Large Scale In-Silico Clinical Trials: Developing a Workflow to Generate Boundary Conditions for Simulating a Functional Spinal Unit.

MECH3890 Individual Engineering Project
Large Scale In-Silico Clinical Trials: Developing a Workflow to Generate Boundary Conditions for Simulating a Functional Spinal Unit.

Author: Joshua Montgomery (201205471)

Supervisor: Zeike Taylor

Examiner: Qingen Meng

Date: 28/04/21

SCHOOL OF MECHANICAL ENGINEERING

UNIVERSITY OF LEEDS

MECH3890 – Individual Engineering Project

Large Scale In-Silico Clinical Trials: Developing a Workflow to Generate Boundary Conditions for Simulating a Functional Spinal Unit.

PRESENTED BY Joshua Montgomery

If the project is industrially linked, tick this box and provide details below.

COMPANY NAME AND ADDRESS:

STUDENT DECLARATION (from the "LU Declaration of Academic Integrity")

I am aware that the University defines plagiarism as presenting someone else's work, in whole or in part, as your own. Work means any intellectual output, and typically includes text, data, images, sound or performance. I promise that in the attached submission I have not presented anyone else's work, in whole or in part, as my own and I have not colluded with others in the preparation of this work. Where I have taken advantage of the work of others, I have given full acknowledgement. I have not resubmitted my own work or part thereof without specific written permission to do so from the University staff concerned when any of this work has been or is being submitted for marks or credits even if in a different module or for a different qualification or completed prior to entry to the University. I have read and understood the University's published rules on plagiarism and also any more detailed rules specified at School or module level. I know that if I commit plagiarism, I can be expelled from the University and that it is my responsibility to be aware of the University's regulations on plagiarism and their importance. I re-confirm my consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to monitor breaches of regulations, to verify whether my work contains plagiarised material, and for quality assurance purposes. I confirm that I have declared all mitigating circumstances that may be relevant to the assessment of this piece of work and that I wish to have taken into account. I am aware of the University's policy on mitigation and the School's procedures for the submission of statements and evidence of mitigation. I am aware of the penalties imposed for the late submission of coursework.

Signed J Montgomery Date: 28/04/21

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

Clinical trials often cost hundreds of millions of dollars, with no guarantee the product will ever make it to the market. In an attempt to reduce the costs associated with these trials, computer simulation has been increasingly used to supplement or replace traditional in-vivo (performed on a living being) trials. So far, in-silico (performed on computer) trials have only replaced sections of the preclinical trial assessment and supplemented the actual trials. Before this changes, research must be performed to prove that the results of these digital simulations are at least as accurate as in-vivo studies and research must be performed to prove that these in-silico clinical trials can be scaled up into large-scale In-Silico Clinical Trials (ISCTs). To the author’s knowledge, there is currently no research on how to scale up the process, hence the goal of this project is to develop a workflow to automate most of the ISCT process and hence prove that the large-scale ISCTs are viable. This will be accomplished by developing a workflow that will format models for processing; apply realistic material properties and generate population-variant boundary conditions for a variety of loading environments.

1 INTRODUCTION

It was calculated by the US Department for Health and Human Services that the average cost of development for a complex medical device is $526.4 million (Sertkaya, et al., 2019), including the cost of failures and lost capital opportunity. This cost can often prove to be prohibitively expensive and stifle innovation (Avicenna Alliance, 2016), as if a product gets into a late stage of development and then a failure mode is discovered that has a simple preventative correction, the clinical trial will have to regress to an earlier stage and potentially even start again, to ensure that the modified device is still safe. Unfortunately, this means that medical devices that would pass a clinical trial with a minor modification are often discarded and their cost incorporated into other products.

In other industries, such as automotive or aerospace engineering, it is standard practice to develop a full computerised model of a product, subject it to a wide array of boundary conditions to determine potential failure modes and then subsequently improve the design (Viceconti, et al., 2016). This design philosophy when applied to the Biomedical industry, is referred to as an In-Silico Clinical Trial (ISCT), because some of the clinical trial is done entirely on a computer. By performing an ISCT, the medical device can be tested on a digital sample of patients far larger and more representative of an average population, at a literal fraction of the cost. An additional benefit of ISCTs is that you can test the medical device with outlier patient models and hence determine failure modes that might only occur in extreme boundary conditions. Before ISCTs are to see widespread adoption, parts of the process will need to become automated, so that the populations (and hence boundary conditions) of any size can be generated and then applied sequentially to the model. Hence removing the need for a person to manually run every simulation, making a truly large scale ISCT viable.

This project will demonstrate the advantages of ISCTs by developing a workflow for executing a large scale ISCT on an FSU (Functional Spinal Unit), using boundary conditions that are procedurally generated and are representative of a population. The finished workflow can then be executed on the command line and the specified number of simulations will be created and sequentially executed without any further user input. Hence the laborious process of manually generating loading values, altering the models and then executing them has been eliminated; ensuring that the maximum number of ISCTs is limited only by available computing resources and not by how many hours of expensive specialist labour can be afforded. As the generated loading conditions will be reflective of the population, outliers will also be modelled so that the models are validated for extreme loading environments. To the extent of the author's knowledge, no other research has been undertaken to develop a workflow that will automate most of the In-Silico Clinical Trial process. Hence, this project will serve to demonstrate the utility of automation in ISCTs and how it facilitates scaling up ISCTs to their full potential.

This report will focus on the research and calculations used in the workflow, instead of the operation of this workflow. Should the reader desire to fully understand the operation of the workflow, developed during this project, the technical documentation has been included in the appendix of the report. However, reading this documentation is not necessary for comprehension of this report.

1.1 AIMS

Develop a workflow that sequentially executes simulations on the L4/L5 (the 4th and 5th lumbar vertebrae) FSU models, using generated boundary conditions that are representative of a population. These boundary conditions will vary due to differences within the modelled population and hence will allow for outlying data to be studied alongside more typical data.

1.2 OBJECTIVES

- Develop a workflow in Python that can be executed on the command line of most operating systems.
- Optimise the workflow execution speed to maximise the time savings from automation of the ISCT process.
- Facilitate usage by writing 'Technical Documentation' that explains the usage of the workflow and explains how to make any changes to expand function.
- Vary the loading values produced by the boundary condition generator using anatomical data representative of a population.
- Further ensure ease of use by extensively commenting the code to allow for convenient operation of the code.
- Validate the boundary condition generator.

1.3 PROJECT REPORT LAYOUT

- As seen above, the introduction section will present the project, the aims, objectives and the reason for this research.
- The second section of this project report will discuss and introduce the different areas of research required for this project. These areas are In-Silico Clinical Trials, spinal modelling, spinal loading calculations, population distribution and material refinement. In each of these sections the relevant literature is reviewed and a summary is provided outlining any gaps or problems in the existing research.
- Section 3 will present the boundary condition generator and the calculations for each of the different loading environments.
- The fourth section will discuss how the material properties of the MySpine models were improved and a validation will be performed on the new material properties.
- Section 5 will be the concluding section and contain the achievements, an overall discussion and the concluding remarks.
- References will be presented in section 6.
- The appendix will be the final section, section 7. This appendix contains the technical documentation designed to support the use of the workflow. Reading the technical documentation is not necessary for the comprehension of this report, however it will outline the entire operation of the workflow, developed during this project.

2 LITERATURE REVIEW

The completion of this project has required extensive research into several, distinct areas that have little overlap. Hence this literature review will be presented with a separate section for each area of research.

2.1 IN-SILICO CLINICAL TRIALS

In-Silico Clinical Trials (ISCTs) are still a long way from being a replacement for a conventional clinical trial, however, they are seeing increasing use as a supplement to these trials, especially for the development of medical devices. One of the earliest examples of using ISCTs for medical device approval is for an embedded medical device that would monitor and control the insulin levels in a diabetic patient, developed at University of Padova and the University of Virginia, starting in 2006 (Dalla Man, et al., 2007). Typically, before such a device can go to clinical trials, a pre-clinical study on animals needs to take place. FDA guidance at the time suggested that they should test the device on pigs, but the researchers raised several objections to doing so as pig biology is significantly different to human biology and would require the device to be completely recoded. They argued that this made such a test irrelevant, instead suggesting that an ISCT on the software of the device tested on digital human models would generate more useful data and show that the device can safely control the blood sugar levels of a human. The FDA agreed to the proposal in 2008 and the preclinical trial was performed entirely In-Silico (on a computer), hence increasing development speed and reducing cost, whilst maintaining safety standards. This pre-clinical assessment and then subsequent approval of the device after the clinical trial, influenced FDA policy, as documented extensively in Marco, et al., (2017) and led to the FDA publishing guidelines (FDA, 2016) (last updated 2016) on performing computer simulations and ISCTs for medical devices.

According to data collected by Morrison et al (2018), 44% of the medical device applications in 2017 that they examined, contained computer simulations in the submission. This statistic shows the rapid adoption of computer simulation for clinical trials for medical devices, however there are still barriers that need to be overcome before computer simulations become true ISCTs. Computer simulations are useful and potentially better than in-vivo tests (testing on a real, living being), however, they require a highly educated and specialised workforce to run every simulation manually. As a result of this the throughput is similar to that of a conventional clinical trial, and hence technologies need to be developed that will automate the running of simulations (Viceconti, et al., 2016); generate unique boundary conditions that are representative of a population and output the data in an easy to interpret way.

At present, there has been little to no research about how to increase the scale of an ISCT and the coding required for automation. There are several potential explanations for this: firstly and most probably, ISCTs are a recent technological development and serious funding only started being available after the EU commissioned the Avicenna Roadmap (Avicenna Alliance, 2016). The term ISCTs was only invented in 2011 by the VPH institute (2011), hence the field of ISCTs is far from reaching maturity, and it would be a fair comment to suggest that the research too was in its infancy and was targeted more towards the development of individual models to assist a conventional clinical trial, rather than as an alternative. An additional reason for the lack of research in this area is that medical device approval bodies, so far, only accept ISCTs as a supplement to a clinical trial or a replacement for a pre-trial in-vivo testing on animals. Before this will change, research needs to be performed to prove the benefits of ISCTs, namely that they allow for a much larger and diverse population to be tested. Thus, providing an ethical method to perform clinical trials on patients with attributes that would make them an outlier or have comorbidities or exacerbating factors that may increase the risk of a trial to unsafe levels. To prove these benefits, research must be done to demonstrate that ISCTs can be at least partially automated and subsequently run for a large (n>1000) number of simulations. Hence this project aims to prove the large-scale validity of ISCTs, by developing a script to partially automate an ISCT.

2.2 Spinal Modelling

Before the approach of this project can be detailed, an explanation of the anatomy of the spine must be made to demonstrate the difficulties of forming a model of the spine and furnish the reader with an understanding of the areas in discussion. To this affect, a diagram of the human spine has been included as Figure 1, with the areas discussed in this report highlighted.

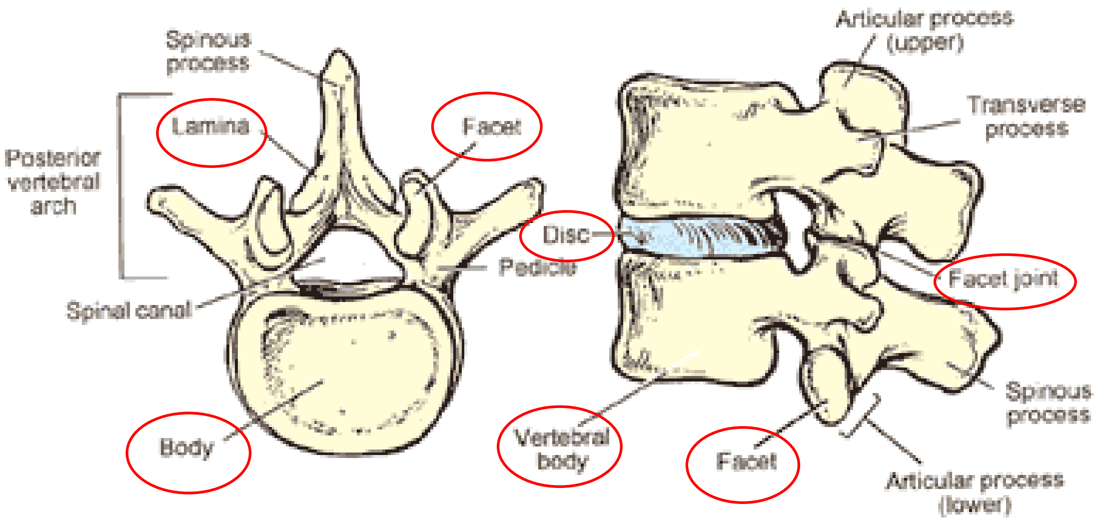


Figure 1- Diagram of a Functional Spinal Unit (Spoonamore, 2018)

An average adult human spine consists of 22 separate vertebrae and around 11 that are fused to other vertebrae (Cedars Sinai Clinic, 2021). Each one of these vertebrae has several contacts with the other vertebrae on the facets of the lamina and are separated by an intervertebral disc which has anisotropic hyperelastic properties (Mosbah & Bendoukha, 2018). To further complicate matters, the nucleus of the intervertebral disc is an incompressible fluid and every one of the muscles in the body affects the spine. This means that creating models of the spine is very difficult and typically a variety of assumptions and simplifications are used.

One of the most accurate and generalised models of the human spine to date, is the Lifting Full-Body (LFB) model (Beaucage-Gauvreau, et al., 2019), consisting of 30 separate sections and 238 muscles. Although this model was created by a team of 8 experts in the field of spinal biomechanical modelling, they still had to make numerous assumptions such as reducing the number of muscles and degrees of freedom. In spite of these assumptions, the model generated is still impractical for use outside of research, due to the prohibitively large computational resources required to execute even a simple simulation. When this approach was being evaluated a simplified, open-source version of this model, consisting of just a just a spine and legs, was acquired and a simple simulation was executed. However, the simulation took 6 hours to solve only the first 0.1 seconds of loading, without any applied boundary conditions. Furthermore, modification of the model required skill in using an open-source piece of software called 'OpenSim' (Seth, et al., 2018), This means that using the LFB model would be impractical for this project, as it is mainly focused on developing a workflow, hence a simpler set of models are required along with a simpler way to calculate the boundary conditions of the FSU.

Before the continued exponential growth of computer processing power facilitated the development of full body models, scientists and health professionals used models of specific areas of risk and separately determine an appropriate loading value. This approach is still in wide usage in industry due the hugely reduced time and skill requirements for only a small reduction in accuracy. As discussed by Eklund (1988), multiple methods are often employed to determine spinal loading, depending on the degree of accuracy required. An expert in their laboratory may use a full-body digital model, but a health and safety inspector in a factory will not have the time or resources to do this and will typically employ the use of a polynomial instead. It is these polynomial equations that will be key for this project, as discussed further in section 2.3.

The work of Han, et al., (2012) shows that across a broad spectrum of height and weight, the L5 vertebrae always experiences the greatest stress when the person is standing. This is as expected, because the L5 vertebrae is the last of the non-fused vertebrae at the bottom the spine and effectively supports all the mass above that. Since the L5 vertebrae experiences the largest stress, it is a logical candidate for modelling. However, by modelling only a single vertebra, the intervertebral disc is neglected, and the model would not be reflective of a real spine. Instead, a model of a Functional Spinal Unit (FSU) is used, as it “is the smallest segment of the spine that exhibits biomechanical properties similar to those of the entire spine” (Panjabi & White, 1980). This FSU consists of two vertebrae and the intervertebral disc in between, as displayed in Figure 1. For this project, the L4/L5 FSU will be studied as it experiences the greatest loading.

The specific model set of L4/L5 FSUs used are those generated as part of the MySpine Project (VPH Institute, 2013) that generated anonymised patient specific models from imaging data. These models are ideal for a variety of reasons, such as being already anonymised as part of the project and hence require no additional consent from the patients to use. Another key advantage is that they are finite elements models that can be imported into Abaqus and as Abaqus is built on a Python kernel, they can be more easily manipulated using scripting. Most importantly, there are 168 of these models, hence a large scale ISCT is possible with this model set. These MySpine models were also enhanced to model the intervertebral disc by Amin Kassab-Bachi, based on the work of (Holzapfel, et al., 2000).

2.3 SPINAL LOADING CALCULATIONS

In this project, the focus will be on generating boundary conditions for a single Functional Spinal Unit (FSU), which consists of two vertebrae and the intervertebral disk between. The coding model developed in this project will be transferable to ISCTs on other body parts but is limited to just the FSU in this project. However, a user of the workflow would be able to change models relatively easily.

To generate boundary conditions for the L4/L5 vertebrae, it must first be determined what the loading environments are, as what the person is doing (i.e. sitting, lying or climbing stairs) will change the load substantially. Because any medical device would have to be functional across all extremes of loading conditions, this project must also generate boundary conditions for a variety of loading conditions. Rohlmann et al (2014) measured the compressive loading conditions of the spine for over 1000 activities and combinations of parameters in post-operative spinal surgery patients to measure which everyday activities led to the greatest loading. This study is useful in directing this project towards which activities should be considered, however their study was performed on 5 post-operative patients, the youngest of whom was 62, which suggests that the numerical data generated may not be as relevant for a native spine (not damaged or surgically altered) or a more distributed population sample. Hence, the numeric data for average loading will be ignored, but the data that in all 5 patients lifting a weight from the ground; lifting their arm with a weight in hand; standing up, sitting down and climbing the stairs were ranked in the top 10 of loading conditions, would suggest that these activities warrant further investigation and incorporation into the boundary condition model. However, these compound manoeuvrers are affected greatly by the gait and approach of the person performing the activity. This would make it impossible to model the orders of magnitude more complex compound movements suggested by Rohlmann et al (2014), as gait matters more than all other physical parameters and there is no research about how to approximate gaits from physical characteristics. Thus, it may be more logical to attempt to model loading environments that are simple, approached universally or are tightly controlled. Regardless of which loading environments are chosen, these loading conditions will need extensive research before their boundary conditions can be automatically generated and hence, in the early stages of coding development, the work of Bruno et al (2017) will be used to generate approximate compressive loads.

Before the boundary conditions can be generated, there must be a method to calculate the compressive loading through the spine during various activities and hence determine what a reasonable distribution would be across a population. However, calculating compressive loads in the spine can be difficult, as movement of the limbs leads to induced moments on the spine and the activation of stabilising muscles in the core further increases the compressive load. So, when picking an item off the floor, the increase in compressive force in the spine is typically greater than the weight of the object. Mathematically explaining these relationships is outside the scope of this project, so they will be gathered from various academic sources.

Calder & Potvin (2012) in their report, generate a polynomial equation to calculate the compressive load in the L4/L5 vertebrae as a function of bending moments in the arm and the height the load has been raised to. This equation is useful as it incorporates the potential energy of the arms/ hands and as a result is significantly more accurate than equations of a similar complexity. The population distribution can then be implemented by simply varying the moments with respect to different members of a population, as discussed in the following section. Calculating boundary conditions for loading that does not involve the arms or major activation of the stabilising core muscles is thankfully significantly easier and can be approached as a more conventional problem, for example, standing and lying down are just statics problems, significantly varying only with a person's weight (Han, et al., 2012). The data from the previously cited article was used to generate a linear equation to determine the spinal loading when standing from a person's weight. As the article showed that height had very little affect compared to weight, height was not a factor in the equation. This assumption is or critical importance as otherwise distributions would need to be found that link height and weight, which would have likely required several different equations.

An additional, simpler polynomial equation (McGill, et al., 1996) was used that does not account for potential energy because it would be more applicable to loading environments were the additional complexity of the potential considering polynomial would lead to overfitting.

2.4 Population Distributions

Calculating spinal loading values that are reflective of a population requires data that accurately represents that population. In this project the spinal loading is largely calculated using polynomial equations that convert moments into approximate spinal loads, but the moments themselves need to be determined. Hence to integrate the population variation the moments must vary with respect to different body sizes and weights. As examples, the spinal loading when standing varies with weight whereas the spinal loading from holding a weight at arm's length largely varies exclusively with arm length. This section of the literature review analyses various sources and distributions to gather the substantial amount of data on human anatomy variation required for this project.

The ideal approach to determine the distributions of these bodily variations would be primary data collection, so that all the data would be cohesive for the population sampled. However, that would be far outside the scope of this project, so the next best alternative would be to use data gathered as part of another project. This approach is unfortunately also a dead end, as the desired data is never published as it is only the means to test the hypothesis of the project. Furthermore, this data is classed as sensitive, patient-specific and hence would be protected by the General Data Protection act (European Union, 2016). The issue is further compounded as the data needs to be cohesive for an individual, as a taller person will logically weigh more on average, hence the anatomical data is not independent. This means that it would be inappropriate to randomly assign all anatomical values, as it would suggest a taller person is equally likely as a short person to have short arms. To resolve this, a single starting variable, either height or weight, is randomly generated for each person and then all subsequent anatomical parameters are calculated from that individually generated parameter, as will be discussed in the following paragraphs.

As outlined in the scoping document, the original intention was to use data representative of British adults, however the Office for National Statistics has stopped publishing average British anatomical data and has never provided distributions. However, the Centre for Disease Control and prevention (CDC) in the United States publishes means and standard deviations for height and weight regularly. For this project, distributions published in the "Vital Health and Statistics" (CDC, 2016) series are used, specifically, the cumulative data for American adults (21 years or older) of both genders. Generating boundary conditions reflective of a different country's population is not ideal, but is ultimately the only way for this project to continue. There was also some consideration if the American data should be used to approximate British values, however this would only serve to increase the error of the data as the two populations are not easily comparable, so the United States population is modelled instead.

However, the CDC does not provide data on the relationship between height and other anatomical features, such as arm length, so instead polynomial equations are used to convert height into corresponding and reasonable values for arm length. The first of the polynomial equations used was designed to convert demi spans (the length from knuckle to knuckle with arms in a T-Pose) into a height for patients who could not stand up to have their height measured (Basey, 1986). This equation is then rearranged to predict demi spans from height for use in this project. The original equation was fitted on bed-bound English patients in 1984, so it will not perfectly represent the average American adult, due to the skew towards elderly patients and nearly 40 years of demographic trends. To determine arm length from demi span shoulder width is also required. Two different papers were considered to provide the necessary data, the first option was a Swedish study (Hanson, et al., 2009) provides data for average shoulder width and the second an Indian study (Shah, et al., 2015) that provides equations to estimate height from other dimensions, intended for use in criminal forensics. One of these equations determined height from shoulder width and could be rearranged to calculate an approximate shoulder width from a provided height. Because the variation in shoulder width is almost negligible, initially the population variation was neglected in favour of the Swedish study, however after a comprehensive data analysis was performed it was determined that there were some ethnic groups in India with similar shoulder width distributions. Specifically, Indian Hindu men and Muslim women are closest in shoulder width to Americans. Hence the Swedish study was replaced with the Indian study for the increase in accuracy that varying shoulder width would bring. According to data from the Indian National Institute of Nutrition (2020) the average height of an Indian male is 165cm, smaller than the 175cm measured for men in the USA. This means that a shoulder width calculated using the equations for Indian people will likely be an extrapolation and hence may be inaccurate, however there is no alternative way to determine shoulder width from a relationship with height.

There was no data available for average positioning of the L4/L5 vertebrae in an adult, so instead measurements of the position of the author of this report’s L4/L5 vertebrae were taken. As the vertebrae were 55% of the author’s height from the floor, it was assumed this was the case for everyone else. This is quite a large assumption, since the sample size is only 1, however the current pandemic severely limits the potential of greater primary data collection. Furthermore, even a small-scale collection of primary data just on the average height of the FSU would require ethical approval and hence would be outside the scope of this project. The anatomical data used in this project will be a large source of error, but for a project of this size, primary data collection is not possible. There may be a wide body of data for average human anatomical data and ergonomic tolerances, but these data sources do not present a distribution and hence are irrelevant to this project. This lack of readily available population data represents a large gap in the research and a potential future target for research.

2.5 Material Refinement

The 168 models provided from the MySpine project have uniform properties assigned as default, roughly analogous to averaging the material properties over the spine and applying that average uniformly. This means that the computational resources required for running the simulation are limited, but the accuracy of the model is greatly impaired, as discussed in section 4.7. To reduce this inaccuracy, new material properties were determined from literature and then applied to the models procedurally.

The first step in this process was to identify the regions of the spine where material properties vary extensively and hence determine logical places to vary the MySpine models. This was achieved by both analysing anatomical texts for spine properties and the computational spinal modelling efforts of other projects. Of all the anatomical texts read for this project, the most useful proved to be the work of Oxland (2015) as it provides a summary of the spinal modelling advances over 25 years and the changing approach to material property modelling. Oxland's work cites several papers that prove that material properties across one vertebrae do not vary and across adjacent vertebrae the differences are negligible (Silva, et al., 1994) (Fazzalari, et al., 2006), hence the two vertebrae in the L4/L5 FSU can be fairly approximated as having identical material properties. The material properties across spinal regions (lumbar, cervical and thoracic) do vary extensively though. The geometries of the vertebrae change substantially even between adjacent vertebrae, but the material properties change negligibly between adjacent vertebrae in the lumbar region. Furthermore, the facets of the spine have uniform simple elastic material properties across the FSU according to (Natarajan & Andersson (1999).

The annulus fibrosis (the outside shell of the intervertebral disc) exhibits anisotropic hyperelastic properties (Mosbah & Bendoukha, 2018) ,whereas the nucleus (the inside core of the intervertebral disc) is an incompressible fluid. These particular material properties will be discussed extensively in this report, hence an explanation of what anisotropic and hyperelastic mean will be provided here. An isotropic material is one in which the material properties do not vary in different dimensions, whereas the inverse is true for an anisotropic material. For example, the vertebrae are stiffer in the vertical axis aligned along the spine, as this is logically the direction wherein the most load will be applied. A material that experiences hyperelasticity does not have a linear stress-strain relationship and instead the stress-strain relationship has to be determined by analysis of a strain energy density function (Bonet & Wood, 2008). Whilst these properties may be the most accurate representation of the intervertebral disc, they are exceedingly difficult to implement programmatically for multiple models without triggering convergence issues. The resolution to these problems will be discussed in section 4.

According to the work of Tan, et al., (2004), there is substantial variation in the material properties of the spine across different ethnicities, due to differences in lifestyle and diet. To truly represent a population, this would need to be considered, however as discussed in the previous section, the data simply is not available. The only way to realistically account for ethnicity in the boundary condition generator would be primary data collection on anatomical variation across ethnic lines, such as height, weight and spinal characteristics. This would be an excellent target for further future research but is outside the scope of this project. There is also a substantial body of research on how material properties of the spine vary with age (Grote, et al., 1995), but again the data simply is not available for these varying material properties to be applied in this project.

2.6 SUMMARY

Section 1 clearly shows the increasing trend in medical device approval applications featuring greater usage of computer modelling and hence, ISCTs. However, most of these applications are consigned to pre-clinical animal trial replacements due to the current limitations of ISCTs. As the field matures, ISCTs will likely be used for more of the clinical trial process, but first, the modelling technology must be proved as reliable as convential trial and research must be performed to demonstrate the utility of ISCTs.

As presented in sections 2.2 and 2.3, there is an abundance of research on the modelling of spines and calculating how they experience loading. This is largely because both of these fields are useful outside of ISCTs, such as spinal loading calculations for occupational health and spinal modelling has long seen extensive usage in medicine. The research already undertaken can be applied to ISCTs, hence meaning that the accurate modelling of the spine is one of the most well researched aspects of ISCTs.

The reoccurring theme of this literature review is that although there may be enough data for spinal modelling of an average adult, there is almost no data pertaining to how spinal properties or anatomy varies across the population. This is obviously problematic as this project is based on developing a workflow for performing a large-scale In-Silico Clinical Trial representative of the population, hence a significant number of assumptions are required to simplify the problem and allow for approximate relationships to be determined. If the scope of this project were to be expanded towards increasing accuracy of the results and population modelling, the only logical place to start would be collecting better anatomical data. However, the primary objective of this project is developing a workflow as a proof of concept, so later research could focus on increasing the accuracy through primary data collection or purchasing an existing data set.

Despite these issues, there is a clear gap in the existing research which this project will help to fill, as almost all existing research is dedicated towards manual operation and small-scale application. This project serves to prove that automation of wide swathes of the process In-Silico is not only possible but also can generate large bodies of data relevant to an actual population.

3 BOUNDARY CONDITION GENERATION

The current practice for In-Silico Clinical Trials (ISCTs) is for the loading environment to be manually calculated and specified; however, this creates a huge bottleneck and drastically reduces the potential scale of any ISCTs. Hence a key component of this project is to develop a workflow that programmatically calculates the boundary conditions, applies these conditions and then executes the simulations. To further demonstrate the advantages of a large-scale ISCT, the boundary conditions will be generated using the population distributions presented in section 3.1, hence meaning that these boundary conditions are representative of the population.

As discussed in the literature review, three different methods are applied to calculate the loading conditions, as detailed in table 1. These methods are introduced here and assigned a short-form name to make method designation easier in the subsequent sections. A validation of these methods will be provided in section 3.3.

Table 1- The three methods employed to calculate boundary conditions for different loading environments.

Name	Reference	Explanation
No Moment Assumption	(Han, et al., 2012)	It is assumed that there are no induced moments from the loading environment, hence all loads act down the spine and a linear interpolation can be applied to determine the spinal loading.
Potential Considering Polynomial	(Calder & Potvin, 2012)	A polynomial equation is used to convert the applied moments into spinal boundary conditions. This polynomial considers the potential from off-axis load support and is correspondingly more accurate for kinematic/more complex loading.
Higher Order Polynomial	(McGill, et al., 1996)	A polynomial equation is used to convert the applied moments into spinal boundary conditions. This polynomial contains less unique variables than the previous polynomial but is of a higher order and hence is more accurate for loading environments which can be modelled with just moments.

3.1 Population Distributions

To ensure that all the anatomical data is coherent and logical, some of the data should correlate. For example, as height increases so too does the average arm length, hence for the generated anatomical data to be accurate the values cannot be independent of each other. The ideal way to approach this dependency would be to determine distributions within the threshold of a primary variable (such as average arm length and standard deviation for men between 160-165cm) and have many different distributions to cover the entire range of data, however there is not enough data available for such an analysis. Another option would be to just generate the anatomical data fully independently, however this approach is flawed as would suggest improbable data, like a shorter person being equally likely as a tall person to have long arms. For this project, it was decided that assuming full anatomical independence was too inaccurate due to the massively increased output of outlying anatomical data that would result.

Instead, various correlation equations were used to generate data reflective of the input primary variable, an overview of these equations are presented in table 2. As an example, for the first equation in table 2, an input height and gender are converted into the average demispan (distance between knuckles when a person's arms are fully extended in a T-pose) for a person of that height. This assumes that everyone has precisely the average demispan, which is obviously not accurate, but it is the only way to generate secondary anatomical data that varies logically with the randomly generated primary data (gender and height or weight). So, for this project gender and height or weight are generated randomly then the secondary anatomical data is generated from those values. For two of the loading environments, height and the associated secondary values are irrelevant as only weight is required.

Table 2- An overview of how secondary anatomical features are derived.

Inputs	Output	Reference	Used for
Height Gender	Demispan	(Basey, 1986)	Two-Handed Horizontal Hold and One-Handed Horizontal Hold
Height Gender	Shoulder Width	(Shah, et al., 2015)	Two-Handed Horizontal Hold, One-Handed Horizontal Hold and Pushing Open a Door.
Height	L5 Height	-	Pushing Open a Door.

3.2 CALCULATING SPINAL LOADING

Each loading environment requires different approaches to determine the boundary conditions of spinal loading at the L4/L5 Functional Spinal Unit (FSU). These programmatic approaches will be detailed in the sub sections listed below.

3.2.1 Standing

The standing spinal load can easily be calculated using the No Moment Assumption method, as described in table 1. This assumption is valid because there are no applied loads, so any present spinal loading is only due to the weight of the individual. According to the work of Han, et al., (2012), there is an almost perfectly linear relationship between weight and the L4/L5 FSU of standing individual. Taller people do tend to have greater spinal loading, but that is entirely explained by the average weight being higher and hence height is not relevant for this calculated. Finally, a weight generated for American adults using the CDC data (CDC, 2016) is interpolated within the Han, et al., data to generate the boundary condition.

3.2.2 Two-Handed Horizontal Hold

This boundary condition calculates the spinal loading when a mass is supported at shoulder height with both arms horizontal and directly in front of the body. For the purpose of this analysis, it does not matter if it is one mass supported with two hands or two identical masses each supported with one hand. Figure 2 shows an image of someone supporting a load in this way.

Figure 2- Two-handed horizontal loading. Image from (Skimble Workouts, 2021)

As this loading regime is static, the Higher Order Polynomial is used, as detailed in table 1. The polynomial is provided below as equation 1 where F_c is the compressive force on the FSU, M_f is the flexion moment around the FSU, M_L is the lateral moment around the FSU and M_A is the axial twist moment.

$$F_c = 1067.6 + 1.219M_f + 0.083M_f^2 - 0.0001M_F^3 + 3.229M_L + 0.119M_L^2 - 0.0001M_L^3 + 0.862M_A + 0.393M_A^2 - 0.0001M_A^3 \qquad (1)$$

Due to the inherent symmetry of this loading environment, there are no lateral or axial moments. The flexural moment is simply equal to half the demispan minus shoulder, width all multiplied by the mass being supported (by default 5kg).

3.2.3 One-Handed Horizontal Hold

Calculating the boundary conditions on the L4/L5 FSU for the one-handed horizontal holding regime is very similar to that of the previous loading condition, apart from the presence of a lateral moment, due the asymmetrical load distribution from using only one arm. This lateral moment is equal to the calculated shoulder width multiplied by the mass being suspended (by default 5kg). If the same load is being suspended with either one or two hands, the spinal compression is greater with only one hand due to the additional induced moments.

3.2.4 Two-Handed Hold by Sides

Because in this loading environment the persons arms are down by their sides, it can be reasonable assumed that there are no moments around the FSU. This is because the symmetrical loading eliminates the axial and lateral moments and as the masses are positioned in the same plane as the FSU, the distance from the FSU to the mass in the forward direction is zero, hence the flexural moment also equals to zero. As a result, the No Moment Assumption method can again be applied to simplify the calculation of the spinal compression. Again, a weight is generated using the distributions provided by the CDC, but for this loading environment the additional load applied is added to the generated weight. This combined value is then interpolated into the standing load dataset (Han, et al., 2012) to generate the boundary condition information. It is assumed that the adding the two loads together and then interpolating into the data is still valid as there are no moments and hence this loading environment is functionally identical to increasing the average weights generated.

3.2.5 Push Open Door

This loading environment was programmed flexibly using the Potential Considering Polynomial, so that it can be easily adapted to another dynamic movement. The exact equation is presented as equation 2, where F_c is the compressive force on the FSU, L_h is the load height above the L4/L5 FSU, L_s is the height of the subject, M_f is the flexion moment around the FSU, M_L is the lateral moment around the FSU and M_A is the axial twist moment.

$$F_c = 3326.4 + 3.81L_h - 16{,}68L_s + 11.81M_f + 0.68M_f^2 + 0.56M_l^2 - 0.06M_A^3 \qquad (2)$$

The height of the subject is generated by the distribution provided by the CDC and is also used to calculate the load height above the FSU, using the assumption that a person's L4/L5 FSU is situated 55% of their height from the ground. This assumption is based on the anatomy of the author as no other data was available.

Pushing open a door was chosen for this loading environment because it is a simple compound movement that would suitably demonstrate the capabilities of the Potential Considering Polynomial. It is also a simple loading environment to model because the force applied to the door will only exert an axial twist moment. However, because of the movement inherent in pushing open a door (not holding), the Potential Considering Polynomial is most accurate.

3.3 VALIDATION

All three of the methods employed in the boundary condition generator are from published and peer-reviewed papers that contain validations for their methods. This means that performing a validation on the equations would be redundant. Instead, the decision to apply a particular method for a loading environment will be validated by comparing how each method models several loading environments, proving the need for multiple methods.

Firstly, the two polynomial equations are evidently unsuitable for the usage of calculating the load from standing as they contain no terms relating to the weight of the person and peer reviewed research shows that this is the predominant factor in determining the compressive force from standing. This is because these polynomial equations were formed to accurately predict the spinal boundary conditions from the application of external loading and the subsequent induction of moments around the FSU. The inverse is also true, the method for calculating standing load does not consider moments and hence is unsuitable for modelling almost all loading environments that apply additional external loads (apart from those loading environments that do not induce moments such as holding two identical weights at by your sides).

Since it is clear that there is no potential for overlap between the two polynomial methods and the No Moment Assumption method, the only numerical validation required is to demonstrate the need for two different polynomial equations to model different loading environments. To do this, both polynomials will be used to calculate the boundary conditions for two loading environments, the aforementioned two-handed horizontal hold and pushing open a door. This will serve to demonstrate that the polynomials cannot be accurately used outside of the conditions for which they were designed and hence the necessity of two different equations. The data for these two loading environments is graphically presented as Figures 3 and 4. To simplify the validation, a list of heights from 150-180cm in 1cm increments served as the primary anatomical data and all secondary data was calculated using the distributions for men. The validation would have shown the same results if the distributions were set for women instead, but as the FSU height relationship was determined through sampling one man, it was more accurate to use men.

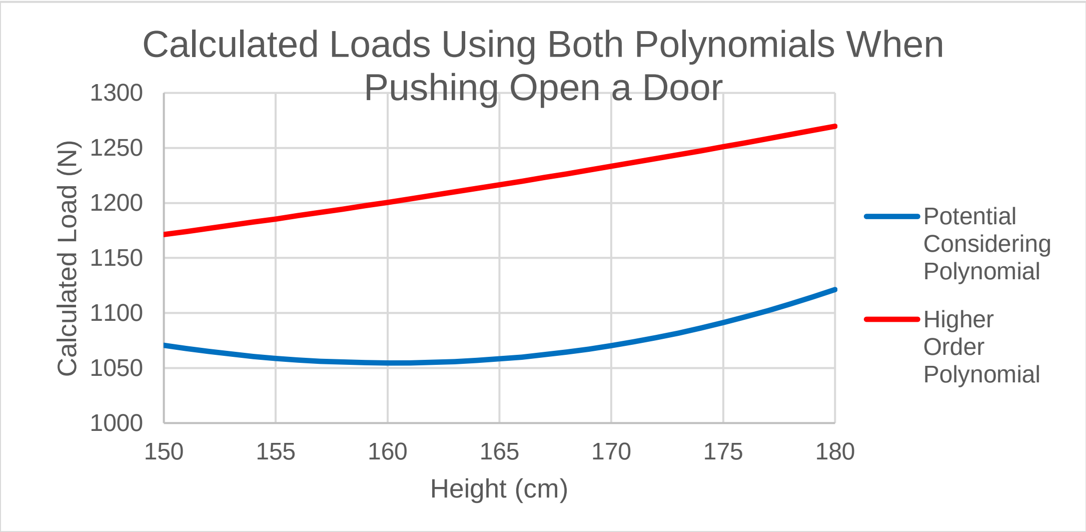


Figure 3- Graph showing how using the two different polynomial equations affects the calculated loading values when pushing open a door.

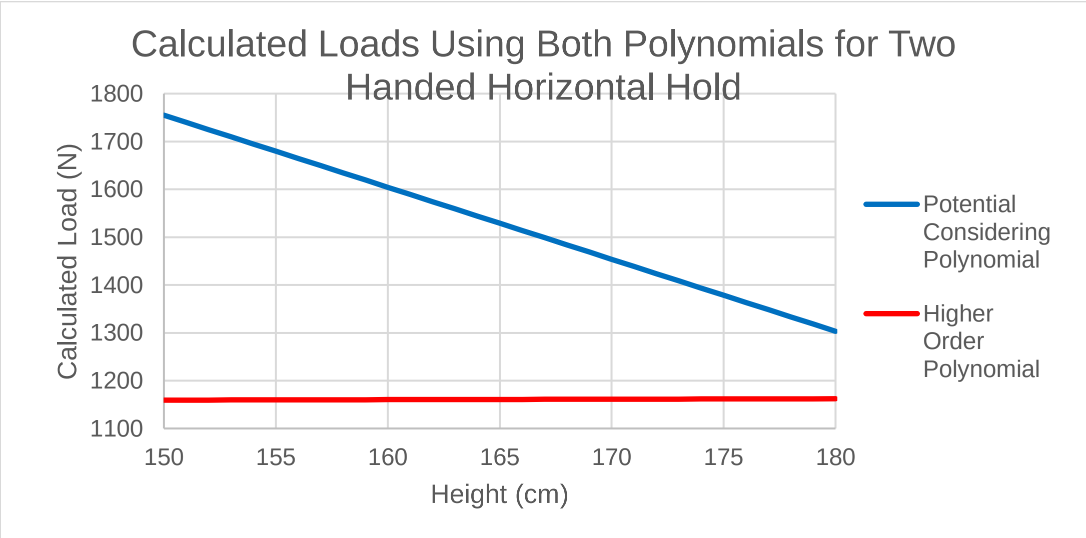


Figure 4- Graph showing how using the two different polynomial equations affects the calculated loading values when holding a weight at shoulder height with both hands.

From Figure 3, it appears that the load values calculated using the Higher Order Polynomial have a linear relationship with height; this is because using the population distributions aforementioned, arm length increases linearly with height and hence so do the moments involved. However, looking at equation 1, it is clear that although the axial moment increases linearly with height, the loading does not. Meaning that the line is actually curved, but it is unnoticeable over a typical range of heights. The potential considering polynomial predicts a more complicated relationship as it also considers the subjects height and the difference between the L4/L5 FSU and the point of loading. This explains the initial dip in loading as height increases because the distance between the door handle and the FSU has decreased. When the height continued to increase beyond the minimum point, the distance between the door handle and the FSU started to increase again and hence so did the loading on the FSU. Because of the additional consideration of relative load height, it is clear that the potential considering polynomial is more accurate in this situation.

However, Figure 4 demonstrates that the potential considering polynomial has its limitations. As would be expected, the load from the higher order polynomial increases close to linearly as arm length increases in proportion to height, whereas the potential considering polynomial produces a questionable trend, as it suggests that as height increases the loading will decrease drastically. The explanation for this is that the polynomial suggests that as height increases, with all other things being the same, the load decreases. In the previous example, the load height variation effectively counteracted this for a logical distribution but for the two-handed hold the load height is equal to shoulder height minus the FSU height from the ground and hence has a much less profound effect on balancing out the distribution. This clearly shows that for some loading environments the potential considering polynomial is overly complicated and overfit, leading to anomalous data being produced.

The results from this validation suggest that it would be most appropriate to apply the potential considering polynomial to more complicated loading environments were the load height plays a larger role, whereas the higher order polynomial is better at calculating loads for simpler loading environments. This agrees with what the literature of each article suggests, as the potential considering polynomial was designed to be an improvement on the higher order polynomial for compound/ more complicated movements by the researchers (Calder & Potvin, 2012).

3.4 DISCUSSION

As demonstrated in the validation, the compression values generated as boundary conditions for the L4/L5 FSU are accurate for the input values and applied correctly. There are a variety of reasons for this, the most important of which being that all the of the equations and distributions were sourced from peer-reviewed research, hence there were few assumptions. The majority of assumptions present in this project reflect a scarcity of available data rather than a simplifying effort, for example the assumption that the average person's L4/L5 FSU is situated 55% of the way up their body. Another key reason for this accuracy is because each loading environment used the most accurate of three different method for that particular loading scenario, largely as intended by the respective researchers. This prevents extrapolation beyond the data used to generate these methods but still allows for a variety of loading environments to be modelled.

The boundary condition generator was designed to be entirely programmatic and require no inputs once the execution has begun. This means that it obviously has to be designed to be robust to prevent an error or outlying data point from crashing the entire execution of the workflow. To facilitate this the boundary condition generator was designed to be able to handle outlying or extreme data points input from the population data without producing errors or inaccurate data. However, if the worst should happen and the model fail, the workflow will simply move onto the next model, hence no progress is lost. This massively facilitates larger scale ISCTs as it means that the overall workflow reliably executes, saving the operator from the frustration of one failing model preventing the execution of thousands more.

One of the other critical design features of the boundary condition generator is adaptability. Because of the extensive documentation and the three different methods for calculating boundary conditions, the workflow is very adaptable, and the user can easily adapt the workflow to any desired additional loading environments. However, the three different methods are also a disadvantage because the two polynomial equations suggest different boundary conditions which may confuse anyone trying to adapt the programme. Ideally the boundary condition generator would use one unified method for all loading environments, but as discussed in the literature review there is no existing research that has been validated over the range of operation required.

The five loading environments were chosen as they represent a broad cross-section and demonstrate the wide array of conditions that can be modelled for an ISCT.

Standing and two handed by sides are both simple, but fundamental conditions to any potential medical device development on the spine, because these loading environments are some of the most common a human will experience, regardless of age. Two and one-handed horizontal hold both represent somewhat niche postures or loading environments, but they both demonstrate the ease with which a static posture can be programmed into the boundary condition generator. This is hugely advantageous, because the chances of programming the specific static loading environment, someone is interested in is of course, minute. Instead, the workflow has been configured to be easily adaptable to any user, even one who has no Python programming environment. Finally, the usage of the potential considering polynomial is significant as it demonstrates the ability of the boundary condition generator to model more complicated, compound movements. Opening a door is a relatively simple example, but it serves to demonstrate the utility of the more complicated polynomial.

4 MATERIAL REFINEMENT

The default MySpine models have uniform elastic properties roughly equivalent to the average properties of a spine, however as discussed in the literature review these values are not an accurate representation of the spine. To increase the accuracy of the model, new material properties are assigned that vary for the sections of the Functional Spinal Unit (FSU). Originally, the intention was to use the most up to date values from contemporary literature (Mosbah & Bendoukha, 2018) (Grote, et al., 1995) that suggest material properties for the various components of the FSU. Most of these properties were easy to implement, however, the literature suggests anisotropic hyperelastic material properties would be the most accurate for the annulus, but these properties proved to be very difficult to implement. After 40 failed attempts to create a converging anisotropic hyperelastic model, it was determined that those materials would not be possible to implement with present computing resources. There are two reasons for this, firstly, neither of the two versions of ABAQUS available to students are capable of implementing these properties without manually editing the ASCII input file, because the local directions cannot be defined in the GUI or command line (SIMULIA, 2017). Due to the complete lack of documentation on the issue of manually editing the input file(.inp), it proved prohibitively complicated. The second issue was processing power available, as it would take at least 4 hours to execute one of the anisotropic hyperelastic models, only for it to fail before completion. Using the available high-performance computing reduced this time significantly, but the iterative process to find a solution was still extremely labour intensive.

This led to the conclusion that the models would only converge if the material properties were simplified. To this end, the hyperelastic anisotropic annulus material was assumed to be isotropic instead. The suggested approach for modelling the anisotropic properties was using a Holzapfel (Holzapfel, et al., 2000) model with the parameters as specified in table 3 for a healthy annulus. Table 4 displays the material properties assigned should the user request the models be refined.

Table 3- Holzapfel material properties for a healthy annulus.

C_{10}	C_{20}	K_1	K_2	$kappa$
1.5715	9.2044	12.2	39.7	0.113

To minimise the inaccuracy from changing the hyperelasticity model, when the annulus is assumed to be isotropic the Holzapfel model is switched to a Neo-Hookean (Ogden, 1984) model. This is because the Neo-Hookean model uses only the C_{10} and C_{20} material properties, which are already provided, hence no further assumptions are required to apply the isotropic assumption.

Table 4- The refined material properties.

Part	**Young's Modulus E (MPa)**	**Poisson's Ratio v**	**Reference**
Cortical	12000	0.3	(Mosbah & Bendoukha, 2018)
Facets	11	0.4	(Mosbah & Bendoukha, 2018)
Part	$\boldsymbol{C_{10}}$	$\boldsymbol{D_1}$	**Reference**
Healthy Annulus	1.5715	9.2044	(Mosbah & Bendoukha, 2018) (Ogden, 1984)
Degenerated Annulus	1	8.6	(Mosbah & Bendoukha, 2018) (Ogden, 1984) (Grote, et al., 1995)
Nucleus	0.16	0	(Mosbah & Bendoukha, 2018) (Ogden, 1984)

Along with providing material properties for a healthy annulus, Mosbah & Bendoukha (2018) also provide material properties for a degenerated annulus, representative of an elderly or otherwise unwell individual. So, to further increase the accuracy of the population distribution a preassigned ratio of the models are randomly assigned degenerated properties. These material properties are relatively similar to those of the healthy annulus but should help to further introduce outlier data to an ISCT.

4.1 VALIDATION

The used material properties have been validated extensively in multiple peer-reviewed papers, such as (Mosbah & Bendoukha, 2018), (Goel, et al., 1995) and (Natarajan & Andersson, 1999). Similar properties are used in many more peer-reviewed articles; hence, it would be redundant to validate the original material properties. Instead, a validation will be performed to determine the accuracy of assuming that the annulus has hyperelastic isotropic Neo-Hookean material properties, as this is the only departure from the peer-reviewed material properties and to prove the necessity of updating the material properties. In the previous section it was mentioned that the anisotropic properties were unsuccessfully implemented due to convergence issues, however the models do converge at or below 400N of loading. This is unrealistically low and hence unsuitable for use in the workflow, but it does allow for a direct comparison to be made between the outputs of the anisotropic and isotropic models below 400N of loading.

To validate the isotropic assumption, loads between 50N-400N in 50N increments were applied to three models. One of these models had the original, unimproved material properties, the second model had the material properties suggested by Mosbah & Bendoukha, whereas the final model had the same material properties as the previous model, but with the isotropic assumption. The results of this validation are displayed in table 5 below.

Table 5-The results of the material property validation

	Stress (MPa)			**Deflection (mm)**		
Load (N)	**Default**	**Anisotropic**	**Isotropic**	**Default**	**Anisotropic**	**Isotropic**
50	0.16	0.26	0.22	2.76E-08	0.13	0.14
100	0.32	0.55	0.44	5.51E-08	0.25	0.28
150	0.48	0.85	0.66	8.27E-08	0.37	0.40
200	0.63	1.17	0.88	1.10E-07	0.48	0.52
250	0.80	1.51	1.10	1.38E-07	0.58	0.64
300	0.96	1.87	1.48	1.65E-07	0.68	0.74
350	1.12	2.24	1.79	1.93E-07	0.78	0.85
400	1.28	2.64	2.10	2.21E-07	0.87	0.94

The data gathered from the validation suggests a relatively small difference in stress and deflection, thus validating the assumption within this range of data. However, the boundary condition generator often generates loading values close to 2000N, which is significantly outside the validation range. As the validation cannot be expanded, the existing data must be analysed for trends and to try to predict the percentage difference at much higher load values. To this end, two graphs are plotted of the stress and deflection against loading, as Figures 5 and 6.

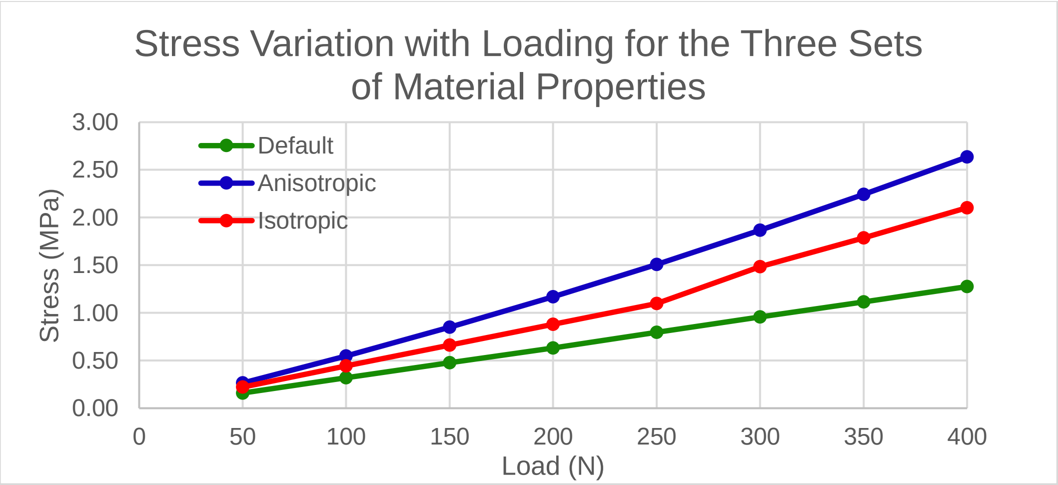


Figure 5- Displays the variation in stress between the models with different material properties.

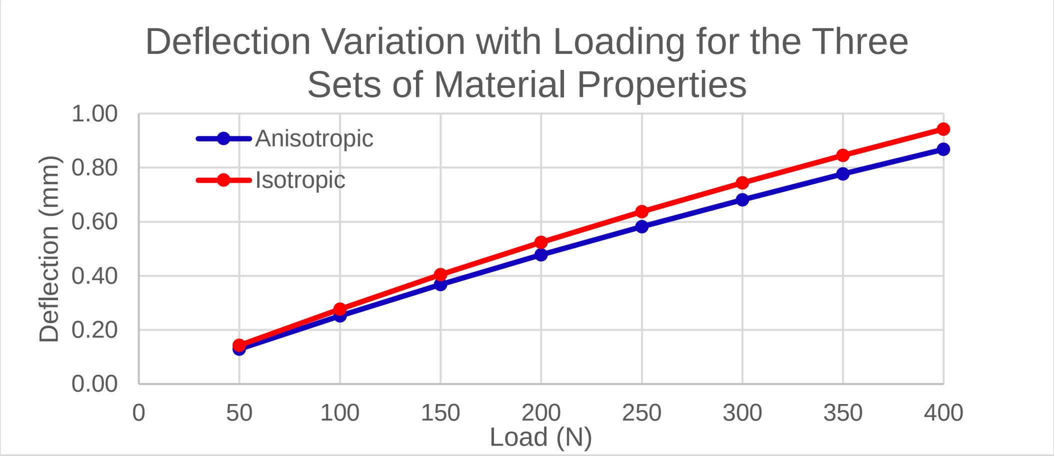


Figure 6- Displays the variation in deflection between the models with different material properties.

By interpreting Figure 5, it is clear that the isotropic assumption leads to a lower calculated stress value than the more accurate anisotropic material properties, however the percentage deviation appears to be constant and hence can be accounted for. As only the models with the default material properties or with the isotropic material properties will converge past 400N of loading, the models cannot be assigned the anisotropic properties. Since the isotropic properties clearly lead to a more accurate stress value than the default properties, their application leads to a more refined model.

Figure 6 does not have the deflection values for the default material properties model plotted as they are several million times lower than the those suggested by the other models. Because of this gigantic disparity, the default material properties are plainly unsuitable. Fortunately, the models with the isotropic material properties assigned produce deflections similar to those of the models with the anisotropic material properties. This validates the isotropic material property assumption as it leads to vastly more accurate results, whilst still producing models that converge over the whole range of generated loading values.

4.2 Discussion

The applied material properties are significantly more accurate than the original uniform properties of the spine, due to the great variation in material properties across one Functional Spinal Unit. These material properties have been extensively validated in literature, but due to limitations with the available software it proved impossible to implement the recommended anisotropic hyperelasticity for the annulus and instead an isotropic assumption was applied, then validated in section 4.1. The validation suggests that the isotropic assumption is acceptable, especially considering that the alternative is that the workflow will not execute. However, as the anisotropic material property models only converge at or below 400N of loading, the validation only covers the range of 50-400N. It is unlikely that any of the data generated by the boundary condition generator would fall within this bound, hence the validation is an extrapolation. This does cast some doubt on the accuracy of the validation as any trends may not be observable within such a small body of data.

The percentage deviation between the isotropic and anisotropic data appears to be relatively constant, hence in operation the user could account for this percentage deviation in an attempt to get data reflective of the anisotropic material properties. This would be an extrapolation but would allow the user to calculate the effect of the isotropic assumption along with estimating the divergence from the true value.

Because stress is a function of only load and area, the only changes to maximum stress values from the isotropic assumption are because of a deformation causing a change in cross sectional area. As the isotropic material properties are close to those of the anisotropic material properties, this difference in cross section deformation is small and hence stress is relatively similar. On the other hand, the original material properties are much less similar and have a correspondingly large difference in predicted maximum stress. This difference is further exacerbated for the deflection, because the default material properties contain no allowance for softer tissues, meaning that the predicted deformation values reflect only the hard cortical bone and as a result are multiple orders of magnitude smaller.

Because of the validation, it would be fair to conclude that the model material properties need to be updated from the inaccurate default material properties, as the models configured with these material properties produce unrealistic results. If the load values generated by the boundary condition generator were always below 400N, the exact material properties collated and used by Mosbah & Bendoukha (2018) would be the most accurate, however the boundary condition generator rarely produces values less than 1000N. Loading values so large would cause the anisotropic models to fail to converge, hence the best option for the material properties is to assume that the anisotropic parts can be modelled as isotropic. This assumption is validated in section 4.1 as producing the most accurate material properties that still successfully converge across the range of loading values.

5 CONCLUSIONS

In this section the success of the project will be evaluated, first by presenting the achievements of the project, then providing an overall discussion, an evaluation will take place in the conclusions section and then finally the recommendations for future work are made.

5.1 ACHIEVEMENTS

- Generated boundary conditions are reflective of the population.
- The code was written and optimised to run efficiently and robustly. In total, over 700 lines of code was written fort this project
- The workflow is easy to use due to the extensive documentation, both in the code and in the technical documentation.
- Robust and sophisticated error handling ensures that if one model fails the others can still be executed and the overall workflow will not fail, this was additional to the initial scope of the project.
- 5 loading environments are modelled by the boundary condition generator. The methods used to generate the values are easily adaptable to other loading environments. This was two more than originally outlined in the scoping document.
- An unlimited number of simulations can be executed using this workflow. The only bottleneck is computing power. To prove this, 500 simulations were executed sequentially, taking over 50 hours.
- The workflow was extensively tested, for over 200 hours of CPU times total in the development stages. Largely because the anisotropic models took over 4 hours to execute.
- To further increase population modelling accuracy, a predefined ratio of models are assigned material properties reflective of someone with a degenerated annulus. This will serve to account for older or unwell people whose spinal health has deteriorated. While this was not in the initial scope of the project it had the additional benefit of demonstration that material properties can also vary realistically with the population.

5.2 CONCLUSIONS

Several goals were achieved that were not in the original scope of the project. The key example being that the models can now be programmatically improved with more accurate material properties and unhealthy/ degenerated annuli can be modelled in the population, as discussed in section 4. This goal was added when it became clear during that the default material properties were inadequate. Numerous quality-of-life improvements were also added, such as functions to output all relevant data into a Comma Separated Variable file for later analysis; a function that cleans-up the working directory by removing files that are generated by the execution of the workflow and an extensive implementation of parallelisation to speed up execution. These quality-of-life improvements serve alongside the documentation to ensure that the operation of the script is easy and clear. Another key addition is the number of loading environments modelled is greater than was originally intended and covers a broader range of activities. Most importantly, these loading environments are easy for a user to alter or expand, hence the workflow could be adapted for different loading environments simply.

The project largely achieved all the goals set out in the scoping document, however there were two exceptions. Firstly, the workflow was initially intended to have a Graphical User Interface, but as the project progressed it became clear that this would have a marginal effect on usability but would have substantial performance ramifications. Put simply, the point of this project was to develop a workflow that would automate a large-scale In-Silico Clinical Trial, so because of the long execution time any performance improvement is preferable to a slightly easier setup. The second unachieved goal was to develop a function for the workflow so that the generated boundary conditions would vary with respect to the proportions of the model they were being applied to. This proved to be unfeasible due to the complete lack of any research about the distribution of the L4/L5 FSU dimensions and proportions.

As mentioned in the section specific discussions, most of the assumptions or sources of error are because of a lack of available data on the relevant population distributions, as opposed to any effort at simplification. This is not ideal but cannot be avoided due to gaps in the existing body of research. The only simplifying assumption is with regards to the difficult to implement anisotropic hyperelastic material properties of the annulus being simplified down to the isotropic properties. As validated, this assumption proved to have a minor impact on the outputs but allowed the models to properly converge. In summation, all of the assumptions and approximations were absolutely required, the only alternative would be far a massively expanded project scope to allow for primary data collection and a greater level of access to finite element software.

The workflow was tested by running 5 simulations on each of 100 models and the workflow successfully performed all aspects of the ISCT. This is hugely significant, because the manual configuration of 500 simulations would have required days of a skilled users time and then still required the same amount of CPU processing time. With this workflow however, the user need only spend 10 minutes specifying parameters in the 'Frontend.py' file and then executing the workflow on the command line. No further user interaction is required and hence the user is free to leave the workflow unsupervised, saving any researcher a huge amount of time. In the aforementioned capacity test, the workflow ran uninterrupted for almost 50 hours, despite two of the 500 simulations failing due to atypical geometry, demonstrating the robust error handling of the workflow. The importance of this in facilitating large-scale ISCTs cannot be stressed enough, as this workflow managed to reduce the amount of skilled labour required to conduct an ISCT by several order of magnitudes. This leads into the conclusion that all large scale ISCTs must feature similar approaches before they can become financially viable and the ISCTs expand to truly being large-scale, instead of a few dozen simulations.

In summation, a workflow (consisting of over 700 lines of Python code) was successfully developed. The project proved so successful that the scope of the project had to be expanded several times, as the initial objectives of the project were met before Christmas. Future research may use this workflow for conducting an ISCT or just to execute a lot of simulations and output relevant data. Regardless of how the workflow is used in future, the project was successful in developing an enormously powerful tool for facilitating future research.

5.3 FUTURE WORK

After completing this project, there are two noticeable areas where further research would be of benefit. Firstly, the greatest barrier to this project has been the lack of data on population distributions and anatomical variation within a population. It would be relatively easy to survey a few thousand people to find anatomical data, then build a unified model that can generate anatomical data representative of a population. If this research were available, the accuracy of the population distributions in boundary condition generator would be massively increased and require far fewer assumptions. Even surveying people just on parameters that they could easily measure themselves, such as arm length, shoulder width and height and then publishing distributions for this data would go a long way towards increasing the accuracy of computer modelling that has to account for a variable population. This would also prove to be useful outside of ISCTs as it would facilitate design not just catering for the average person, but also those with more outlying physical parameters.

More specific to ISCTs, if the ISCTs are to become truly large-scale and valid for medical device approval, there must also be research on how spinal anatomies vary across a population and how to model these distributions. For example, there is a scientific consensus that as people get older, the spinal failure loads decrease and the chance of spinal injury increases. There has been a lot of research about the associated failure mechanisms and why this degeneration occurs, but almost no research numerically quantifying the changing properties of the spine across all age ranges. This further research, as to how the material properties of the spine change with time would help the development of population accurate large-scale ISCTs and would help the development of medical devices for the spine that are effective across a large population range. Similar issues exist with regards to ethnicity and race, in that research exists to suggest that across different ethnicities spinal properties do vary, but there is little research quantifying the data. This means that population modelling tends to assume a homogeneous population, hence any medical product developed using these models may be less efficacious for minority ethnicities.

The second proposed area for future work is the further development of large-scale ISCTs. This project has proven the utility of developing a framework to facilitate the usage of computer modelling in clinical trials, but further research would be required before ISCTs could see widespread adoption. Due to the cost and risk involved it will likely be many years before an ISCT completely replaces a convential clinical trial, if ever. However, the usage of computer modelling in clinical trials will almost certainly continue to increase, meaning that the continued development of validated frameworks would be of great utility in decreasing the costs associated with computer modelling.

6 REFERENCES

Avicenna Alliance, 2016. *In-Silico Clinical Trials: How Computer Simulation will Transform the Biomedical Industry,* Brussels: Avicenna Consortium.

Basey, E., 1986. Demi-span as a measure of skeletal size. *Annals of Human Biology,* 13(5).

BBC News, 2010. *Statistics reveal Britain's 'Mr and Mrs Average'.* [Online]
Available at: https://www.bbc.co.uk/news/uk-11534042
[Accessed 12 02 21].

Beaucage-Gauvreau, E. et al., 2019. Validation of an OpenSim full-body model with detailed lumbar spine for estimating lower lumbar spine loads during symmetric and asymmetric lifting tasks. *Computer methods in biomechanics and biomedical engineering,* 22 (5), pp. 451-464.

Bonet, J. & Wood, R., 2008. *Nonlinear continuum mechanics for finite element analysis.* 2nd ed. Cambridge : Cambridge University Press.

Bruno, A. G. et al., 2017. Spinal Loading Patterns From Biomechanical Modeling Explain the High Incidence of Vertebral Fractures in the Thoracolumbar Region. *Journal of bone and mineral research,* 32(6), pp. 1282-1290.

Calder, I. C. & Potvin, J. R., 2012. instability, A polynomial equation to predict low back compression force: accounting for the effects of load height on. *Work ,* Volume 41, pp. 388-393.

CDC, 2016. Anthropometric Reference Data for Children and Adults: United States, 2011–2014. *Vital and Health Statistics,* Series 3(Number 39).

Cedars Sinai Clinic, 2021. *Anatomy of the Spine.* [Online]
Available at: https://www.cedars-sinai.org/health-library/diseases-and-conditions/a/anatomy-of-the-spine.html#:~:text=The%20average%20person%20is%20born,during%20normal%20growth%20and%20development.
[Accessed 05 March 2021].

Center for Disease Control, 2019. *Vital and Health Statistics,* Atlanta: Centre for Disease Control.

Dalla Man, C., Rizza, R. A. & Cobelli, C., 2007. Meal Simulation Model of the Glucose-Insulin System. *EEE transactions on biomedical engineering,* 54(10), pp. 1740-1749.

Dassault Systems, 2020. *ABAQUS Scripting User's Manual.* 6.5 ed. Vélizy-Villacoublay: Dassault Systems.

Eklund, J., 1988. Industrial seating and spinal loading. *Clinical biomechanics,* 3(4), pp. 225-226.

European Union, 2016. REGULATION (EU) 2016/679 OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL. *Official Journal of the European Union,* Issue 4.5.2016.

Fazzalari, N. L., Parkinson, I. H., Fogg, Q. A. & Sutton-Smith, P., 2006. Antero–postero differences in cortical thickness and cortical porosity of T12 to L5 vertebral bodies. *Joint, bone, spine : revue du rhumatisme,* 73(3), pp. 293-297.

FDA, 2016. *Reporting of Computational Modeling Studies in Medical Device Submissions,* Rockville: FDA.

Goel, V. K. et al., 1995. Interlaminar shear stresses and laminae separation in a disc : finite element analysis of the L3-L4 motion segment subjected to axial compressive loads. *Spine,* 20(6), pp. 689-698.

Grote, H. et al., 1995. Intervertebral variation in trabecular microarchitecture throughout the normal spine in relation to age. *Bone ,* 16(3), pp. 301-308.

Gym Geek, 2018. *How To Get Big Traps With Standing Dumbbell Shrugs.* [Online] Available at: https://gymgeek.com/exercises-routines/shoulder/how-to-get-big-traps-with-standing-dumbbell-shrugs/
[Accessed 12 February 2021].

Han, K.-S., Rohlman, A., Zander, T. & Taylor, W. R., 2013. Lumbar spinal loads vary with body height and weight. *Medical Engineering & Physics,* Volume 35, pp. 969-977.

Hanson, L. et al., 2009. Swedish anthropometrics for product and workplace design. *Applied ergonomics,,* 40(4), pp. 797-806.

Hickson, M. & Frost, G., 2003. A comparison of three methods for estimating height in the acutely ill elderly population. *Journal of human nutrition and dietetics,* 16(1), pp. 13-20.

Holzapfel, G. A., Gasser, T. C. & Ogden, R. W., 2000. A New Constitutive Framework for Arterial Wall Mechanics and a Comparative Study of Material Models. *Journal of elasticity,* 61(1), pp. 1-48.

Marco, V., Claudio, C., Tarek, H. & Adam, H., 2017. In silico assessment of biomedical products: The conundrum of rare but not so rare events in two case studies. *Institution of Mechanical Engineers,* 231(5), pp. 455-466.

McGill, S. M., Norman, R. W. & Cholewichi, J., 1996. A simple polynomial that predicts low-back compression during complex 3-D tasks. *Ergonomics,* 39(9), pp. 1107-1118.

Morrison, T. M., Pras, P., Mariam, A. & Edward, M., 2018. *Advancing Regulatory Science With Computational Modeling for Medical Devices at the FDA's Office of Science and Engineering Laboratories,* Silver Spring: Frontiers in Medecine.

Mosbah, M. & Bendoukha, M., 2018. Anisotropic response of the Holzapfel's constitutive model for the lumbar spine considering degenerative conditions. *Journal of Silicate Based and Composite Materials,* 70(4), pp. 110-114.

Natarajan, R. & Andersson, G., 1999. The Influence of Lumbar Disc Height and Cross Sectional Area on the Mechanical Response of the Disc to Mechanical Loading. *Spine ,* 24(18), pp. 1873-1881.

Natarajan, R. & Andersson, G., 1999. The influence of lumbar disc height and cross-sectional area on the mechanical response of the disc to physiologic loading. *Spine ,* 24(18), pp. 1873-1881.

National Institute of Nutrition, 2020. *Nutrient Requirements for Indians.* Hyderabad: Indian Council of Medical Research.

Ogden, R. W., 1984. *Non-linear elastic deformations.* Chichester: Horwards.

Oxland, T. R., 2015. Fundamental biomechanics of the spine—What we have learned in the past 25 years and future directions. *Journal of biomechanics,* 49(6), pp. 817-832.

Panjabi, M. M. & White, A. A., 1980. Basic Biomechanics of the Spine. *Neurosurgery,* 7(1), pp. 76-93.

Rohlmann, A. et al., 2014. *Activities of Everyday Life with High Spinal Loads,* Berlin: PLOS one.

Sertkaya, A., Jessup, A. & DeVries, R., 2019. *Cost of Developing a Therapeutic Complex Medical Device for the U.S.,* Washington: US Department of Health and Human Services.

Seth, A. et al., 2018. OpenSim: Simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement. *PLoS computational biology,* 14(7).

Shah, T., Patel, M., Nath, S. & Menon, S., 2015. A Model for Construction of Height and Sex from Shoulder Width, Arm Length and Foot Length by Regression Method. *Journal of Forensic Science and Criminology,* 2(4).

Silva, M., Wang, C., Keaveny, T. & Hayes, W., 1994. Direct and computed tomography thickness measurements of the human, lumbar vertebral shell and endplate. *Bone ,* 15(4), pp. 409-414.

SIMULIA, 2017. *Orientations.* [Online]
Available at: https://abaqus-docs.mit.edu/2017/English/SIMACAEMODRefMap/simamod-c-orientation.htm
[Accessed 9 April 2021].

Skimble Workouts, 2021. *Modified Dumbell Front Raises.* [Online]
Available at: https://www.skimble.com/workouts/890655-modified-dumbell-front-raises-jw
[Accessed 12 February 2021].

Spoonamore, M. J., 2018. *Spine Anatomy.* [Online]
Available at: https://www.uscspine.com/spine-health-education/spinal-anatomy/
[Accessed 05 March 2021].

Tan, S., Teo, E. & Chua, H. C., 2004. Quantitative three-dimensional anatomy of cervical, thoracic and lumbar vertebrae of Chinese Singaporeans. *European Spinal Journal,* 13(2), pp. 137-146.

Twisha, S., Patel, M., Nath, S. & Shobhana, K. M., 2015. A Model for Construction of Height and Sex from Shoulder Width, Arm Length and Foot Length by Regression Method. *Journal of Forensic Sciences and Criminology,* 2(4).

Viceconti, M., Henney, A. & Morley-Fletcher, E., 2016. In silico clinical trials: how computer simulation will transform the biomedical industry. *International Journal of Clinical Trials,* Issue May 3, pp. 37-46.

VPH Institute, 2011. *From Challenges to Opportunities: Towards a Common Strategic Framework,* s.l.: VPH Institute.

VPH Institute, 2013. *MYSPINE: PATIENT SPECIFIC SPINAL TREATMENTS SIMULATIONS.* [Online]
Available at: https://www.vph-institute.org/news/myspine-patient-specific-spinal-treatments-simulations.html
[Accessed 07 March 2021].

7 APPENDICES (TECHNICAL DOCUMENTATION)

TABLE OF CONTENTS

7.1 OPERATION

The script this document supports, is designed to facilitate large scale In-Silico Clinical Trials (ISCTs) by developing a workflow to programmatically generate boundary conditions and then apply these boundary conditions to prebuilt models. This will serve to render truly large scale ISCTs far more achievable, as the majority of the work required is performed automatically in this script.

To operate this workflow, the user must input the desired parameters into the ‘Frontend.py’ script (explained in section 1.1) and then execute the ‘Backend.py’ script (explained in section 1.2) on the command line. There are other scripts required but that are called by ‘Backend.py’ and the user should not interact if. All these various scripts are explained in section 3 to facilitate adapting the workflow to a different set of models.

5 different set loading environments can be used to generate boundary conditions, and these are explained in section 2.

7.1.1 Input Parameters

All parameters that may be changed by the user, in standard operation, are evaluated in the script ‘Frontend.py’. These parameters will be presented and explained so the user may make an informed decision as to appropriate values to assign.

The first block of parameters are Boolean values that control which parts of the script will execute when ‘Backend.py’ is executed on the command line. This serves many purposes, such as allowing loading values to be manually specified and saving computational time by only running desired components. These parameters will be presented below, and some suggestions will be made about when these parameters can be set to false.

- **modelbuilder- If true, the models will be imported and formatted for processing.** If the user does not want to import ‘.inp’ model files and process them into ‘.cae’ files, set this parameter to false. Then the user may put any prebuilt ‘.cae’ files in the working directory and they will be used instead, however this may require some changes to ‘job_creator.py’.
- **bcgenerate- If true, boundary conditions will be generated according to input parameters.** If the user does not want to us the boundary condition generator, they may set this parameter to be false and then manually input their own loading data into the file specified by load_csv_file_name.

- **jobcreate- If true, the jobs and input files will be created.** If the user wishes to execute their own '.inp' job files, they may place the '.inp' files into the working directory, set the previous three parameters to zero and then execute the script as usual.
- **jobrun- If true, the jobs will be run.** This is the part of the workflow that takes the longest to complete, so it can be worth running preventing the jobs from running while testing.
- **outputlog- If true an output log will be created.** If an output log is not required, this parameter can be set to false and computational time can be saved.
- **cleanup- If true,'.inp', '.cae' and '.jnl' files will be deleted once no longer useful.** If the user wants to keep these files, they should set this parameter to false,

The next block of parameters are essential to the execution of the workflow, as they specify file locations and toggle some functional aspects.

- **inp_filepath- Specifies the location of the folder containing the '.inp' model files to be imported.**
- **load_csv_file_name- Specifies the name of the '.csv' file that will contain the generated boundary conditions or if boundary condition generation is disabled, specifies the name of the '.csv' file containing the manually specified data.**
- **outputlog_name- Specifies the name of the '.csv' file were the output data will be exported/saved.**
- **cae_creation_num- The quantity of models that should be imported and processed.**
- **cpus- The number of CPUs/ logical processors the jobs should be run on.** By default, this value is set to 4, as this is the limit for most teaching licenses. However, the Student Editions can only use 1 CPU. The user should set this parameter to the largest value that their license will allow.
- **newlog- If true, the contents of the output log will be cleared before additional data is inserted.** Otherwise, the data is appended.
- **degen_ratio- The ratio of annuli that will be assigned degenerated material properties.**

The next block of parameters controls how many boundary conditions are generated for each of the loading environments per model. Details on the boundary condition generator are in section 2.

- **standing_load_num- How many standing load boundary conditions will be generated for each model.**
- **two_handed_horizotal_num- How many two handed, horizonal mass suspension boundary conditions will be generated for each model.**
- **one_handed_horizotal_num- How many one handed, horizonal mass suspension boundary conditions will be generated for each model.**
- **two_handed_by_side_num- How many two hands supporting identical masses with arms down by sides boundary conditions will be generated for each model.**
- **push_open_door_num- - How many pushing open a door boundary conditions will be generated for each model.**

The loads and masses used in the boundary condition generator are defined in the next block of parameters.

- **two_handed_horizontal_mass- The mass supported horizontally with two hands.**
- **one_handed_horizontal_mass- The mass supported horizontally with one hand.**
- **two_handed_vertical_mass- The mass held symmetrically with both arms by sides.**
- **force_to_open_door- The force used to open the door.** By default, this is negative to simulate a pushing force.

The final block of input parameters defines the population distribution data, used to generate boundary conditions representative of a population.

- **average_weight- A tuple of length two in cm units. The first value is the mean average weight of a man and the second is the mean average weight of a woman.** Default values are from the Office of National Statistics in the United Kingdom (BBC News, 2010).
- **std_weight- A tuple of length two in cm units. The first value is the standard deviation of male weight and the second is the standard deviation of female weight.** Default values are from the Centre for Disease Control in the United States (Center for Disease Control, 2019).

- **average_height- A tuple of length two in cm units. The first value is the mean average height of a man and the second is the mean average height of a woman.** Default values are from the Centre for Disease Control in the United States (Center for Disease Control, 2019).
- **std_height- A tuple of length two in cm units. The first value is the standard deviation of male height and the second is the standard deviation of female height.** Default values are from the Centre for Disease Control in the United States (Center for Disease Control, 2019).
- **decimal_places- Number of decimal places for output loads.**

7.1.2 Execution

Once the input parameters have been set, the workflow can be executed. Step by step instructions are provided below.

1. Define a working directory in Windows PowerShell (or non-Microsoft equivalents) that contains the required scripts detailed in section 3.
2. Next the python command line plug-in must be installed from the Microsoft Store (or non-Microsoft equivalents).
3. Install the Numpy python library by typing ‘pip install numpy’ on the command line.
4. Execute the workflow by typing ‘python Backend.py’ on the command line.
5. Wait for completion.

7.2 Boundary Conditions

Loading values for the L4/L5 Functional Spinal Unit (FSU) can be generated for 5 different loading conditions, that are detailed below.

7.2.1 Standing

The work of Han et al., (2013) demonstrates that the spinal loading when standing is almost perfectly linearly proportional to weight, and that height has little impact. Hence to calculate spinal loading when standing, a linear equation for L5 spinal loading was formed using data from the same article. A weight is then randomly generated using the input weight distribution, which is then turned into a l4/l5 spinal loading value with the equation.

7.2.2 Two-Handed Horizontal

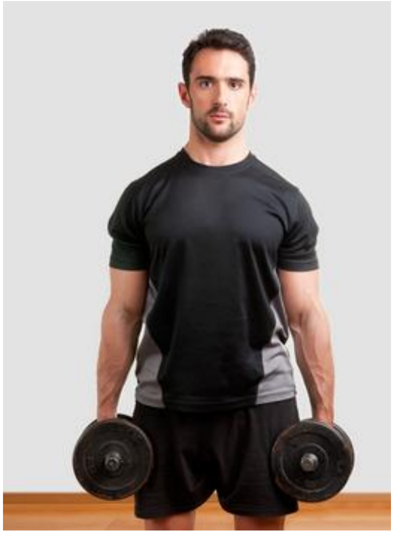
This boundary condition calculates the spinal loading when a mass is supported at shoulder height with both arms horizontal and directly in front of the body. For the purpose of this analysis, it does not matter if it is one mass supported with two hands or two identical masses supported with one hand each. To calculate the L4/L5 spinal load a polynomial equation derived by McGill et al., (1996) is used. This equation requires the calculation of moments around the L4/L5 vertebrae and hence requires a determination of arm length. There is no academic data available about average arm length and standard deviation, hence an arm length distribution was derived from the height distribution. To calculate this distribution, an equation that relates demi-span (distance from palm to palm with arms colinear in a T-pose) to height (Hickson & Frost, 2003) and then subtracting shoulder width. Again, the shoulder width was representative of the population by using an equation that linked height to shoulder width (Twisha, et al., 2015).

Figure 2.2- Holding identical weights at shoulder height with horizontal arms is equivalent to holding one object with two hands under the same conditions. Image from (Skimble

7.2.3 One-Handed Horizontal

This boundary condition calculates the spinal loading when a mass is supported at shoulder height with one arm horizontal and directly in front of the body. It is calculated almost identically to the two-handed version, the only difference being that the asymmetrical loading induces an additional lateral moment instead of just a flexural moment. Fortunately, shoulder width from height is already derived so there are no additional relationships to determine.

7.2.4 Two-Handed By Sides



Due to the distribution of the weights, there are no induced moments around the L4/L5 vertebrae and instead the additional mass acts straight down the spine. This massively simplifies analysis as it means that the additional mass can be added to the generated mass of the individual and then the equation used to determine standing load can be applied again.

Figure 2.4- Holding two identical weights with arms by sides. Image from (Gym Geek, 2018).

7.2.5 Push Open Door

The L4/L5 spinal load is calculated in this boundary condition when a static person pushes open a door. To accomplish this accurately, a mew polynomial equation is used that takes into account how load height affects stability (Calder & Potvin, 2012). The distributions of shoulder width (Hickson & Frost, 2003) and arm length (Twisha, et al., 2015) against height are again used, however the new equation also requires calculating the distance from the load to the L5 vertebrae. No data could be found about the average height of an individual's L5 vertebrae from the floor, and as there no were distributions either, an academic evaluation could not be made. Instead, the author calculated that their l5 vertebrae was positioned 55.6% of their height from the floor and assumed that this percentage did not vary across a population.

When opening a door, only an axial moment is induced around the spine, however this equation can account for flexural and lateral moments. Hence, if additional boundary conditions were to be added, the easiest way to so would be to use this equation again.

7.3 Script Outline

In this section each of the scripts will be outlined, so that the workflow could easily be modified to process a new set of models or otherwise alter the workflow. Hence this section will aid the understanding of the user on sections of the script they would not need to access normally. However, although the user may not interact with any of these scripts apart from 'Frontend.py', all scripts are required in the working directory for a successful execution of the script.

7.3.1 Frontend

This is the script which the user interacts with and is explained in section 1.1. It contains only user-defined parameters and hence is called/imported by all the other scripts to retrieve the desired parameters. If the user wished to define any additional parameters, it must be done in this script if they want the parameter to be accessible across the other scripts.

7.3.2 Backend

'Backend.py' calls the other scripts as required and contains all the python script that should be executed outside an ABAQUS environment. It is also the script that needs to be executed on the command line to begin the workflow, as detailed in section 1.2.

The boundary condition generation and cleaning is performed inside this script, so if the user wanted to make any changes, they must do so in this file.

7.3.3 Model_builder

The user would never need to access this script in standard operation, as it just controls the importing and processing (adding loads and boundary conditions) of the models. However, if the user wanted to adapt the script to import/process different models, this is the only file they would need to change, as long as the load was still named 'Axial Load'.

Due to the specialisation of this script to one set of models, if the user wished to run preprepared models they must disable this script and place the '.cae' files inside the working directory. If the load is still called 'Axial Load', the new models will run. Greater difficulties would occur however, if the user wished to also programmatically import/process these new models as this would require completing rewriting the code for a new model workflow. This would be most easily done using the ABAQUS 'Macro Recorder' tool in consultation with the 'ABAQUS Scripting User's Manual' (Dassault Systems, 2020).

7.3.4 Job_creator

Again, the user would never need to access this script in standard operation as requires no interaction. The 'job_creator.py' script applies the boundary conditions in the specified '.csv' file, then creates the jobs and the job input files. If the user wished to use different models, it is unlikely they would need to edit this script, if the name of the load in the models was still 'Axial Load'.

7.3.5 Data_outputter

'Data_outputter.py' is also one of the scripts the user would never need to access under standard operation. This script outputs key variables into a specified csv folder and should continue to function without error if the models are changed. However, if the load step is changed to run for a different number of frames, this new frame value will need to be specified in this script to ensure that the output data is for the correct frame.

7.4 References

BBC News, 2010. *Statistics reveal Britain's 'Mr and Mrs Average'.* [Online]
Available at: https://www.bbc.co.uk/news/uk-11534042
[Accessed 12 02 21].

Calder, I. C. & Potvin, J. R., 2012. instability, A polynomial equation to predict low back compression force: accounting for the effects of load height on. *Work ,* Volume 41, pp. 388-393.

Center for Disease Control, 2019. *Vital and Health Statistics,* Atlanta: Centre for Disease Control.

Dassault Systems, 2020. *ABAQUS Scripting User's Manual.* 6.5 ed. Vélizy-Villacoublay: Dassault Systems.

Gym Geek, 2018. *How To Get Big Traps With Standing Dumbbell Shrugs.* [Online]
Available at: https://gymgeek.com/exercises-routines/shoulder/how-to-get-big-traps-with-standing-dumbbell-shrugs/
[Accessed 12 February 2021].

Han, K.-S., Rohlman, A., Zander, T. & Taylor, W. R., 2013. Lumbar spinal loads vary with body height and weight. *Medical Engineering & Physics,* Volume 35, pp. 969-977.

Hickson, M. & Frost, G., 2003. A comparison of three methods for estimating height in the acutely ill elderly population. *Journal of human nutrition and dietetics,* 16(1), pp. 13-20.

McGill, S. M., Norman, R. W. & Cholewichi, J., 1996. A simple polynomial that predicts low-back compression during complex 3-D tasks. *Ergonomics,* 39(9), pp. 1107-1118.

Skimble Workouts, 2021. *Modified Dumbell Front Raises.* [Online]
Available at: https://www.skimble.com/workouts/890655-modified-dumbell-front-raises-jw
[Accessed 12 February 2021].

Twisha, S., Patel, M., Nath, S. & Shobhana, K. M., 2015. A Model for Construction of Height and Sex from Shoulder Width, Arm Length and Foot Length by Regression Method. *Journal of Forensic Sciences and Criminology,* 2(4).