be prescribed its own Young's modulus value ({LCL, MCL, ACL, PCL}: [280, 224, 123, 168] Nmm⁻¹) based on the literature values (Orozco et al., 2018) and a prestrain stretch factor (Maas et al., 2016) which will be optimised individually. The menisci will be modelled with a Fung orthotropic material model.

The mechanical properties including the material models used, specific material model parameters, and the calibrated prestrain values, will be provided in the complete re-calibrated model and benchmarking model *.feb files. They will also be described separately in *.txt files.

4.1.3 Coordinate systems

The anatomical coordinate system (ACS) will be the same one used in the MC phase, based on the Grood and Suntay coordinate system (Grood and Suntay, 1983) and will be implemented via the rigid cylindrical joint (RCJ) facility in FEBio:

- For the tibial ACS, the x-axis will point laterally, the y-axis will point anteriorly, and the z-axis will point superiorly.
- For the femoral ACS, the x-axis will point laterally, the y-axis will point anteriorly, and the z-axis will point superiorly.
- A separate RCJ will be used to implement the floating axis of the Grood & Suntay Knee Joint Coordinate System (JCS). FE points laterally, VV points anteriorly, IE points superiorly

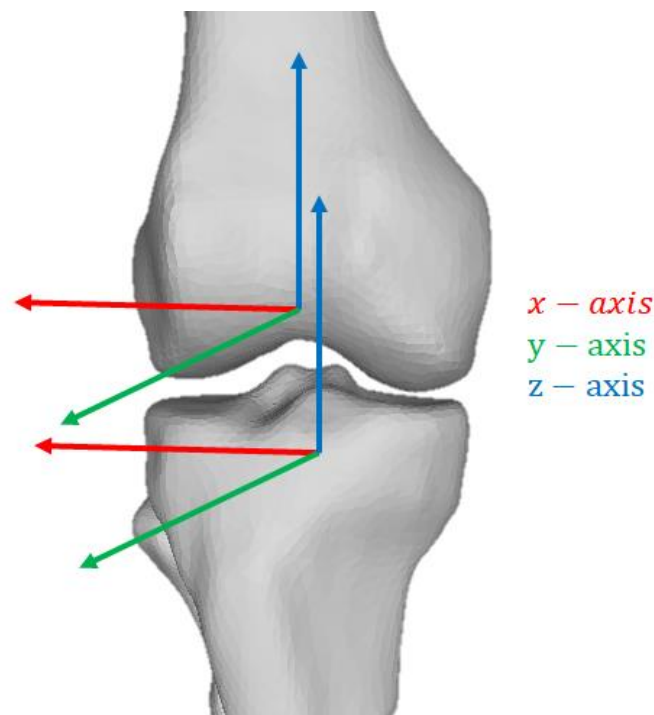


Figure 2. Local ACS directions for the femur and tibia.

All intermediate outputs, including the scripts used to find the anatomical landmarks (ALMs), *.txt files containing the ALM node numbers for the femur and tibia, and scripts for calculating and defining the ACS of the model for the RCJ will be provided.

Two ALM node number files will be provided, one for the femur and one for the tibia/fibula. The contents of the files will be provided in the following order.

Femur ALM node numbers:

- The femoral origin, described as the most distal point on the posterior side of the femur (a.k.a. Point 3 in DESCRIPTION-NaturalKneeData_DataDescriptionStandard)
- The most posterior point on the articular surface of the medial femoral condyle (a.k.a. Point 4)

- The most posterior point on the articular surface of the lateral femoral condyle (a.k.a. Point 5)
- Femoral head anterior (F1)
- Femoral head posterior (F2)

Tibia ALM node numbers:

- Medial plateau dwell (a.k.a. Point 1)
- Lateral plateau dwell (a.k.a. Point 2)
- Medial spine of tibial eminence (a.k.a. Point 3)
- The tibial origin, described as the center of the tibial eminence (a.k.a. Point 4)
- Closest node to the center of the distal intramedullary canal (T1)
- Node opposite to closest node to center of distal intramedullary canal (T2)

4.1.4 Loading and boundary conditions (for the simulation cases)

The loading and boundary conditions for the recalibration and benchmarking simulations will be provided.

The loading and boundary conditions will be provided for the following benchmarking simulation cases:

- Passive flexion from 0 to 90 degrees (with ACL)
- Anterior-posterior translation (no ACL) at ~19.1 degrees knee flexion
- Anterior-posterior translation (no ACL) at ~55.7 degrees knee flexion
- Internal-external rotation (no ACL) at ~17.7 degrees knee flexion
- Internal-external rotation (no ACL) at ~56.0 degrees knee flexion
- Varus-valgus laxity (no ACL) at ~15.4 degrees knee flexion
- Varus-valgus laxity (no ACL) at ~58.6 degrees knee flexion

For all simulation cases, the *.feb file and the corresponding *.log and *.xplt file will be provided. Also an overview (*.txt file) of the settings per simulation will be provided.

4.1.5 Interactions between model components (contacts, ties)

Cartilages will be modelled with the same rigid body material as their corresponding bones and therefore no contact implementation is required to maintain contact.

A sliding-elastic contact will be used for the contact between the tibial cartilage and femur cartilage. It will also be used for contact between any cartilage and the menisci.

Ligaments will be tied to their corresponding bones with a nodal rigid-tied interface.

A rigid cylindrical joint between the femur and the tibia, using two imaginary rigid bodies will be used to prescribe the flexion - extension rotation in the knee joint (Erdemir and Sibole, 2010).

The interactions will be provided in the complete calibrated model *.feb file and will also be described separately in *.txt files.

4.2 Intermediate and final calibration outcomes

4.2.1 Re-calibrated parameters

The prestrain values of the ACL, PCL, MCL and LCL that will be optimised during the calibration phase using the ~16.5 degrees and ~80 degrees robot data will be provided in a *.xlsx file.

4.2.2 Calibration fit error

The calibration error (a.k.a. Total error) to be minimized will be an error that depends on a number of factors including if the model converged, if the AP position after prestretch was in the specified range, if the model ran to completion, and how far it converged.

The process for obtaining the Total error is detailed in section 5.1.5.

The optimised Total errors and Normalised errors (used to calculate the total error) obtained through each optimiser will be provided in a *.xlsx file.

4.2.3 Benchmarking error

Benchmarking errors will be reported as the RMSE between the simulated kinematics (position and angle) and robot kinematics, and the simulated kinetics (forces and moments) and robot kinetics, in all converged time steps for all 6 degrees of freedom (medial translation, posterior translation, superior translation, flexion, valgus rotation, internal rotation). Figures will be provided as *.png files and data in a *.xlsx file.

4.2.4 Changed model components

During the recalibration and benchmarking phase, only the prestrain stretch values of the ligaments will be changed. These will be provided as described in 4.2.2.

4.3 Intermediate and final outcomes of analysis of experimental load cases

4.3.1 Source data (as extracted from earmarked data set)

The robot data that will be used for the recalibration and benchmarking simulations will be provided in their original form (*.csv file).

4.3.2 Processed data (as analyzed to make ready for use in simulations)

Lookup tables consisting of displacements at any applied load will be used for recalibration. The lookup tables will be obtained from the earmarked robot data through polynomial fitting a response surface and evaluating the response surface.

The equations for the fitted response surfaces used to interpolate the robot data will be provided as *.txt files:

- AP_poly54_fit.txt
- IE_poly54_fit.txt
- VV_poly54_fit.txt

The lookup tables generated from the response surfaces will be provided as *.npy files:

- AP_forces_80.npy

- AP_positions_80.npy
- IE_angles_80.npy
- IE_moments_80.npy
- VV_angles_16_5.npy
- VV_moments_16_5.npy

The load curves used for the prescribed displacement/rotation for the simulation cases will be provided as *.txt files and figures provided as *.png files.

4.4 Simulation cases

4.4.1 Loading and boundary conditions

The loading and boundary conditions for all simulations will be provided as plots in *.png files and next to that their corresponding *.feb, *.log and *.xplt files will be provided per simulation.

The simulations will be divided into five groups:

1. Passive flexion from 0 to 90 degrees (with ACL)
2. Anterior-posterior translation (no ACL) at ~19.1 degrees knee flexion
3. Anterior-posterior translation (no ACL) at ~55.7 degrees knee flexion
4. Internal-external rotation (no ACL) at ~17.7 degrees knee flexion
5. Internal-external rotation (no ACL) at ~56.0 degrees knee flexion
6. Varus-valgus laxity (no ACL) at ~15.4 degrees knee flexion
7. Varus-valgus laxity (no ACL) at ~58.6 degrees knee flexion

4.4.2 Target metrics for predictions

The RMSE between the kinematics and kinetics in the robot the model simulation results will be plotted and provided in *.png with data provided in *.txt files.

4.4.3 Numerical analysis settings

Static analyses will be used for the simulations. The settings can be found in the *.feb files per simulation.

4.4.4 Anticipated results

The tibiofemoral kinematics (6 DoF) during the simulations will include the model's anterior position (simulation cases will be plotted and provided as a *.png file), joint distraction (mm), external rotation (rad), flexion rotation (rad), medial position (mm), and valgus rotation (rad) plotted against simulation time-step. These will be overlaid with the robot data.

The RMSE between the robot force and the simulated force in the model will be provided in *.txt files. The simulation results will be available in the *.log and *.xplt files per simulation.

5. Detailed descriptions of M&S processes

5.1 Steps to calibrate the models

5.1.1 Select robot data

We will recalibrate the knee against the robot data in two positions:

1. knee at ~16.5 degrees flexion.
2. knee at ~80 degrees knee flexion.

As our workflow involves fitting a response surface to the robot data, we will use the VV robot data for the recalibration at ~16.5 degrees and IE robot data for recalibration at ~80 degrees from File 1 (Table 1). For AP, we will use the robot data from File 2 (Table 2) to allow for more data points to ensure a good fit.

5.1.2 Generate lookup tables

Robot data will be fitted with a response surface polynomial in MATLAB to smooth the calibration data and allow for interpolation between the data points.

The script (*DU02_Robot_data_lookup_table_generation.py*) will be used to generate lookup tables to allow for rapid error calculations during calibration.

5.1.3 Anatomical coordinate system alignment

The ACS of the model will be registered to match the ACS of the robot data, which is based on the Grood and Suntay coordinate system (Grood and Suntay, 1983). The local ACS for each bone is defined by ALMs which will be obtained using the process below:

1. Register the femur mesh (*raw_nodecoordinates_DU02.txt*) onto the robot probed bone points including:
 - a. *DU02_fem_Bone.txt*.
 - b. *DU_fem_perimeter.txt*.
 - c. *DU02_fem_GSPts.txt*.
 - d. *DU02_HipBall.txt*.
2. Register the tibia mesh to the robot probed bone points including:
 - a. *DU02_tib_bone.txt*
 - b. *DU02_tib_GSPts.txt*
3. Find the closest bone mesh nodes to the probed ALMs:
 - a. *DU02_fem_GSPts.txt*.
 - b. *DU02_tib_GSPts.txt*.
4. Define the Grood and Suntay ACS as in 4.1.3. Please note:
 - a. For the femur, the proximal point is defined as the midpoint of a line drawn from femoral head anterior F1 to the femoral head posterior F2.
 - b. For the tibia, the distal point is defined as the proportion of the distance between a line drawn from the Closest node to the center of the distal intramedullary canal (T1) to the center of the distal intramedullary canal (Point 6), and a line drawn from T1 through Point 6 to the node opposite to closest node to center of distal intramedullary canal (T2).

This process will be done by modifying the script *DU02_ACS_alignment_model2.py*.

5.1.4 Updating the template optimisation model

We will use the template models from the MC phase.

The template files that will be used for recalibration include:

- Base template model (defines control settings, geometry, initial orientation, initial loads, and boundary conditions, and applies prestrain):
 - DU02_KM2.feb
- Template for calibration at ~16.5 degrees:
 - DU02_VV.feb
- Template for performing flexion to ~80 degrees:
 - DU02_Extension_to_Flexion_80.feb
- Templates for calibration at ~80 degrees:
 - DU02_IE.feb
 - DU02_AP.feb

The RCJ in the base template file (DU02_KM2.feb) will be updated with the new ALMs and ACS obtained from 5.1.3.

The control settings will remain as in the MC phase with the following parameters:

- max_refs = 100 instead of 200
- rtol was deleted
- lstol changed from 0.5 to 0.9
- Timestepper:
 - min step size (dtmin) = 0.02 instead of 0.01
- Max retries = 3 instead of 5

The loads and boundary conditions will remain as in the MC phase with no additional axial force applied during the recalibration simulations at ~16.5 degrees (VV) and ~80 degrees (AP and IE).

The RCJ parameters will remain as in the MC phase with the following parameters:

- Tolerance = 0
- gaptol = 0.01
- Angtol = 0.0001
- Force_penalty = 10000
- Moment_penalty = 3000000

The sliding-elastic contacts will remain as in the MC phase with the following parameters:

- contact type = "sliding-elastic"
- laugon = 0
- tolerance = 0
- gaptol = 0
- penalty = 1 OR 10
- auto_penalty = 0
- two_pass = 1
- search_tol = 0.01
- symmetric_stiffness = 0
- search_radius = 0.005

- seg_up = 0
- tension = 0
- minaug = 0
- maxaug = 40
- fric_coeff = 0
- smooth_aug = 0
- node_reloc = 0
- knmult = 0

For specific details on how the template model was obtained, please refer to the MC phase protocol deviations document.

5.1.5 Recalibration Optimisation

We will recalibrate using the same objective function and optimisation methods in the MC phase except we will only use three optimisation algorithms: TNC, L-BFGS-B, and Powell. We will exclude using the Nelder-Mead optimisation algorithm as it performed poorly during MC.

This will be done using the scripts prepared in the MC phase:

- DU02_Optimisation_Powell.py
- DU02_Optimisation_TNC.py
- DU02_Optimisation_LBFGSB.py

Five different sets of initial prestrain factor values will be used (Table 4), along with two different cartilage contact penalty factors (set to either 1 or 10), and three different optimisation algorithms (TNC, L-BFGS-B, and Powell) resulting in 30 separate optimisations (5 x 2 x 3). These will be run on high performance computers (HPCs).

Table 4. Five sets of initial prestrain values for recalibration.

Ligament / Set	1	2	3	4	5
ACL	0.88	0.80	0.88	0.88	0.88
PCL	1.13	1.13	1.20	1.13	1.13
MCL	0.97	0.97	0.97	0.89	0.97
LCL	0.97	0.97	0.97	0.97	0.93

5.1.5.1 Objective function simulations:

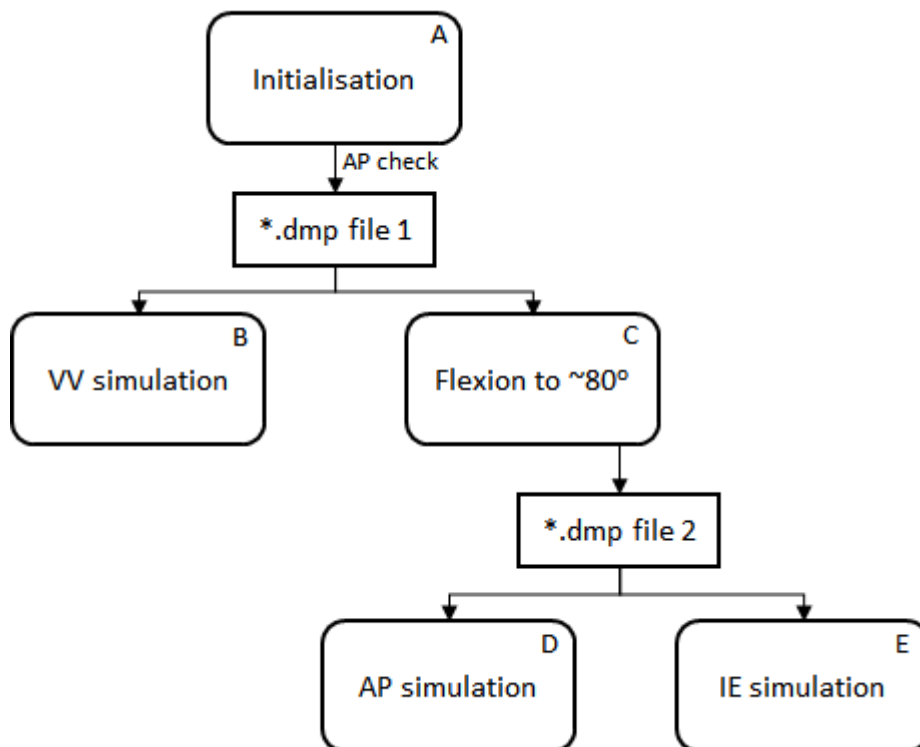


Figure 3. Overview of the 5 simulations (A, B, C, D, and E) within the objective function.

- A. The objective function will begin by initialising the model in the imaging state (~16.5 degrees of knee flexion) and applying the prestrain. A restart *.dmp file (*.dmp file 1) will then be created.
- This initialisation will consist of two timesteps of 0.1 (i.e. 0.0 - 0.2).
 - Prestrain will be applied using the following load curve:


```

<loadcurve id="1, 2, 3 & 4" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
</loadcurve>

```
- B. The restart facility in FEBio will be used to restart the simulation from the end of simulation A (*.dmp file 1) and perform the VV simulation.
- An AP check will be performed (see section 5.1.5.2)
 - This simulation will consist of ten additional timesteps of 0.1 (i.e. 0.2 - 1.2).
 - The VV moment will be applied using the following load curve:


```

<loadcurve id="9" type="linear">
  <point>0,0</point>
  <point>0.45,0</point>
  <point>0.7,5000</point>
  <point>1.2,-5000</point>
</loadcurve>

```

- d. The flexion angle will be fixed with the following load curve:
- ```
<loadcurve id="5" type="linear">
 <point>0,0</point>
 <point>4,0</point>
</loadcurve>
```
- C. The simulation will be restarted from the end of simulation A (\*.dmp file 1) to flex the knee to ~80 degrees. This will be done simultaneously with simulation B via multithreading in python. Once complete a restart \*.dmp file (\*.dmp file 2) will be created.
- a. This simulation will consist of ten additional timesteps of 0.1 (i.e. 0.2 - 1.2).
- b. Flexion will be prescribed using the following load curve:
- ```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.3,0</point>
  <point>0.8,0.5*Flex80</point>
  <point>1.2,Flex80</point>
  <point>4,Flex80</point>
</loadcurve>
```
- D. The simulation will be restarted from the end of simulation C (*.dmp file 2) to perform the AP simulation.
- a. This simulation will consist of ten additional timesteps of 0.1 (i.e. 1.2 - 2.2).
- b. An AP force will be applied using the following load curve:
- ```
<loadcurve id="10" type="linear">
 <point>0,0</point>
 <point>1.45,0</point>
 <point>1.7,50</point>
 <point>2.2,-50</point>
</loadcurve>
```
- c. The flexion angle will be fixed by the following load curve:
- ```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.3,0.0</point>
  <point>1.2,Flex80</point>
  <point>4,Flex80</point>
</loadcurve>
```
- E. Simultaneously to simulation D, the simulation will be restarted from the end of simulation C (*.dmp file 2) to perform the IE simulation.
- a. This simulation will consist of ten additional timesteps of 0.1 (i.e. 1.2 - 2.2).
- b. An IE moment will be applied using the following load curve:
- ```
<loadcurve id="7" type="linear">
 <point>0,0</point>
 <point>1.45,0</point>
 <point>1.7,5000</point>
 <point>2.2,-5000</point>
</loadcurve>
```

- c. The flexion angle will be fixed with the following load curve:
- ```

<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.3,0</point>
  <point>1.2,Flex80</point>
  <point>4,Flex80</point>
</loadcurve>

```

For all simulations (A, B, C, D, and E) the tibia will be fixed in all degrees of freedom, and the femur will be free in all degrees of freedom. Knee flexion will be prescribed in the RCJ for all simulations, which will consequently constrain the femur in the flexion-extension axis. The flexion angle, Flex80, will be the angle in radians that the knee needs to be flexed from the initial position to 80 degrees. This will be calculated after the ACS has been defined, but was 1.281 rad in the MC phase.

The input variables into the objective function will be the prestrain factors of the ACL, PCL, MCL and LCL.

5.1.5.2 Objective function minimisation error:

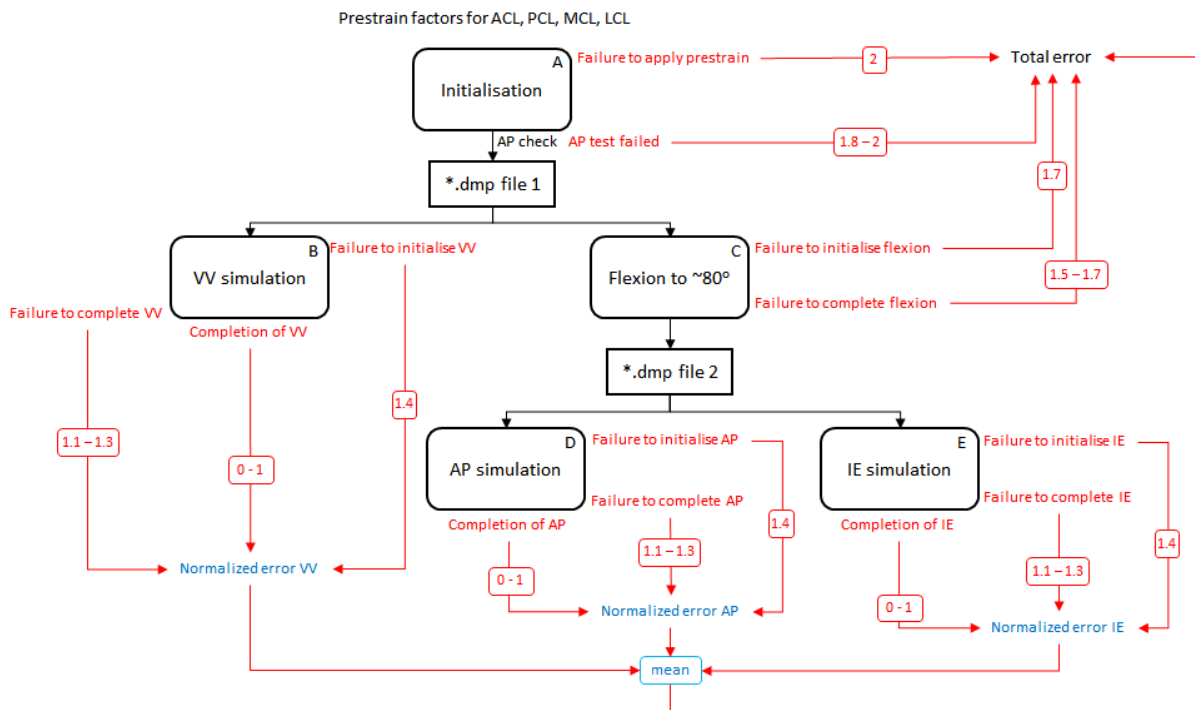


Figure 4. Schematic of the objective function including error ranges.

The conditions within the objective function and the ranges of the minimisation error that will be returned by the objective function are shown in Table 5.

Table 5. Error values and ranges during optimization for different conditions.

| Condition | Error Type | Error Value/Range |
|-------------------------------|---------------------|-------------------|
| Failure to apply prestrain | Total error | 2 |
| AP test failed* | Total error | 1.8 - 2 |
| Failure to initialise flexion | Total error | 1.7 |
| Failure to complete flexion | Total error | 1.5 - 1.7 |
| Failure to initialise VV | Normalised error VV | 1.4 |
| Failure to complete VV** | Normalised error VV | 1.1 - 1.3 |
| Failure to initialise AP | Normalised error AP | 1.4 |
| Failure to initialise IE | Normalised error IE | 1.4 |
| Failure to complete AP** | Normalised error AP | 1.1 - 1.3 |
| Failure to complete IE** | Normalised error IE | 1.1 - 1.3 |
| Completion of VV*** | Normalised error VV | 0 - 1 |
| Completion of AP*** | Normalised error AP | 0 - 1 |
| Completion of IE*** | Normalised error IE | 0 - 1 |

* If the AP position is greater than -15.4 mm then the “AP test failed” condition is met. The Total error will be returned according to equation 1.

** When a “Failure to complete” condition is met, the Normalised error will be set according to Table 6.

*** When a “Completion” condition is met, the Normalised error will be calculated by normalizing the RMSE between the simulated position/angle and the robot position/angle in all converged time steps to the maximum position/angle in the robot data (Table 7).

$$\text{AP test failed Total error} = 1.8 + (((|-15.4 - \text{AP position}|)/15.4)/5) \quad (1)$$

Table 6. Normalised errors for “Failure to complete” conditions.

| Condition | Max converged time | Normalised error* |
|-----------------------------|----------------------|-------------------|
| VV failed to complete | $0.2 \leq t < 0.45$ | 1.200 |
| | $0.45 \leq t < 0.70$ | 1.175 |
| | $0.70 \leq t < 0.95$ | 1.150 |
| | $0.95 \leq t < 1.20$ | 1.125 |
| AP or IE failed to complete | $1.95 \leq t < 2.20$ | 1.200 |
| | $1.70 \leq t < 1.95$ | 1.175 |
| | $1.45 \leq t < 1.70$ | 1.150 |
| | $1.20 \leq t < 1.45$ | 1.125 |

* If the “Failure to complete” condition is met in consecutive objective function iterations, an adjustment factor will be applied to the normalized error according to equation 2.

$$\text{Normalised error} = \text{Normalised error} * (1 + (0.001 * (N - 1))) \quad (2)$$

where N is the number of consecutive iterations where the “Failure to complete” condition is met.

Table 7. Normalization values for MC and MB phase. Note the small difference due to only using the processed data in MB.

| Simulation case | MC Normalisation value | MB Normalisation value |
|-----------------|------------------------|------------------------|
| AP at ~80° | 22.079 mm | 22.263 mm |
| IE at ~80° | 0.52330 rad | 0.4793 rad |
| VV at ~ 16.5° | 0.19588 rad | 0.19588 rad |

Unless any of the first four conditions in Table 5 are met where the “Total error” is returned immediately, the “Total error” will be calculated using equation 3. This is the error that will be minimized during re-calibration.

$$\text{Total error} = (\text{Normalised error AP} + \text{Normalised error IE} + \text{Normalized error VV})/3 \quad (3)$$

5.1.6 Write recalibration outcomes to file

Recalibration will result in optimised prestrain factors for the ACL, PCL, MCL, and LCL. These will be written into the template model (DU02_KM2.feb) used in the recalibration for further benchmarking simulations. The optimised prestrain factors will also be written to a *.txt file.

5.2 Steps to implement load cases as loading and boundary conditions

New models will be generated from the optimised template created in 5.1.6, with simulation case specific loads and boundary conditions.

5.2.1 Default control settings

The benchmarking simulations will consist of 20 time steps of 0.1 with the following control settings:

```
<Control>
  <time_steps>20</time_steps>
  <step_size>0.1</step_size>
  <max_refs>100</max_refs>
  <max_ups>0</max_ups>
  <diverge_reform>1</diverge_reform>
  <reform_each_time_step>1</reform_each_time_step>
  <dtol>0.01</dtol>
  <etol>0.01</etol>
  <lstol>0.9</lstol>
  <time_stepper>
    <dtmin>0.02</dtmin>
    <dtmax>0.1</dtmax>
    <max_retries>10</max_retries>
    <opt_iter>10</opt_iter>
  </time_stepper>
  <min_residual>0.001</min_residual>
  <qnmethod>0</qnmethod>
  <analysis type="static" />
  <symmetric_stiffness>0</symmetric_stiffness>
</Control>
```

5.2.2 Template loads and boundary conditions

This section describes the default loads and boundary conditions from which the simulations will be updated from.

The tibia will be fixed in all degrees of freedom using a rigid constraint:

```
<rigid_body mat="2">
  <fixed bc="x" />
  <fixed bc="y" />
  <fixed bc="z" />
  <fixed bc="Rx" />
  <fixed bc="Ry" />
  <fixed bc="Rz" />
</rigid_body>
```

There will be 11 load curves defined in this template model. Not all load curves will be used. They will correspond to:

1. Load-displacement characteristics of the ACL
2. Load-displacement characteristics of the PCL
3. Load-displacement characteristics of the MCL

4. Load-displacement characteristics of the LCL
5. Prescribed flexion rotation in the RCJ
6. Prescribed medial force in the RCJ
7. Prescribed internal rotation moment in the RCJ
8. Prescribed superior force in the RCJ
9. Prescribed valgus moment in the RCJ
10. Prescribed anterior force in the RCJ
11. Prescribed anterior translation in the RCJ

Be default, their corresponding load curves will be:

```
<loadcurve id="1" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
```

```
</loadcurve>
```

```
<loadcurve id="2" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
```

```
</loadcurve>
```

```
<loadcurve id="3" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
```

```
</loadcurve>
```

```
<loadcurve id="4" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
```

```
</loadcurve>
```

```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.2,0</point>
  <point>1.5,0</point>
  <point>3.5,0</point>
```

```
</loadcurve>
```

```
<loadcurve id="6" type="linear">
  <point>0,0</point>
  <point>4,0</point>
```

```
</loadcurve>
```

```
<loadcurve id="7" type="linear">
  <point>0,0</point>
  <point>1.5,0</point>
  <point>2.0,0</point>
  <point>2.5,0</point>
  <point>3.0,0</point>
  <point>3.5,0</point>
</loadcurve>
```

```
<loadcurve id="8" type="linear">
  <point>0,0</point>
  <point>0.1,0</point>
  <point>0.2,1</point>
  <point>4,1</point>
</loadcurve>
```

```
<loadcurve id="9" type="linear">
  <point>0,0</point>
  <point>1.5,0</point>
  <point>2.0,0</point>
  <point>2.5,0</point>
  <point>3.0,0</point>
  <point>3.5,0</point>
</loadcurve>
```

```
<loadcurve id="10" type="linear">
  <point>0,0</point>
  <point>1.5,0</point>
  <point>2.0,0</point>
  <point>2.5,0</point>
  <point>3.0,0</point>
  <point>3.5,0</point>
</loadcurve>
```

```
<loadcurve id="11" type="linear">
  <point>0,0</point>
  <point>4,0</point>
</loadcurve>
```

5.2.3 Passive flexion from 0 to 90 degrees

Flexion rotation will be prescribed from 0 to 90 degrees by adjusting load curve 5. Axial loads will not be applied. The parameters Flex0 and Flex90 will need to be calculated according to the ACS. In the MC phase, Flex0 was -0.28855 and Flex90 was 1.28225.

```
<loadcurve id="5" type="linear">  
  <point>0,0</point>  
  <point>0.2,0</point>  
  <point>0.5, Flex0</point>  
  <point>2, Flex90</point>  
</loadcurve>
```

5.2.4 Anterior-posterior translation (no ACL) at ~19.1 degrees knee flexion

This simulation will consist of setting the flexion angle while applying an AP load. Flex19 will need to be calculated according to the new ACS but will be ~0.044807887.

Flexion rotation will be set to 19.1 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">  
  <point>0,0</point>  
  <point>0.2,0</point>  
  <point>0.5, Flex19</point>  
  <point>2, Flex19</point>  
</loadcurve>
```

AP load will be applied by adjusting load curve 10.

```
<loadcurve id="10" type="linear">  
  <point>0,0</point>  
  <point>0.6,0</point>  
  <point>1.3,-80</point>  
  <point>2,80</point>  
</loadcurve>
```

5.2.5 Anterior-posterior translation (no ACL) at ~55.7 degrees knee flexion

This simulation will consist of setting the flexion angle while applying an AP load. Flex55 will need to be calculated according to the new ACS but will be ~0.683598393.

Time steps will be increased to 30 timesteps of 0.1

Flexion rotation will be set to 55.7 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">  
  <point>0,0</point>  
  <point>0.2,0</point>  
  <point>1.0, Flex55</point>  
  <point>3.0, Flex55</point>  
</loadcurve>
```

AP load will be applied by adjusting load curve 10.

```
<loadcurve id="10" type="linear">  
  <point>0,0</point>  
  <point>1.0,0</point>
```

```
    <point>2.0,-80</point>
    <point>3.0,80</point>
</loadcurve>
```

5.2.4 Internal-external rotation (no ACL) at ~17.7 degrees knee flexion

This simulation will consist of setting the flexion angle while applying an IE moment. Flex17 will need to be calculated according to the new ACS but will be ~0.020373278.

Flexion rotation will be set to 17.7 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.2,0</point>
  <point>0.5, Flex17</point>
  <point>2, Flex17</point>
</loadcurve>
```

IE moment will be applied by adjusting load curve 7.

```
<loadcurve id="7" type="linear">
  <point>0,0</point>
  <point>0.6,0</point>
  <point>1.3,-6000</point>
  <point>2,6000</point>

</loadcurve>
```

5.2.5 Internal-external rotation (no ACL) at ~56.0 degrees knee flexion

This simulation will consist of setting the flexion angle while applying an IE moment. Flex56 will need to be calculated according to the new ACS but will be ~0.688834381.

Time steps will be increased to 30 timesteps of 0.1

Flexion rotation will be set to 55.7 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.2,0</point>
  <point>1.0, Flex56</point>
  <point>3.0, Flex56</point>
</loadcurve>
```

IE moment will be applied by adjusting load curve 7.

```
<loadcurve id="7" type="linear">

  <point>0,0</point>
  <point>1.0,0</point>
  <point>2.0,-6000</point>
  <point>3.0,6000</point>
</loadcurve>
```

5.2.4 Varus-valgus laxity (no ACL) at ~15.4 degrees knee flexion

This simulation will consist of setting the flexion angle while applying a VV moment. Flex15 will need to be calculated according to the new ACS but will be ~ -0.019769295 .

Flexion rotation will be set to 15.4 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.2,0</point>
  <point>0.5, Flex15</point>
  <point>2, Flex15</point>
</loadcurve>
```

VV moment will be applied by adjusting load curve 9.

```
<loadcurve id="9" type="linear">
  <point>0,0</point>
  <point>0.6,0</point>
  <point>1.3,-10000</point>
  <point>2,10000</point>
</loadcurve>
```

5.2.5 Varus-valgus laxity (no ACL) at ~58.6 degrees knee flexion

This simulation will consist of setting the flexion angle while applying a VV moment. Flex58 will need to be calculated according to the new ACS but will be ~ 0.734212942 .

Time steps will be increased to 30 timesteps of 0.1

Flexion rotation will be set to 55.7 degrees by adjusting load curve 5.

```
<loadcurve id="5" type="linear">
  <point>0,0</point>
  <point>0.2,0</point>
  <point>1.0, Flex58</point>
  <point>3.0, Flex58</point>
</loadcurve>
```

VV moment will be applied by adjusting load curve 9.

```
<loadcurve id="9" type="linear">
  <point>0,0</point>
  <point>1.0,0</point>
  <point>2.0,-10000</point>
  <point>3.0,10000</point>
</loadcurve>
```


5.3 Steps to perform benchmark simulations

Once the calibrated prestrain values and new load curves have been written to the simulation case *.feb files, the benchmarking simulations will be run on a computer or HPC. The loads, moments will be output to the *.log file for plotting and analysis.

```
<logfile>
  <node_data data="x;y;z">9018, 9087, 15026, 15417, 105234, 21871, 3840, 39</node_data>
  <rigid_body_data data="x;y;z" delim=",">1,2,3</rigid_body_data>
  <rigid_body_data data="Fx;Fy;Fz;Mx;My;Mz" delim=",">1,2,3</rigid_body_data>
  <rigid_connector_data data="RCFx;RCFy;RCFz;RCMx;RCMy;RCMz"
delim=",">1,2,3</rigid_connector_data>
</logfile>
```

5.4 Burden

The software and hardware requirements, anticipated man hours and expertise level and computational cost per process are described below.

1. Select robot data

Software requirements: Microsoft office Excel

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM.

Anticipated man hours: ~ 10 minutes

Expertise level needed: Low

Computational cost: Low

2. Generate lookup tables

Software requirements: Python 3.6 & MATLAB R2017a Academic Licence

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM.

Anticipated man hours: ~ 1 day

Expertise level needed: Medium

Computational cost: Low

3. ACS alignment

Software requirements: Python 3.6 & MATLAB R2017a Academic Licence

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM.

Anticipated man hours: ~ 1 week

Expertise level needed: Medium

Computational cost: Low

4. Update template model

Software requirements: Python 3.6 & FEBio version 3.0.0 (with prestrain plug-in)

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM.

Anticipated man hours: ~ 1 week

Expertise level needed: Medium

Computational cost: Low

5. Recalibration

Software requirements: Python 3.6 & FEBio version 3.0.0 (with restart function & prestrain plug-in)

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM & High performance computer (Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz 1024GB Ram)

Anticipated man hours: ~ 1 month

Expertise level needed: Medium/High

Computational cost: High

6. Benchmarking simulations

Software requirements: Python 3.6 & FEBio version 3.0.0 (with prestrain plug-in)

Hardware requirements: Intel(R) Core(™) i7-7700HQ CPU @ 2.8GHz, 32.0 GB RAM & High performance computer (Intel(R) Xeon(R) Gold 6136 CPU @ 3.00GHz 1024GB Ram)

Anticipated man hours: ~ 1 week

Expertise level needed: Medium

Computational cost: High

6. References

ABI Model calibration documents and M&S outputs

Erdemir, A., & Sibole, S. (2010). Open knee: a three-dimensional finite element representation of the knee joint. User's guide, version, 1(0).

Good, E. S., & Suntay, W. J. (1983). A joint coordinate system for the clinical description of three-dimensional motions: application to the knee. *Journal of biomechanical engineering*, 105(2), 136-144.

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.

Maas, S.A., Erdemir, A., Halloran, J.P., Weiss, J.A. (2016). A general framework for application of prestrain to computational models of biological materials. *Journal of the Mechanical Behavior of Biomedical Materials*, 61, 499-510.

Orozco, G. A., Tanska, P., Mononen, M. E., Halonen, K. S., & Korhonen, R. K. (2018). The effect of constitutive representations and structural constituents of ligaments on knee joint mechanics. *Scientific reports*, 8(1), 1-15.