Finite Element Modelling of Residual Limb and Transtibial Prosthesis Socket to Predict Interfacial Pressure Considering Muscle Contraction

Manuel Lucas Sampaio de Oliveira

Thesis submitted to the
University of Ottawa
in partial fulfillment of the requirements
for the Doctorate of Philosophy in Biomedical Engineering

Ottawa–Carleton Institute for Biomedical Engineering
Department of Mechanical Engineering
Faculty of Engineering
University of Ottawa
Ottawa, Canada

© **Manuel Lucas Sampaio de Oliveira, Ottawa, Canada, 2025**

To my grandmother Marina,

who taught me the meaning of love, resilience, and kindness.

Your love and memory will forever live in my heart.

Rest in peace knowing you have left a legacy of love.

Abstract

A lower-limb prosthesis is a mechanical device that replaces part of a biological limb to restore mobility. Approximately 7,400 lower-limb amputation surgeries are performed in Canada each year. The tissues in the residual limb (the "residuum") experience atypical stresses. Importantly, the distal end of the residuum must bear the load that would typically be borne by the foot. To avoid pain and tissue damage, the part of the prosthesis that interfaces with the residuum (the "socket") must be designed to fit properly not only at rest but also during walking and other daily activities. The shape of the residuum changes during movement as muscles contract, which can affect the fit of the socket. To study the residuum–socket interface, we first developed a muscle material model that is able to contract, demonstrates the "force enhancement" behaviour observed during active stretch experiments, and is more numerically stable than existing muscle material models. We implemented our muscle material model using an approximation of the elasticity tensor, which simplified the implementation without sacrificing numerical accuracy. We then created a finite element model of the residuum–socket interface using our muscle material model to evaluate the effect of muscle contraction on the interfacial pressure. During gait, the interfacial pressure at heel strike and toe off were greater at key regions of the residuum when the gastrocnemius muscle was active rather than passive. This difference in pressure distribution could influence analyses of prosthesis design and fit. This thesis advances muscle constitutive modelling by providing a verified approximation of the elasticity tensor applied to muscle models, and by providing one of the few muscle material models available in the literature that exhibits force enhancement and is suitable for simulating active stretching. Our muscle material model implementation has been made publicly available so that others can reproduce and extend our results. This work also advances the modelling of the residuum–socket interface as it is the first finite element

modelling study to consider the effect of muscle contraction on residuum shape and socket fit. Our results provide insight into how muscle contraction affects socket fit throughout the gait cycle, potentially leading to improvements in the design and manufacture of prostheses.

Statement of Originality

I, Manuel Lucas Sampaio de Oliveira, hereby declare that the work presented in this dissertation, entitled "Finite Element Modelling of Residual Limb and Transtibial Prosthesis Socket to Predict Interfacial Pressure Considering Muscle Contraction," is the result of my own original research conducted under the supervision of Professor Thomas K. Uchida.

This dissertation has not been submitted in whole or in part for any other degree or qualification at any other academic institution. All sources of information, data, and materials from the work of others have been appropriately cited and acknowledged in accordance with academic standards.

The research findings, data analysis, and conclusions drawn herein are solely my own, except where explicitly stated otherwise. Any assistance received during the course of this research has been duly acknowledged.

Acknowledgements

This thesis marks the realization of a dream——a milestone in a significant journey, both academically and personally. It would not have been possible without the guidance, support, and encouragement of numerous individuals.

First and foremost, I extend my deepest gratitude to my advisor, Prof. Thomas K. Uchida, for believing in me and for giving me the honor of being your first PhD student. Your invaluable guidance, insightful feedback, and unwavering support throughout this research have made me a better researcher, professional and human. I am also grateful to my colleagues and friends at the University of Ottawa. Thank you for creating a collaborative and stimulating environment that fueled my intellectual growth.

On a personal note, I am deeply indebted to my family for their unconditional love and encouragement. To my parents, Manuel Rômulo (in memoriam) and Solange, thank you for instilling in me the values of perseverance and curiosity and for believing in me even when I doubted myself. I would not be here without you. To my brother, Vinícius, I cannot thank you enough for being my foundation and for holding the fort during the toughest times. I would also like to thank my grandmother, Marina (in memoriam), whose love and support are beyond words.

To my beloved girlfriend, Agata, your love, support, and belief in me have been my greatest source of strength. Sharing this journey in life with you has been one of my greatest joys. Thank you for standing by my side, celebrating my achievements, and supporting me every single day through every challenge.

To those who helped me unwind and disconnect from work, I owe special thanks to my friends in the "Crackudos de Ottawa" group——Andre, Carlao, Junior, Omar, and Rodrigo——for the laughter, banter, and "azedices." Life in Ottawa was great because of you. Thank you, James, for sharing NFL games with me and being a loyal friend.

Finally, I dedicate this work to those who have experienced lower limb amputation. You are my greatest motivation, and I hope my work, in some way, can contribute to making life easier for you.

Nomenclature

α	Muscle activation
α_0	Unfolding rate of proximal Ig domain at zero force
$\alpha\left(F_{\mathrm{Ig}}\right)$	Unfolding rate of proximal Ig domain
β_0	Folding rate of proximal Ig domain at zero force
$\beta\left(F_{\mathrm{Ig}}\right)$	Folding rate of proximal Ig domain
$\Delta\mathbf{F}$	Perturbation of the deformation gradient
$\Delta\ell$	Value of muscle lengthened in the benchmarking simulations
ϵ	Perturbation parameter to deformation gradient
λ	Fiber stretch
λ^*	Fiber stretch above which the passive function is linear
λ_{act}	Fiber stretch at the onset of activation
λ_{min}	Minimum fiber stretch at which cross-bridges can form
λ_{opt}	Optimal fiber stretch
ν	Poisson's ratio
Ω	Current configuration
Ω_0	Reference configuration
Ψ	Strain energy function
Ψ_{iso}	Isochoric component of the strain energy
Ψ_{vol}	Volumetric component of the strain energy
$\boldsymbol{\sigma}$	Cauchy stress
σ_{disp}	Stress at the displaced end of the muscle model in the benchmarking simulations
σ_{fiber}	Cauchy stress generated by the fiber stretch
σ_{max}	Maximum isometric stress
τ	Kirchhoff stress
ζ_{Ansys}	Stress or longitudinal strain from Ansys calculated from the simulation of the approximation of the tangent modulus

ζ_{FEBio}	Stress or longitudinal strain from FEBio calculated from the simulation of the approximation of the tangent modulus
A	Fiber cross-sectional area
$\mathbf{a_0}$	Unit vector in the direction of the fiber in the referential configuration
A_{disp}	Area at the displaced end of the muscle model in the benchmarking simulations
A_{mid}	Mid-belly area of the muscle geometry used in the benchmarking simulations
A_{sar}	Sarcomere area
$\mathbf{B}$	Left Cauchy–Green deformation tensor
$\bar{\mathbf{B}}$	Modified left Cauchy–Green tensor
B_1	Along-fiber shear invariant
B_2	Cross-fiber shear invariant
b_u	Width of the activation barrier for an unfolding event to occur
b_f	Width of the activation barrier for a folding event to occur
C	Shear modulus in Neo-Hookean formulation
$\mathbf{C}$	Right Cauchy–Green deformation tensor
$\bar{\mathbf{C}}$	Modified right Cauchy–Green tensor
$\mathbb{C}$	Material tangent modulus
$\mathbb{C}^J$	Tangent modulus associated with the objective Jaumann rate of Cauchy stress
$\mathbb{C}^S$	Spatial tangent modulus
$\bar{\mathbb{C}}_{\text{iso}}$	Fictitious material elasticity tensor
$\bar{\mathbb{C}}_k$	Fictitious material elasticity tensor calculated with respect to each term of the isochoric strain energy function W_1, W_2, and W_3
d	Half the width of the A-band region of a sarcomere
D	Incompressibility parameter in Neo-Hookean formulation
$\mathbf{D}$	Symmetric part of the spatial velocity gradient
d_{sar}	Sarcomere diameter
E	Young's modulus
$\mathbf{F}$	Deformation gradient
$\hat{\mathbf{F}}$	Perturbed deformation gradient
F_o^M	Maximum isometric force
f_{active}	Active force normalized by the maximum isometric force

$F^{\text{mus}}_{\text{active}}$	Active muscle force
F_{calc}	Calculated titin force
F_{disp}	Force produced at the displaced end of the muscle model in the benchmarking simulations
F_{exp}	Titin force reported from experimental measurements
F_{Ig}	Force applied to the proximal Ig segment
f_{passive}	Passive force normalized by the maximum isometric force
$f^{a=0}_{\text{passive}}$	Normalized version of titin force that would be produced during a purely passive stretch
$f^{a=1}_{\text{passive}}$	Normalized version of titin force that would be produced during a fully active stretch
$F^{\text{mus}}_{\text{passive}}$	Passive muscle force
F_{PEVK}	Force applied to the PEVK segment
f_{total}	Total fiber force
$f^{a=0}_{\text{titin}}$	Titin force produced during a purely passive stretch
$f^{a=1}_{\text{titin}}$	Titin force produced during a fully active stretch
G_1	Along-fiber shear modulus
G_2	Cross-fiber shear modulus
h	Half-length of the muscle geometry used in the benchmarking simulations
$\mathbf{I}$	Identity matrix
I_{k}	Invariants of the modified right Cauchy–Green tensor, with $\text{k} \in \{1, 2, 4, 5\}$
ISL	Sarcomere length at the onset of activation
J	Volumetric ratio
K	Effective bulk modulus
K_0	Stretch modulus parameter
$K_{0,\text{active}}$	Stretch modulus parameter of the PEVK segment corresponding to the active state
$K_{0,\text{ISL}}$	Stretch modulus parameter of the PEVK segment as a function of the sarcomere length at the onset of activation
$K_{0,\text{passive}}$	Stretch modulus parameter of the PEVK segment corresponding to the passive state
k_B	Boltzmann constant
L	Final length of the element from the simulation of the approximation of the tangent modulus

$\mathbf{L}$	Spatial velocity gradient
ℓ	Muscle Length
L_0	Initial length of the element from the simulation of the approximation of the tangent modulus
$\ell_{\text{fib}}^{\text{opt}}$	Optimal fiber length
ℓ_{HS}	Half-sarcomere length
L_{Ig}	Contour length of one Ig domain
$\ell_{\text{Ig}}^{\text{dist}}$	Length of the distal Ig segment
$\ell_{\text{Ig}}^{\text{prox}}$	Length of each proximal Ig domain
$\ell_{\text{Ig,f}}^{\text{prox}}$	Length of one proximal Ig domain when folded
$\ell_{\text{Ig,u}}^{\text{prox}}$	Length of one proximal Ig domain when unfolded
L_{PEVK}	Total contour length of the PEVK segment
ℓ_{PEVK}	Length of the PEVK segment
L_{PEVKr}	Contour length of one PEVK residue
ℓ_{sar}	Sarcomere length during active stretch
$\ell_{\text{sar}}^{\text{max-act}}$	Sarcomere length at which the descending limb of the active FL curve reaches zero force
$\ell_{\text{sar}}^{\text{opt}}$	Optimal sarcomere length
N^{PEVKr}	Number of PEVK residues
N^{prox}	Number of proximal Ig domains
N_f^{prox}	Number of folded proximal Ig domains
N_u^{prox}	Number of unfolded proximal Ig domains
p	Hydrostatic pressure
PCSA_{rat}	Physiological cross-sectional area of a typical rat tibialis anterior muscle
p_{Ig}	Persistence length of one Ig domain
p_{Ig}^f	Persistence length of one folded Ig domain
p_{Ig}^u	Persistence length of one unfolded Ig domain
P_{k}	Constants in the normalized passive force, with $\text{k} \in \{1, 2, 3, 4\}$
p_{PEVK}	Persistence length of the PEVK segment
$p_{\text{PEVK,active}}$	Persistence length of the PEVK segment corresponding to the active state

$p_{\text{PEVK,ISL}}$ Persistence length of the PEVK segment as a function of the sarcomere length at the onset of activation

$p_{\text{PEVK,passive}}$ Persistence length of the PEVK segment corresponding to the passive state

P_u Probability of Ig domain unfolding

$\mathbf{R}$ Rotation matrix

r_1 Minimum radius of the muscle geometry used in the benchmarking simulations

r_2 Maximum radius of the muscle geometry used in the benchmarking simulations

$r(x)$ Radius along the length of the muscle geometry used in the benchmarking simulations

s Longitudinal deformation of each element from the simulation of the approximation of the tangent modulus

$\mathbf{S}$ Second Piola–Kirchhoff stress tensor

$\bar{\mathbf{S}}$ Fictitious second Piola–Kirchhoff tensor

$\mathbf{S}_{\text{iso}}$ Isochoric second Piola–Kirchhoff stress tensor

$\bar{\mathbf{S}}_{\text{iso}}$ Fictitious isochoric second Piola–Kirchhoff stress tensor

$\bar{\mathbf{S}}_k$ Fictitious second Piola–Kirchhoff stress tensor with respect to each term of the isochoric strain energy function W_1, W_2, and W_3

$\mathbf{S}_{\text{vol}}$ Volumetric component of the second Piola–Kirchhoff tensor

T Absolute temperature

$\mathbf{U}$ Right stretch matrix

v Muscle velocity

W_1 Along-fiber shear strain energy

W_2 Cross-fiber shear strain energy

W_3 Muscle fiber stretch strain energy

$\mathbf{x}$ Position in the current configuration

$\mathbf{X}$ Position in the reference configuration

x_{Ig} Elongation of one domain in the proximal Ig segment

$x_{\text{Ig}}^{\text{ISL}}$ Elongation of one proximal Ig domain at the onset of activation (inital sarcomere length)

x_{PEVK} Elongation of the PEVK segment

x_{sar} Sarcomere stretch

$x_{\text{sar}}^{\text{initial}}$	Sarcomere stretch at the onset of activation with respect to the optimal sarcomere length
x_{titin}	Elongation of titin filament

Table of Contents

List of Figures **xvii**

List of Tables **xix**

1 Introduction **1**
1.1 Background on lower-limb amputation and prostheses 1
1.2 Current challenges 4
1.3 Research objectives and contributions 7
1.4 Thesis organization 10
Bibliography for Chapter 1 12

2 Literature Review **18**
2.1 Muscle biology 18
2.2 Force enhancement during eccentric contraction 23
2.3 Models of muscle force generation 26
2.4 Hyperelastic material equations 29
2.5 Muscle model of Blemker et al. (2005) 31
2.6 Finite element models for residuum and prosthesis mechanics 34
2.7 Variation of volume and shape of the residuum 44
Bibliography for Chapter 2 48

3 Muscle Constitutive Model with a Tangent Modulus Approximation: Ansys Implementation and Verification **58**
3.1 Abstract 60
3.2 Introduction 61
3.3 Methods 64
3.3.1 Kinematics 65
3.3.2 Stress measures 65
3.3.3 Tangent modulus and its approximation 67
3.3.4 Hyperelastic muscle material model 69
3.3.5 Models to verify the approximation 71
3.3.6 Simulations to verify the approximation 72
3.4 Results and discussion 74
3.4.1 Selection of the perturbation parameter 74
3.4.2 Simulations of RR, RTR, and RTO models 76
3.4.3 Simulations of RTR model variants 79
3.4.4 Sensitivity to the effective bulk modulus 81
3.5 Conclusions 83
Bibliography for Chapter 3 85

4 Phenomenological Muscle Constitutive Model With Actin–Titin Binding for Simulating Active Stretching **90**
4.1 Abstract . . . 92
4.2 Introduction . . . 93
4.3 Methodology . . . 99
4.3.1 Micromechanical model of actin–titin binding . . . 99
4.3.2 Parameters for PEVK segment during active stretching . . . 102
4.3.3 Parameters for worm-like chain model . . . 103
4.3.4 Macro-scale constitutive model with force enhancement . . . 105
4.4 Numerical examples . . . 110
4.4.1 Parameters and curves used in numerical examples . . . 110
4.4.2 Numerical stability (Tests 1A and 1B) . . . 112
4.4.3 Benchmark using 3D muscle geometry (Test 2) . . . 114
4.4.4 Fiber length distribution in an isometric contraction (Test 3) . . . 116
4.5 Results and discussion . . . 117
4.6 Conclusions . . . 121
4.7 Supplementary Material: Normalized active force–length curve . . . 124
4.8 Supplementary Material: Calculation of normalized passive force–length curve . . . 125
Bibliography for Chapter 4 . . . 131

5 Pressure Between Residual Limb and Transtibial Prosthesis Socket Considering Muscle Contraction: a Finite Element Study **139**
5.1 Abstract . . . 140
5.2 Introduction . . . 141
5.3 Methodology . . . 145
5.3.1 Geometry . . . 146
5.3.2 Material models . . . 148
5.3.3 Mesh . . . 150
5.3.4 Contact and boundary conditions . . . 151
5.3.5 Simulation steps . . . 152
5.3.6 Sensitivity analysis . . . 153
5.3.7 Model analysis . . . 154
5.4 Results . . . 154
5.4.1 Benchmarking against an existing model . . . 155
5.4.2 Sensitivity of predicted interfacial pressure to muscle parameters . . . 156
5.4.3 Effect of muscle activation and size on residuum–socket pressure . . . 157
5.5 Discussion . . . 158
Bibliography for Chapter 5 . . . 165

6 Conclusions **171**
6.1 Summary of literature gap . . . 172
6.2 Summary of key findings . . . 173
6.3 Limitations of the studies . . . 176
6.4 Future work . . . 178

Bibliography for Chapter 6 180

A Software Implementations **183**
A.1 Approximation of the elasticity tensor in Fortran 183
A.2 Proposed muscle material model in C++ 214

List of Figures

1.1 Anatomy of an individual with transtibial amputation. 2
1.2 Components of a transtibial prosthesis. 3
2.1 Muscle structure from the molecular level (actin and myosin forming cross-bridges) to the macro level (muscle fascicle). 19
2.2 Sarcomere active force–length curve. 21
2.3 The total force–length relationship of muscle (blue) is the sum of the active (green) and passive (orange) components. 22
3.1 Models used to verify our Ansys implementation. 3D models were generated by revolving (a) a rectangle (RR), (b) a right trapezoid (RTR), and (c) an obtuse trapezoid (RTO) around the centerline. 72
3.2 Visualization of fiber stretch after deformation of (a) the RR model, (b) the RTR model, and (c) the RTO model. 77
3.3 Stress and strain from Ansys and FEBio simulations, for elements along the length of the RR, RTR, and RTO models. 78
3.4 Longitudinal stress (a) and longitudinal strain (b) from Ansys and FEBio simulations, for elements along the centerline of the RTR model with three inclination angles (θ, as labeled). 79
3.5 Longitudinal stress and longitudinal strain from Ansys ((a) and (b)) and FEBio ((c) and (d)) for elements along the centerline of the RTR model ($\theta = 54.58°$), using four values of parameter K. 82
4.1 Typical shapes of force–length curves used in Hill-type models. 95
4.2 Model of a half-sarcomere during passive and active stretches. 97
4.3 Stress vs. half-sarcomere length calculated by the proposed worm-like chain (WLC) model with experimental data from stretching myofibrils under three conditions: (a) passive stretch, (b) active stretch from the optimal sarcomere length (2.4 µm), and (c) active stretch from a sarcomere length on the descending limb of the active FL curve (3.4 µm). 112
4.4 Normalized stress vs. normalized fiber length predicted by our simulations of active stretching. 113
4.5 Geometry of 3D filament model used in the stability analysis. 114
4.6 3D muscle geometry used to demonstrate force enhancement. 115
4.7 Convergence analysis for mesh refinement of the 3D muscle geometry. . . 115
4.8 Eigenvalues of the spatial elasticity tensor during an active stretch of a unit cube using three muscle material models: (a) the model of Blemker et al. [1], (b) the model of Blemker et al. augmented with our model of force enhancement, and (c) our proposed model. 118
4.9 Fiber stretch in each element of a 3D filament model during an active stretch using our proposed model, where the fibers were initially of unequal normalized lengths. 119

4.10 Three-step simulation of active stretching using the 3D muscle geometry with our proposed model. 120
4.11 To model the normalized active force–length relationship, we replaced the piecewise-quadratic function proposed by Blemker et al. [1] (solid red line) with a Weibull distribution [60] (dotted blue line). 124
4.12 Process to calculate the normalized passive FL curve during passive stretching. 126
4.13 Subprocess to calculate the number of folded and unfolded proximal Ig domains. 127
4.14 Process to calculate the elongation of one proximal Ig domain at the onset of activation. 129
4.15 Process to calculate the normalized passive FL curve during active stretching at maximal activation. 130
5.1 Geometry used in our finite element simulations. 147
5.2 Muscle geometry created for our finite element models. 148
5.3 Regions in which residuum–socket pressures were considered. 155
5.4 Residuum–socket pressures predicted by the Small-Muscle Model as activation (α) varied from 0 (passive) to 1 (fully active). 157
5.5 Residuum–socket pressures predicted by the Small-Muscle Model and Standard-Size-Muscle Model when the muscle was passive ($\alpha = 0$) and fully active ($\alpha = 1$). 158
5.6 Colour map of the 95th percentile residuum–socket pressure at heel strike predicted by the Small-Muscle Model. 159
5.7 Lateral view of the socket at heel strike in the Small-Muscle Model with a passive muscle (transparent) and a fully active muscle (opaque). 162

List of Tables

2.1 Summary of existing work on finite element analysis of the residuum–socket interface. . . . 36
3.1 Number of elements and nodes in each model. . . . 72
3.2 Number of iterations required to reach equilibrium at each simulation step for different values of perturbation parameter ϵ, using the RR model. . . . 75
3.3 Sensitivity of Von Mises stress and longitudinal strain to perturbation parameter ϵ, using the RR model. . . . 75
3.4 Error in Von Mises stress, longitudinal stress, and longitudinal strain between Ansys and FEBio simulations for the RR, RTR, and RTO models. 77
3.5 Error in longitudinal stress and longitudinal strain between Ansys and FEBio simulations for the three variants of the RTR model. . . . 80
4.1 Parameters used in the worm-like chain (WLC) equations for titin during passive and active stretches. . . . 111
4.2 Validation of sarcomere lengths during simulation of 3D muscle geometry. 121
4.3 Sets of parameters used to calculate the normalized passive force–length curve. . . . 128
5.1 Parameters for the socket, liner, and residuum material models [29] and muscle material model [33, 37]. . . . 149
5.2 Element size settings used to create the meshes in our Muscleless, Standard-Size-Muscle, and Small-Muscle Models. . . . 151
5.3 Principal axes of the tibia. . . . 153
5.4 Loads applied to the inferior tip of the socket with respect to the tibia's principal axes. . . . 153
5.5 Benchmark of our Muscleless Model. . . . 156

1

Introduction

This chapter provides a background on lower-limb amputation and prostheses, describes the current challenges in prosthesis design, and summarizes the aims and structure of the thesis.

1.1 Background on lower-limb amputation and prostheses

In 2015, between 30 and 40 million people worldwide lived with amputation [1], and there are approximately one million amputation surgeries performed per year, or one every 30 seconds [2]. Among the causes of amputation, the most common are vascular disease (82%), trauma, cancer, and congenital anomalies [3]. The most common amputations are transtibial (below the knee) and transfemoral (above the knee), as shown in Fig. 1.1. The part of the limb that remains after an amputation surgery is called the "residual limb" or simply "residuum."

Despite having different shapes and designs, transtibial prostheses generally have the same five components: the socket, compliant interface material, suspension, structure, and foot/ankle [5] (Fig. 1.2). The socket connects the prosthesis to the residual limb. The design of the socket is of great importance because the socket is the interface between the residual limb and the rest of the prosthesis. The compliant interface material is located between the socket and the residual limb, and may be made of foam or may be a liner or thick sock made of urethane, silicone, or elastomer. The patient and prosthetist

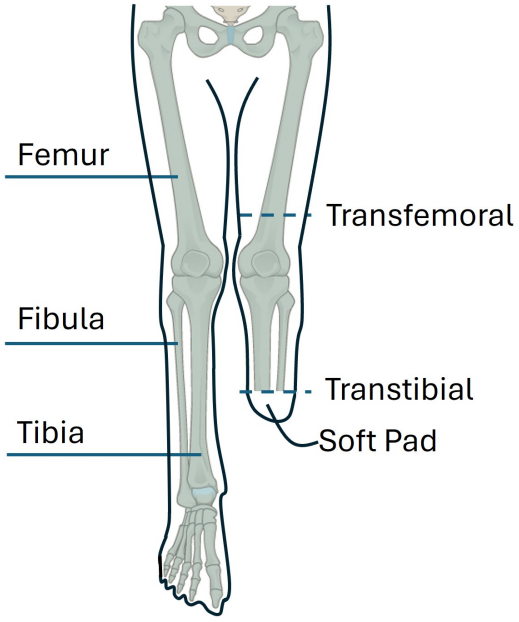

Figure 1.1: Anatomy of an individual with transtibial amputation. Dashed lines: typical amputation levels. Adapted from OpenStax [4] under CC-BY-4.0 license.

sometimes opt for direct contact between the residuum and the socket, without any interface material [5]. The suspension component is responsible for keeping the socket attached to the residual limb [6, 7]. For example, the suspension mechanism can involve pin and lanyard locks, straps, or vacuum suction [8].

The remaining two components of a transtibial prosthesis are the pylon and the foot/ankle. The pylon connects the socket to the foot/ankle, enabling the prosthetist to adjust the prosthesis to the appropriate length and to align the other prosthesis components. The foot/ankle is the part of the prosthesis that is designed to mimic the mechanical function of the anatomical foot/ankle. The foot is usually interchangeable and many options exist, including nonarticulated devices (e.g., Solid Ankle Cushioned Heel), articulated devices (single-axis or multi-axial), energy-storing devices, and microprocessor-controlled devices [10].

There are two main strategies for designing the physical interface between the limb and the prosthesis socket: specific surface bearing and total surface bearing. These types

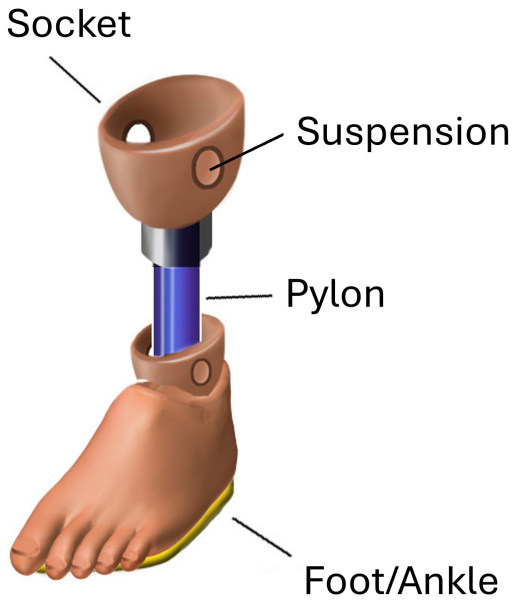

Figure 1.2: Components of a transtibial prosthesis. Adapted from BruceBlaus [9] under CC-BY-SA-4.0 license

of interface differ in the way that the socket is attached to the residual limb. Total surface bearing prostheses attach around the entire residuum volume, which distributes the load with the intention to avoid stress concentrations. Specific surface bearing prostheses attach only to specific areas of the residuum, typically at regions that are least sensitive to pain. The most popular specific surface bearing prosthesis is the patellar tendon bearing for transtibial amputation [11].

The residual limb and the remaining soft tissues are responsible for bearing the body weight and other loads generated during motion. For example, during transtibial amputation, a soft tissue pad is created over the cut ends of the tibia and fibula bones from the remaining tissues posterior to the bones, including the gastrocnemius muscle (Fig. 1.1). During prosthesis-assisted gait, the soft tissue pad is subjected to transverse and axial loads, potentially causing pain and skin damage [12]. Improving the design of prosthesis sockets can help to mitigate the effect of an amputation on an individual's mobility.

The current process for prosthesis socket manufacturing is time consuming and relies on the prosthetist's experience and skill [11]. On average, prosthesis users visit the prosthetist nine times during the first year of prosthesis use to improve the fit of the device [13]. The manual aspects of the prosthesis manufacturing process reduce repeatability [14]. Furthermore, an improper fit can generate pain, produce discomfort, and reduce the quality of life of the wearer [3]. Even mild pain can result in changes to posture or gait, which can then lead to other issues such as chronic back pain [15].

The design of prosthesis sockets may be improved by complementing the experience of prosthetists with analyses from numerical simulations. Finite element analysis is a numerical method that divides a physical system into smaller parts called finite elements; the dynamics of each of these elements is governed by a constitutive model that describes the material's behaviour mathematically. Finite element models can facilitate the evaluation of candidate socket designs, new materials, and various loading conditions without the need to create physical prototypes or perform experiments.

1.2 Current challenges

Individuals with lower-limb amputation lack a foot and ankle, causing the remaining muscles and other tissues to be subjected to forces, moments, and environmental conditions (e.g., temperature and humidity) to which these tissues are not accustomed. For instance, during prosthesis-assisted gait, the distal end of the residual limb bears the load typically borne by the foot. High pressures and stresses can lead to cutaneous blood flow occlusion and skin breakdown [16]. These effects are exacerbated by elevated temperature and humidity [17]. The prolonged pressure and strain on the residuum can cause deep tissue injury [18].

Dissatisfaction with the prosthesis is a common issue among individuals with amputation. The rate of rejection for prosthetic limbs does not have a standard definition; however, it can be viewed as the proportion of users who report being dissatisfied with their prosthesis. By this definition, the rate of rejection for lower-limb prostheses is high,

occurring in about 57% of individuals [12]. Skin damage contributes to a higher rate of rejection and affects up to 55% of lower-limb prosthesis users [3]. The rate of rejection can be assessed by questionnaires, one of the most popular of which is the Prosthesis Evaluation Questionnaire [19].

In recent years, several socket designs have been proposed to improve load distribution and increase prosthesis acceptance. However, numerous designs still fail to address the issues of non-uniform pressure distribution, non-uniform shear stress distribution, temperature variation, and limb volume fluctuation [3]. These four factors are the main contributors affecting the comfort of prosthetic limbs and causing skin damage [12] and, thus, are likely to cause prosthesis rejection.

Changes in the volume and shape of the residuum can affect the stresses on its surface. The volume of the residuum changes dramatically in the period between 12 and 18 months following an amputation surgery [20]. However, the residual limb undergoes volume and shape changes even after the post-operative period. Short-term volume changes occur throughout the day due to the movement of interstitial fluids when the residual limb experiences pressure and shear stress [21, 22]. The residual limb also undergoes rapid changes in shape due to the activation of the muscles within it [23, 24], which are typically still innervated [25]. If the volume of the residuum decreases, it can move within the socket causing a pistoning effect (vertical relative motion between the residuum and socket) and increasing the stress at the distal end [3]. On the other hand, if the volume of the residuum increases, the stress on the skin can increase in regions where the residuum cross-section increases [24]. Consequences of increases in stress include discomfort, pain and, ultimately, skin ulceration [26].

Muscle activity plays an important role in how the socket interfaces with the residual limb. Lilja et al. [23] emphasized the important question of whether prosthetists should consider the effect of muscle contraction when manufacturing the prosthesis. Lilja et al. also acknowledged that, as of the late 1990s, only a few studies had explored how muscle contraction affects prosthesis fit. More recently, this statement was corroborated

by Paternò et al. [11, 24], who confirmed the need to consider muscle shape changes during socket design.

A finite element model of a residual limb with a prosthesis socket can be used to explore the effect of muscle contraction on the interfacial pressure. The development of models that incorporate muscle contraction is hindered by a lack of built-in muscle material models that account for muscle activity in widely used commercial finite element software, such as Ansys and Abaqus. The difficulty of implementing even simple user-defined muscle models has hindered the widespread adoption of state-of-the-art muscle material models when using commercial software packages. Consequently, all existing studies involving finite element simulations of a residual limb with a prosthesis socket ignore muscle contraction [27].

The muscle material models available in the literature are limited in their applicability to finite element simulations. Muscle force is typically modelled as the sum of two components: a passive force, which is always present, and an active force, which is modulated by activation signals from the nervous system. Many finite element simulations employ muscle material models that either account only for passive force generation [28] or simulate active force based on experimental data from purely isometric contractions, where muscle fibers are activated while held at a constant length [29]. Forces measured during active stretching experiments, where muscle fibers are stretched after being activated, are always greater than the forces measured at the same length but where the muscle had first been stretched passively and then activated [30]. The increase in the observed force following active stretching is called force enhancement [31]. Most simulations using finite element methods neglect this effect [32–35]. Neglecting force enhancement may affect answers to research questions that are sensitive to muscle behaviour. Furthermore, when force enhancement is not considered in simulations involving active stretching, the simulations may be numerically unstable due to a negative stiffness in the total fiber force–length curve [36]. In this thesis, we use the term "numerical stability" to describe the robustness of muscle material models during finite element simulations, particularly

their ability to avoid convergence issues and nonphysical behaviours. The region of negative stiffness in the total fiber force–length relationship presented in some muscle models [33, 35, 37] can cause numerical instability in simulations involving active stretching. The region of negative stiffness leads to a strain energy function that violates convexity conditions (such as strong ellipticity, which ensures a unique solution to the equilibrium equation, and polyconvexity, a broader condition that guarantees the existence of solutions by requiring that the strain energy function be convex [38]). These convexity conditions are essential in hyperelastic material models to avoid unstable or nonphysical behaviour during simulations of large deformations [39, 40]. In the literature, muscle material models have often been used in finite element simulations without first establishing that the muscle models are numerically stable [37].

1.3 Research objectives and contributions

As discussed above, discomfort is the main cause of rejection of prostheses; therefore, improving their design has the potential to improve the lives of individuals with amputation—reducing pain and improving mobility. In this thesis, we consider the change in shape of the residual limb over a gait cycle, which is one of the primary causes of prosthesis discomfort. Pain is a complex experience that depends on pressure, shear stress, temperature, humidity, relative motion between the residual limb and the socket, and many other physical and psychological factors [11]. Pressure at the interface between the residual limb and the socket has been used to study prosthesis fit and possible user discomfort in an objective way [41].

The change in the residuum volume is influenced by several factors; however, our work will focus on answering the question of **how the predicted pressure over the interface between the residual limb and prosthesis socket is affected by muscle contraction during gait**. To understand the effect of muscle activation and the consequent muscle bulging and changes in the shape of the residuum, we will use Ansys to create a finite element model (FEM) of the residuum and socket. Past work has

used finite element analysis to study boundary conditions and constitutive models in residuum–socket models [27]; however, to the best of our knowledge, no existing studies have considered muscle contraction (this is discussed in more detail in Section 2.6). We develop a muscle material model in the open-source finite element software FEBio, as well as in Ansys using the USERMAT subroutine to implement a custom material model. The stress on the residuum surface can be measured by force transducers; however, the use of force transducers can produce stress concentrations and modify gait, and the results are restricted to the location of the sensor [42]. FEMs have the potential to overcome these experimental limitations and enable the prediction of the stress–strain state over the entire limb surface.

This thesis has three main research contributions:

1. **Implementation of a method to approximate the tangent modulus tensor of a muscle material model based on the work of Miehe [43].** The tangent modulus or elasticity tensor is a fourth-order tensor that describes the stress–strain relation of the material model. The Ansys USERMAT subroutine has as inputs the deformation gradient and parameters for the material model, and outputs the deformation gradient tensor and the tangent modulus tensor. Implementing a material model can be challenging as deriving and programming the calculation of a fourth-order tangent modulus tensor for an intricate muscle material model requires considerable algebraic manipulation and is prone to error [44]. The approximation we propose facilitates broader adoption of muscle models in Ansys by reducing implementation complexity; however, the convergence and accuracy of the approximation are not guaranteed for all materials and elements [45]. Consequently, we verified the approximation by comparing the stress and strain calculated using our Ansys implementation with analogous simulations in FEBio, which has the same built-in muscle material model and calculates the analytical value of the elasticity tensor. FEBio served as a benchmark for verification; the approximation of the muscle material model was

implemented in Ansys because the finite element model of the residual limb and transtibial prosthesis socket (described below) was to be developed in Ansys due to its popularity and superior functionality.

2. **Development of a constitutive model of skeletal muscle that considers force enhancement while improving the numerical stability of an existing model.** The muscle model we developed considers the effect of the fibers, including the passive and active tensile forces they generate. We performed numerical stability tests and used a 3D benchmark muscle model to validate the predicted stress. The 3D benchmark model was also used to validate the predicted fiber length distribution with experimental measurements reported in the literature. We implemented our muscle material model in Ansys and FEBio using the approximation described above (Contribution 1). We shared our subroutine code with the biomechanics community to improve reproducibility in the muscle finite element modelling literature, which is currently lacking. Making our muscle model freely available can benefit the scientific community by accelerating future research [46, 47].

3. **Development of a finite element model of the residual limb and transtibial prosthesis socket, considering the presence and activation of the muscle within the residuum.** We used a geometric model of the residual limb from the OpenLimb repository [48]. The liner and socket geometries were defined based on the external surface of the residual limb. We assigned constitutive models for the soft tissues using built-in material models in Ansys with parameters found in the literature [49]. The boundary conditions reflected the forces and moments experienced during walking at heel strike and toe off. We validated our model by comparing the distribution of stress over the residuum–socket interface with analogous results in the literature for a model with no muscle [49]. We then modified the residual limb geometry to include an approximation of the gastroc-

nemius muscle geometry. We used our new muscle model (Contribution 2) to model the behaviour of the gastrocnemius muscle tissue. Because the material model contains several parameters, each one carrying uncertainty, we performed a sensitivity analysis to explore how the variation of muscle material model parameters affects the stress on five regions of the residuum that are important for socket fit and comfort [50]. This study provides insight into the effect of muscle contraction on the pressure over the residuum–socket interface during gait.

1.4 Thesis organization

This thesis is organized into six chapters, which collectively address the research question of how muscle contraction affects the pressure over the residuum–socket interface during transtibial prosthesis–assisted gait. The remaining chapters are summarized as follows:

- **Chapter 2: Literature Review**

  This chapter reviews the literature on muscle biology, muscle material models, hyperelastic theory, finite element models of the residuum–socket interface, and shape and volume changes of the residual limb.

- **Chapter 3: Muscle Constitutive Model with a Tangent Modulus Approximation: Ansys Implementation and Verification**

  This chapter presents our Ansys implementation of an approximation of the tangent modulus applied to a muscle material model. The stress and strain from several test cases were compared to results from analogous simulations in FEBio, which has the same built-in muscle material model and calculates the analytical value of the elasticity tensor.

- **Chapter 4: Phenomenological Muscle Constitutive Model with Actin–Titin Binding for Simulating Active Stretching**

  This chapter presents our muscle constitutive model that exhibits force enhancement. The numerical stability of the model was tested with a simulation of a

unit cube model and the model was validated using a 3D benchmark model.

- **Chapter 5: Pressure Between Residual Limb and Transtibial Prosthesis Socket Considering Muscle Contraction: A Finite Element Study**

  This chapter presents our finite element model that considers the effect of muscle contraction on the residuum–socket interfacial pressure. The interfacial pressure was computed in several key regions of the residuum when the muscle was passive and when muscle contraction was considered.

- **Chapter 6: Conclusions and Future Work**

  The final chapter summarizes our key findings, contributions of the thesis, and suggestions for future work in muscle modelling, finite element analyses of the residuum–socket interface, and prosthesis design.

For convenience and because Chapters 3–5 comprise self-contained studies, a bibliography has been provided at the end of each chapter.

Bibliography for Chapter 1

[1] M. Marino et al. Access to prosthetic devices in developing countries: pathways and challenges. In: *Proceedings of the 5th IEEE Global Humanitarian Technology Conference*. 2015, pp. 45–51.

[2] Access Prosthetics. 15 limb loss statistics that may surprise you. Accessed: 2021-05-17. Oct. 2017. URL: https://accessprosthetics.com/15-limb-loss-statistics-may-surprise.

[3] S. Chatterjee et al. Review: problems with use of transtibial prosthesis. *Journal of Medical Imaging and Health Informatics*, 6, 269–284, 2016.

[4] OpenStax. OpenStax Anatomy and Physiology (Version 8.25). Accessed: 2025-01-27. May 2016. URL: https://openstax.org/books/anatomy-and-physiology/pages/7-1-divisions-of-the-skeletal-system.

[5] A.C. Donaghy et al. Team approach to prosthetic prescription decision-making. *Current Physical Medicine and Rehabilitation Reports*, 8, 386–395, 2020.

[6] A.F.T. Mak, M. Zhang, and D.A. Boone. State-of-the-art research in lower-limb prosthetic biomechanics-socket interface: a review. *Journal of Rehabilitation Research and Development*, 38, 161–173, 2001.

[7] C.J.C. Lamoth et al. Variability and stability analysis of walking of transfemoral amputees. *Medical Engineering and Physics*, 32, 1009–1014, 2010.

[8] J.T. Kahle et al. The effects of vacuum-assisted suspension on residual limb physiology, wound healing, and function: a systematic review. *Technology and Innovation*, 15, 333–341, 2014.

[9] BruceBlaus. An illustration depicting a foot prosthesis. Accessed: 2025-01-27. Apr. 2017. URL: https://commons.wikimedia.org/wiki/File:Foot_Prosthesis.png.

[10] W. Lovegreen et al. Lower Limb Amputation and Gait. In: *Braddom's Physical Medicine and Rehabilitation*. Elsevier, 2021, pp. 174–208.

[11] L. Paternò et al. Sockets for limb prostheses: a review of existing technologies and open challenges. *IEEE Transactions on Biomedical Engineering*, 65, 1996–2010, 2018.

[12] H.E. Meulenbelt et al. Determinants of skin problems of the stump in lower-limb amputees. *Archives of Physical Medicine and Rehabilitation*, 90, 74–81, 2009.

[13] L.E. Pezzin et al. Use and satisfaction with prosthetic limb devices and related services. *Archives of Physical Medicine and Rehabilitation*, 85, 723–729, 2004.

[14] J.E. Sanders, E.L. Rogers, and D.C. Abrahamson. Assessment of residual-limb volume change using bioimpedance. *Journal of Rehabilitation Research and Development*, 44, 525–535, 2007.

[15] R. Gailey et al. Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *Journal of Rehabilitation Research and Development*, 45, 15–29, 2008.

[16] K. Butler et al. A systematic review of the key factors affecting tissue viability and rehabilitation outcomes of the residual limb in lower extremity traumatic amputees. *Journal of Tissue Viability*, 23, 81–93, 2014.

[17] J. Kottner et al. Microclimate: A critical review in the context of pressure ulcer prevention. *Clinical Biomechanics*, 54, 62–70, 2018.

[18] C.W.J. Oomens et al. Pressure induced deep tissue injury explained. *Annals of Biomedical Engineering*, 43, 297–305, 2015.

[19] M.W Legro et al. Prosthesis evaluation questionnaire for persons with lower limb amputations: assessing prosthesis-related quality of life. *Archives of Physical Medicine and Rehabilitation*, 79(8), 931–938, 1998.

[20] J.E. Sanders and S. Fatone. Residual limb volume change: systematic review of measurement and management. *Journal of Rehabilitation Research and Development*, 48, 949–986, 2011.

[21] S.G. Zachariah et al. Shape and volume change in the transtibial residuum over the short term: preliminary investigation of six subjects. *Journal of Rehabilitation Research and Development*, 41, 683–694, 2004.

[22] R.T. Youngblood et al. Effects of activity intensity, time, and intermittent doffing on daily limb fluid volume change in people with transtibial amputation. *Prosthetics and Orthotics International*, 43, 28–38, 2019.

[23] M. Lilja, S. Johansson, and T. Öberg. Relaxed versus activated stump muscles during casting for trans-tibial prostheses. *Prosthetics and Orthotics International*, 23(1), 13–20, 1999.

[24] D. Solav et al. A framework for measuring the time-varying shape and full-field deformation of residual limbs using 3-D digital image correlation. *IEEE Transactions on Biomedical Engineering*, 66, 2740–2752, 2019.

[25] Y. Yurttas et al. A novel technique for transtibial amputation in chronic occlusive arterial disease: Modified Burgess procedure. *Prosthetics and Orthotics International*, 33, 25–32, 2009.

[26] C.C. Lyon et al. Skin disorders in amputees. *Journal of the American Academy of Dermatology*, 42(3), 501–507, 2000.

[27] A.S. Dickinson, J.W. Steer, and P.R. Worsley. Finite element analysis of the amputated lower limb: a systematic review and recommendations. *Medical Engineering and Physics*, 43, 1–18, 2017.

[28] S. Avril et al. Mixed experimental and numerical approach for characterizing the biomechanical response of the human leg under elastic compression. *Journal of Biomechanical Engineering*, 132(3), 031006, 2010.

[29] W. Zeng et al. Modeling of active skeletal muscles: a 3D continuum approach incorporating multiple muscle interactions. *Frontiers in Bioengineering and Biotechnology*, 11, 1153692, 2023.

[30] D.E. Rassier and W. Herzog. Force enhancement following an active stretch in skeletal muscle. *Journal of Electromyography and Kinesiology*, 12(6), 471–477, 2002.

[31] W. Herzog et al. A new paradigm for muscle contraction. *Frontiers in Physiology*, 6, 174, 2015.

[32] P. Meier and R. Blickhan. FEM-simulation of skeletal muscle: the influence of inertia during activation and deactivation. *Skeletal Muscle Mechanics: From Mechanisms to Function*, 12, 207–233, 2000.

[33] S.S. Blemker, P.M. Pinsky, and S.L. Delp. A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38, 657–665, 2005.

[34] A.E. Ehret, M. Böl, and M. Itskov. A continuum constitutive model for the active behaviour of skeletal muscle. *Journal of the Mechanics and Physics of Solids*, 59, 625–636, 2011.

[35] C. Kleinbach et al. Implementation and validation of the extended Hill-type muscle model with robust routing capabilities in LS-DYNA for active human body models. *Biomedical Engineering Online*, 16(109), 1–28, 2017.

[36] T.L. Allinger, M. Epstein, and W. Herzog. Stability of muscle fibers on the descending limb of the force-length relation. A theoretical consideration. *Journal of Biomechanics*, 29, 627–633, 1996.

[37] S.H. Yeo et al. Numerical instability of Hill-type muscle models. *Journal of the Royal Society Interface*, 20(199), 20220430, 2023.

[38] G. Chagnon, M. Rebouah, and D. Favier. Hyperelastic energy densities for soft biological tissues: a review. *Journal of Elasticity*, 120, 129–160, 2015.

[39] A.E. Ehret and M. Itskov. A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues. *Journal of Materials Science*, 42(21), 8853–8863, 2007.

[40] T. Jin, A. Chams, and X. Zhang. Impact of mathematical requirements on the invariant-based anisotropic constitutive models for non-linear biomaterials. *International Journal of Non-Linear Mechanics*, 147, 104188, 2022.

[41] J.C. Cagle et al. A finite element model to assess transtibial prosthetic sockets with elastomeric liners. *Medical and Biology Engineering Computing*, 56, 1227–1240, 2018.

[42] D. Lacroix and J.F. Ramírez Patiño. Finite element analysis of donning procedure of a prosthetic transfemoral socket. *Annals of Biomedical Engineering*, 39, 2972–2983, 2011.

[43] C. Miehe. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Computer Methods in Applied Mechanics and Engineering*, 134, 223–240, 1996.

[44] W. Sun, E.L. Chaikof, and M.E. Levenston. Numerical approximation of tangent moduli for finite element implementations of nonlinear hyperelastic material models. *Journal of Biomechanical Engineering*, 130(6), 061003, 2008.

[45] C. Suchocki. A finite element implementation of Knowles stored-energy function: theory, coding and applications. *Archive of Mechanical Engineering*, 58(3), 319–346, 2011.

[46] J.L. Hicks et al. Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement. *Journal of Biomechanical Engineering*, 137(2), 020905, 2015.

[47] T.T. Dao and M.-C.H.B. Tho. A systematic review of continuum modeling of skeletal muscles: current trends, limitations, and recommendations. *Applied Bionics and Biomechanics*, 2018(1), 7631818, 2018.

[48] F. Sunderland et al. OpenLimbTT: an open source transtibial residual limb model for simulation and design. *medRxiv*, 2024.

[49] J.W. Steer et al. Key considerations for finite element modelling of the residuum–prosthetic socket interface. *Prosthetics and Orthotics International*, 45(2), 138–146, 2021.

[50] M. Zhang and W.C.C. Lee. Quantifying the regional load-bearing ability of transtibial stumps. *Prosthetics and Orthotics International*, 30(1), 25–34, 2006.

2

Literature Review

This chapter contains a summary of topics that form the theoretical foundation for the development of our muscle constitutive model and finite element model of the residual limb considering muscle contraction. We begin by describing the biology of muscle and how muscle force is generated. Next, we describe the force enhancement behaviour that is observed when muscles are lengthened while activated. We then review the literature on muscle constitutive models, including methods to construct the muscle geometry and architecture. We describe the equations that are used to create the hyperelastic models in Ansys USERMAT subroutines and review the muscle material model on which our model is based. Finally, we review existing finite element models that simulate the residuum–prosthesis interface and motivate our investigation of how changes in the shape of the residuum affect the interfacial pressure.

2.1 Muscle biology

Muscles have a hierarchical structure. The force produced by a muscle originates at the molecular level through the interaction of two proteins: actin and myosin (Fig. 2.1). These proteins are arranged in subcellular structures called sarcomeres. The ends of

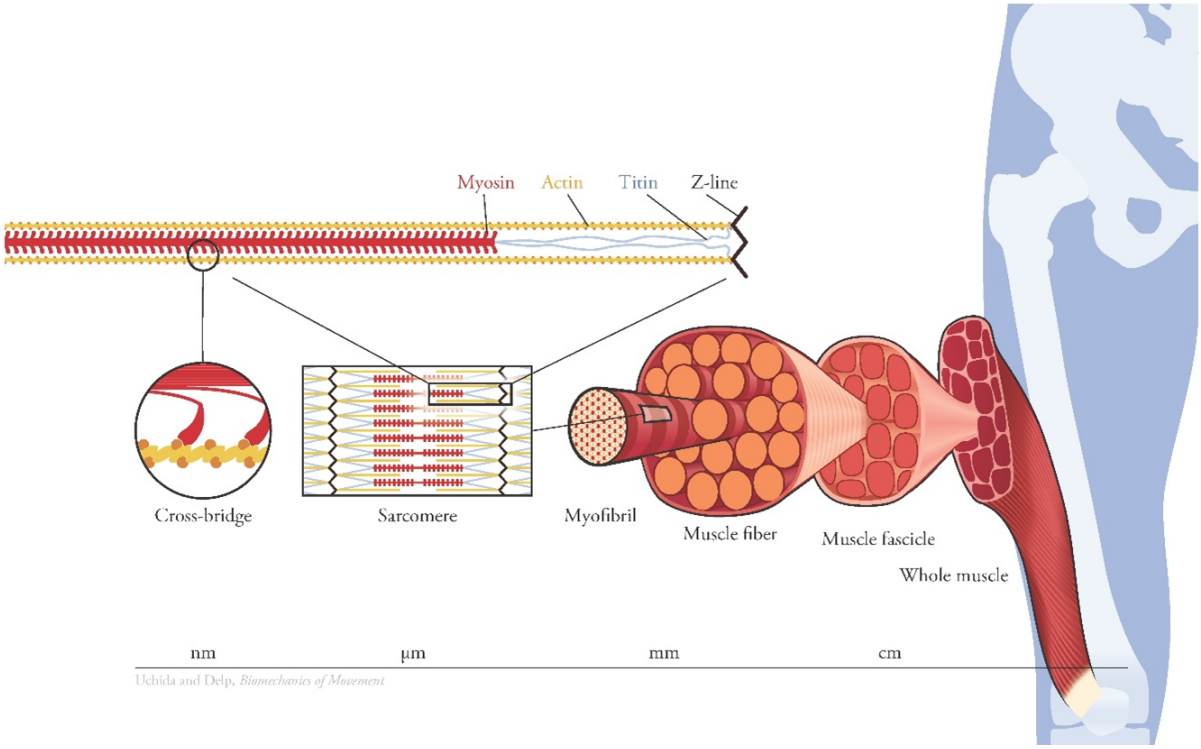

Figure 2.1: Muscle structure from the molecular level (actin and myosin forming cross-bridges) to the macro level (muscle fascicle). Image © 2020 David Delp [1].

each sarcomere are defined by rigid structures called Z-discs (which are often called Z-lines because they appear as lines in images). Another protein called titin attaches the ends of the myosin filament to the Z-discs at either end of the sarcomere. Sarcomeres are arranged in series to form strands called myofibrils, and myofibrils are arranged in parallel to form muscle fibers (muscle cells). Several fibers, along with a connective tissue, form the muscle fascicle. The connective tissue is an extracellular matrix composed of collagen, nerve fibers, and blood vessels. Several fascicles, along with more connective tissues, form the whole muscle [1]. Thus, muscles are composed primarily of sarcomeres arranged in series and in parallel. For this reason, when a muscle is recruited ("activated") by the nervous system to generate force, its overall response is similar to the behaviour produced at the sarcomere level but at a larger scale [2].

In the early 1950s, Hugh Huxley and Andrew Huxley discovered (independently) that the actin and myosin proteins have connections between them, which they called

cross-bridges [1]. They proposed a theory called the cross-bridge cycle to explain how muscles generate force through the coordination of these connections [3]. According to this theory, the myosin filament attaches and detaches cyclically at specific biding sites on the actin filament, forming cross-bridges. When the muscle is activated by the central nervous system, the myosin filaments pull on and translate along the actin filaments. This pulling action, and the consequent change in the length of the sarcomere, generates force in the muscle [4]. The force that is generated by a muscle when it is activated is called the active force.

The relationship between sarcomere length and the maximum active force that can be generated can be explained by the sliding filament theory. The theory proposes that, upon activation, the actin and myosin filaments slide past each other, which changes the amount of overlap between them. If there is a greater amount of overlap between the two filaments, more cross-bridges can be formed. Each cross-bridge contributes a small amount to the total active force generated by the muscle, so increasing the actin–myosin overlap increases the active force that can be generated [1].

The active force–length curve characterizes the relationship between sarcomere length and active force (Fig. 2.2). The active force–length curve is often described as comprising four regions, and its shape can be explained by noting the interaction between the actin and myosin filaments in a model of the sarcomere. Moving from left to right on this curve, the first region is the steep ascending limb, where tension increases dramatically as the sarcomere lengthens. In this region, the myosin filaments are compressed at the Z-discs, producing a force that opposes the force produced by the cross-bridge cycling. As the sarcomere lengthens in this region, the amount of myosin compression decreases and, therefore, so does the magnitude of this opposing force. Next, in the shallow ascending limb, we observe a (smaller) opposing force caused by the actin filaments from opposite ends of the sarcomere interfering with each other in the middle of the sarcomere. As the sarcomere lengthens in this region, the amount of interference decreases and this opposing force decreases. In the plateau region, the sarcomere is at or near the length

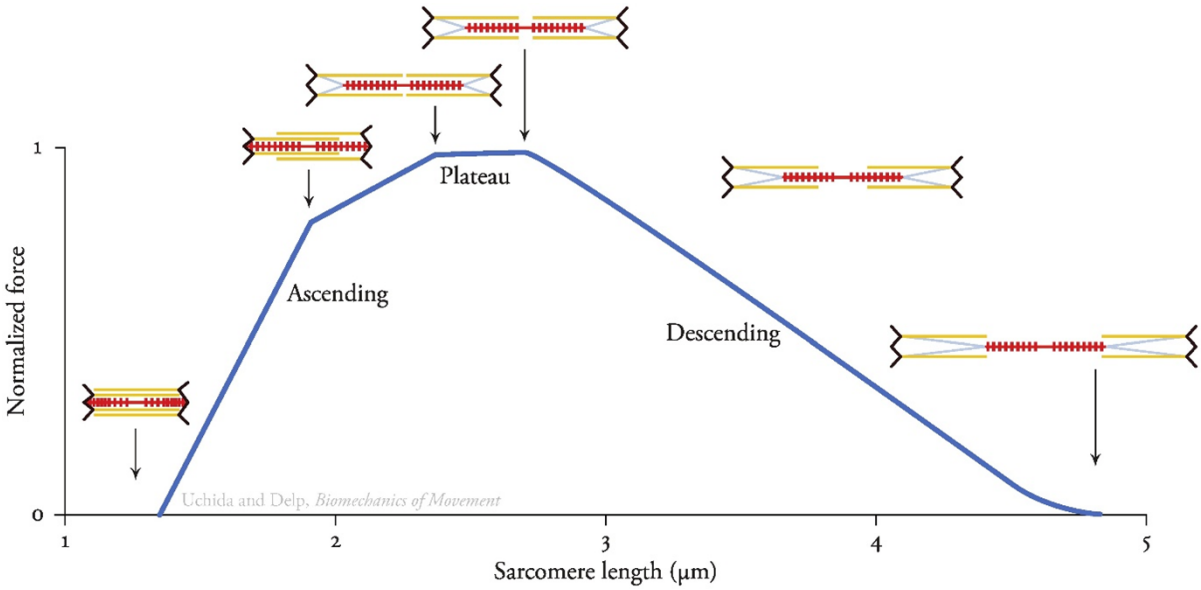

Figure 2.2: Sarcomere active force–length curve. The vertical axis represents the active force that is generated by the sarcomere when fully activated, normalized by the maximum isometric force that can be generated at any length. The horizontal axis represents the sarcomere length. Image © 2020 David Delp [1].

at which the greatest number of cross-bridges can be formed (its "optimal length") and, therefore, the greatest active force can be generated. The force that is generated with the sarcomere held at this length ("isometric") is called the maximum isometric force. As the sarcomere lengthens beyond the plateau region, the overlap between the actin and myosin decreases, thus decreasing the number of cross-bridges that can be formed and, consequently, decreasing the maximum active force that can be generated.

Muscles generate force when they are stretched beyond their optimal fiber length, even when they are not activated (i.e., when they are passive). This passive force–length relationship is due to the nonlinear-spring-like behaviour of the protein titin and the connective tissues surrounding fibers, fascicles, and whole muscle [1]. The total force generated by the muscle is the sum of the active and passive components (Fig. 2.3).

The speed at which a sarcomere is lengthening or shortening also affects the amount of active force that it can generate. Cross-bridges take time to form; therefore, relative to the case when the sarcomere is isometric, less force can be generated when the sarcomere

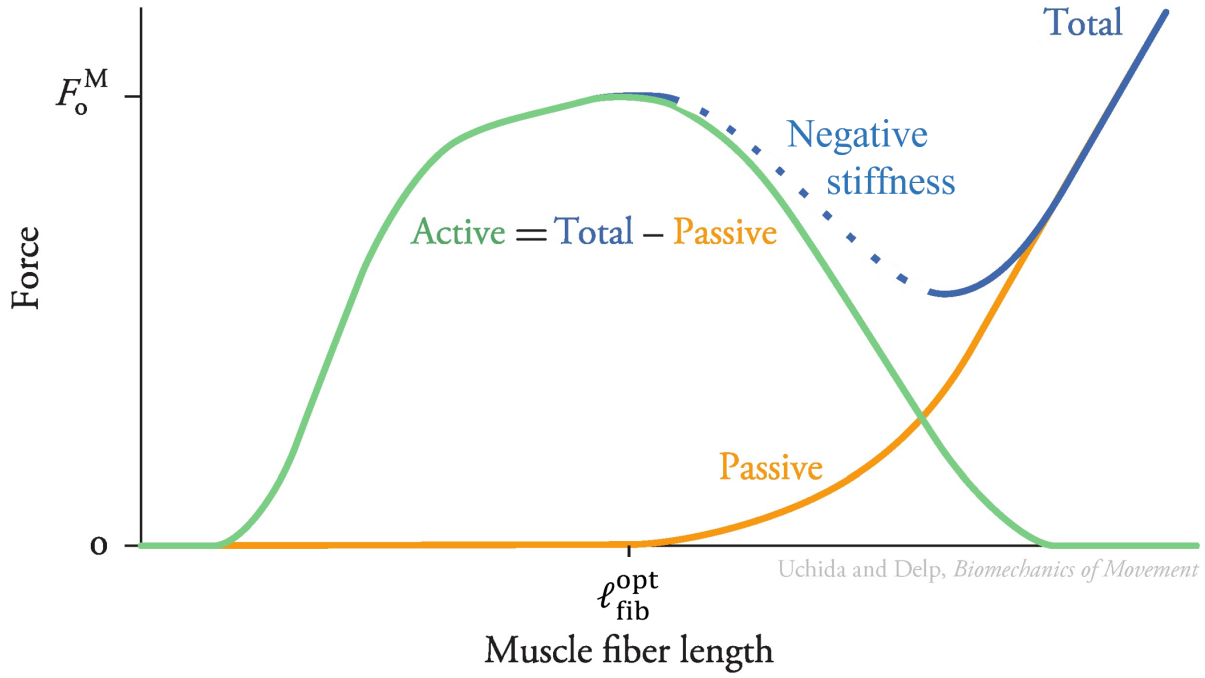

Figure 2.3: The total force–length relationship of muscle (blue) is the sum of the active (green) and passive (orange) components. The vertical axis represents the force produced by the muscle. The horizontal axis represents the muscle fiber length. F_o^M is the maximum isometric force, which occurs at the optimal muscle fiber length ($\ell_{\text{fib}}^{\text{opt}}$). Image adapted from Uchida and Delp [1]. Original image © 2020 David Delp.

is shortening while active (concentric contraction) and the actin and myosin filaments are sliding past each other quickly. On the other hand, active force is greater during active lengthening (eccentric contraction) than during isometric contraction because the myosin filaments are strained before they detach from the actin filaments [1].

Finally, the force generated by a muscle also depends on its activation α, which modulates the active force–length and force–velocity relationships. In models of muscle force generation, this modulation is achieved by simply multiplying the maximum active muscle force by the activation between 0 (passive) and 1 (fully active):

$$\text{Total Force} = (F^{\text{mus}}_{\text{active}}(\ell, v) \cdot \alpha) + F^{\text{mus}}_{\text{passive}}(\ell) \tag{2.1}$$

where $F^{\text{mus}}_{\text{active}}$ is the maximum active force, $F^{\text{mus}}_{\text{passive}}$ is the passive force, and ℓ and v are, respectively, the length and shortening velocity of the muscle.

2.2 Force enhancement during eccentric contraction

The cross-bridge theory proposed by Huxley [3] accurately predicts forces generated during isometric and concentric contraction [5]; however, the forces and energetic expenditure during eccentric contraction (i.e., when muscles are actively stretched) are not predicted accurately [6]. The forces produced during and after active stretching exceed the forces produced during an isometric contraction at the same muscle length [7]. The increase in force during active stretching is called force enhancement; after stretching, the increase in force persists and is then referred to as residual force enhancement [8]. Residual force enhancement has been observed at all muscle structural levels: in half-sarcomeres [9], in single isolated sarcomeres [10], in myofibrils [9], in single skinned fibers [11, 12], and in whole muscle [13]. Residual force enhancement is referred to as a history-dependent effect because the amount by which the force increases depends on the contractile history; however, this amount does not depend on the stretch velocity [14]. An active stretch involving a greater length change results in a greater residual

force enhancement [15].

For more than half a century, force enhancement was understood to be caused by non-uniformities in the lengths of the sarcomeres within a muscle [6, 16]. According to this theory, sarcomeres are unstable on the descending limb of the active force–length curve [17, 18] due to the negative stiffness of the total force–length curve (see Fig. 2.3). During active stretching of a myofibril on the descending limb of the active force–length curve, the hypothesis was that some sarcomeres will stretch more than others, resulting in non-uniform sarcomere lengths within the myofibril. If the average sarcomere length during this active stretch experiment is equal to the average length measured in an analogous isometric experiment (where all the sarcomeres are of equal length), the force produced during the active stretch experiment will be greater due to the contribution of the passive forces in the longer-than-average sarcomeres [18]. In contrast, during an isometric contraction on the descending limb, the sarcomeres remain at similar lengths and each sarcomere will generate the same passive force [5].

The non-uniform sarcomere theory is unable to explain some experimental observations. First, instability is not observed when stretching serially arranged sarcomeres on the descending limb of the active force–length curve [19]. Second, according to the non-uniform sarcomere theory, the produced force would not exceed the isometric force obtained in the plateau region of the active force–length curve. However, experiments performed in this region have shown that the forces produced during active stretching exceed those produced during isometric experiments by a substantial amount in whole muscle, single fibers, and isolated sarcomeres [10, 20, 21]. Finally, if force enhancement requires sarcomeres to be of non-uniform lengths, the phenomenon would not be observed in a single sarcomere. This conclusion contradicts the observations from the experiments performed by Leonard et al. [10], in which the force measured in a single sarcomere during active stretching exceeded the force measured during isometric contraction.

Over the last two decades, researchers have explored a new explanation for residual force enhancement associated with the titin filament [22]. The currently accepted theory

is that titin behaves as an elastic element whose stiffness changes when the muscle is activated due to two phenomena: (i) its material properties change, and (ii) its free length changes [6]. Regarding the material properties of titin, it was found that calcium ions bind to specific sites on the titin filament, increasing its stiffness. Labeit et al. [23] found that a particular segment of the titin filament called the PEVK segment (P-proline, E-glutamate, V-valine, and K-lysine) is a biding site for calcium ions. Regarding the free length of titin, Schappacher-Tilp et al. [24] hypothesized that the titin binds to the actin filament at a region of titin called the N2A region when calcium ions are present. This binding reduces the titin's free length, leaving only the PEVK segment "free" to lengthen during active stretching. Later, Dutta et al. [25] found experimental evidence that supported this hypothesis. Nishikawa et al. [26] suggested that the PEVK segment winds on the actin filament during cross-bridge formation. Rode et al. [27] proposed a "sticky-spring" mechanism in which the PEVK segment attaches to actin as a piece of adhesive tape would attach to a flat surface. Despite the experimental evidence and detailed phenomenological theories that have emerged, the mechanism responsible for force enhancement remains a topic of debate.

The effect of residual force enhancement on daily activities is a topic requiring further research [28]. Recent studies suggest that force enhancement is important to consider for optimizing muscle performance and preventing injuries [29]. During active stretching, force enhancement provides additional force at low energetic cost [6]. Seiberl et al. [29] have shown that less activation is required to sustain a given amount of force following active stretching, which reduces metabolic energy expenditure and improves neuromuscular efficiency. Furthermore, during active stretching, titin forces increase as the muscle stretches on the descending limb of the active force–length curve, thus providing a mechanism to prevent damage by avoiding overstretching [30].

2.3 Models of muscle force generation

Computational models of muscle force generation can be used to predict muscle behaviour in simulations. Muscle models have been used to expand our knowledge and understanding of muscle structure and function. For example, Blemker et al. [31] used a muscle model to understand how several features of biceps architecture (such as fiber direction and fascicle curvature) generate nonuniform strains in the muscle. As another example, Nazari et al. [32] employed musculoskeletal models to study muscle coordination during orofacial movement. Musculoskeletal models have also greatly contributed to our understanding of gait, sport performance, workplace ergonomics, and other human and animal movements [33].

Muscle models can be divided into two categories: phenomenological models and biophysical models. Phenomenological muscle models attempt to fit equations to experimental data without explaining the underlying aspects of muscle physiology. The one-dimensional "Hill-type" muscle model is one of the most well-known examples. Hill-type models approximate the behaviour of a muscle using three components: a contractile element in series with a nonlinear elastic spring element and in parallel with another nonlinear elastic element [34]. Biophysical muscle models represent the response of a muscle by considering its intrinsic physiological properties. The cross-bridge theory proposed by Huxley [35], described in Section 2.1, is an example of a biophysical model. Another example is the model developed by Röhrle et al. [36], which was the first muscle model based on the electrophysiological characteristics of muscle cells.

One-dimensional muscle models are relatively simple models that represent the mechanical behaviour and force generation of a muscle along a single line or curvilinear path. The benefit of these models is their low computational cost compared to 3D models, making them very useful for large musculoskeletal simulations that involve many muscles and joints. Typically, one-dimensional models cannot accurately represent real muscle geometry, fiber orientation, or the interaction between the muscle and the sur-

rounding tissues.

Three-dimensional models overcome the limitations of 1D models and can be used to model muscles comprised of fibers with unequal orientations, different material properties in different directions (anisotropy), and nonlinear material behaviour [37]. Creating a 3D model of a muscle requires first defining its geometry. Muscle geometry is typically built from magnetic resonance images (MRIs), using multiple image planes to construct a 3D representation. However, identifying each internal structure (called "segmentation") can be a difficult and time-consuming process as the boundaries of the internal structures (e.g., bones, muscles, tendons, and fat) may not be well defined in the images. This issue is a common source of error in segmentation algorithms. For instance, it can be very difficult to automatically segment the forearm muscles because they are surrounded by only a small layer of intermuscular fat [38]. Muscles that are surrounded by thick fatty tissues have a clearer boundary in MRI images, which makes segmentation easier.

Another important aspect of building a 3D model of a muscle is defining the muscle architecture. The architecture describes the arrangement of the fibers within the muscle, which has a direct effect on the direction of the contractile force and, thus, on active force production. Several authors have considered the fiber direction in their models at various levels of detail. Blemker and Delp [31] proposed a set of four simple templates using Bézier splines to create the fiber geometry analytically. The templates were applied to four muscle types that differed in the alignment of their fibers: parallel (fibers aligned with the tendon), pennate (fibers oriented at an acute angle relative to the tendon), curved (fibers have curved paths), and fanned (the muscle has a broad origin with fibers converging to the tendon) [39]. This technique is useful when the geometry of the biological fibers is unknown. Sánchez et al. [40] used the diffusion tensor image method to reproduce a more realistic fiber architecture. This method uses the correspondence between the fiber direction and the principal direction of the water diffusion inside the muscle cells [41]. Kinugasa et al. [42] used a mapping technique in which they first defined a reference fiber based on the orientation of the fascicle measured from ultrasound

images. The fiber direction for each element in a finite element model was then estimated as the tangent vector of an interpolated spline curve on the reference fiber.

Besides muscle geometry and architecture, muscle models require equations that compute the active and passive forces that are generated. Several constitutive models have been developed to simulate the passive behaviour of muscles. In the simpler models, muscle is considered to be a linearly elastic material [43]. More realistic models consider viscoplasticity behaviour by using the nonlinear Rubin–Bodner model [44]. To capture the nonlinear stress–strain behaviour of muscles, hyperelastic material models are often used. These models derive stress–strain relationships from the strain energy function, which represents the energy stored per unit volume of material as a function of deformation. The passive response has also been modelled using compressible hyperelastic materials based on the Neo-Hookean formulation [45, 46] and the Mooney–Rivlin model [40, 47]. When the active component of the muscle force is relevant to the research question, this component must also be considered in the model, but active force is generated in the fiber direction only. Therefore, muscles have been modelled as transversely isotropic materials, in which physical properties are symmetric about an axis that is normal to a plane of isotropy defined by the fiber architecture [31, 40, 48].

Many muscle models assume that the active and passive stresses are independent [31, 49, 50]. This practice is motivated by experiments in which muscles are tested when inactive (to measure the passive force) and active (to measure the total force). The active force is the difference between the forces measured during these active and inactive tests. Ehret et al. [48] developed a phenomenological, nearly incompressible, transverse isotropic model to consider the coupled active and passive behaviour for a muscle continuum model. Later, Weickenmeier et al. [51] examined the stress–strain curves derived from mathematical formulations of the strain energy function in a muscle model, investigating the effect of coupling versus decoupling the volumetric strain energy (representing dilatational changes) and the distortional strain energy (representing isochoric, volume-preserving changes). Their analysis was based on the constitutive model

proposed by Ehret et al. [48]. Weickenmeier et al. [51] concluded that the decoupled model is mathematically inconsistent.

Only a few muscle constitutive models consider the force enhancement effect. The first phenomenological constitutive model that attempted to consider force enhancement replaced the nonlinear active force–length curve with a linear relationship in order to avoid a region of negative stiffness in the total force–length curve [52] (see Fig. 2.3). This model did not suffer from instability on the descending limb of the active force–length curve, but it produced unrealistic behaviour. Seydewitz et al. [53] proposed a phenomenological model to simulate force enhancement that considers both velocity- and history-dependent effects; however, this model does not consider the length of the fiber at the onset of activation. Heidlauf et al. [54] proposed a model that exhibits force enhancement, but the forces produced by the actin–titin interaction were not validated.

2.4 Hyperelastic material equations

In the finite element method, hyperelastic theory describes the behaviour of materials undergoing large deformations whose stress–strain relationship is nonlinear. The theory is based on the Helmholtz free energy density function $\Psi(\mathbf{F}) = \Psi(\mathbf{C})$, also known as the strain-energy function, which is a scalar-valued function that relates the strain energy per unit volume to the deformation gradient ($\mathbf{F}$). $\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F}$ is the right Cauchy–Green tensor [55]. The deformation gradient ($\mathbf{F}$) is defined as the transformation from an incremental change in the referential configuration ($d\mathbf{X}$) to the corresponding incremental change in the current position ($d\mathbf{x}$):

$$\mathbf{F} = \frac{d\mathbf{x}}{d\mathbf{X}} = \mathbf{R}\mathbf{U} \tag{2.2}$$

where $\mathbf{R}$ is a rotation matrix describing the orientation of the local frame and $\mathbf{U}$ is the right stretch matrix, which measures the local stretching (or contraction) along the orthogonal eigenvectors of $\mathbf{U}$.

Rubber-like or hyperelastic materials have been used in a broad range of biomedical

and non-biomedical applications [56]. Creating models that can accurately predict the behaviour of these materials is important in computer-guided design. In the biomedical domain, hyperelastic material models have been used to emulate the behaviour of several soft tissues, including ligaments [57] and muscles [31]. Hyperelastic models are commonly used in commercial finite element software packages such as Abaqus and Ansys. These software packages have built-in hyperelastic material models, including Mooney–Rivlin, Yeoh, and Neo-Hookean models.

Some physical materials cannot be adequately represented using only the built-in material models that are available in the most popular finite element software packages. Soft tissues comprising collagen, elastin, muscle, and extracellular matrix are examples of materials that demand specialized models to simulate their nonlinear stress–strain behaviour. Abaqus and Ansys provide subroutines UMAT and USERMAT, respectively, for users to implement their own material models.

User-defined UMAT and USERMAT subroutines must be programmed to provide the Cauchy stress and the tangent modulus (also referred to as the elasticity tensor or material Jacobian) as outputs, given the deformation gradient and the material model parameters. The Cauchy stress is a second-order tensor that consists of nine elements (only six of which are independent due to symmetry) and defines the stress state at every integration point within the material element. The Cauchy stress ($\boldsymbol{\sigma}$) is calculated as follows for hyperelastic materials under large deformations:

$$\boldsymbol{\sigma} = 2J^{-1}\mathbf{F}\left(\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}}\right)\mathbf{F}^{\mathrm{T}} \tag{2.3}$$

where $J = \det(\mathbf{F})$, the determinant of $\mathbf{F}$. The tangent modulus ($\mathbb{C}$) is a fourth-order tensor for the incremental-iterative Newton–Raphson solution method that is required to solve the nonlinear equations used in the finite element method. The tangent modulus is calculated as follows:

$$\mathbb{C} = 4\frac{\partial^2 \Psi(\mathbf{C})}{\partial \mathbf{C} \partial \mathbf{C}} \tag{2.4}$$

The Cauchy stress (Eq. (2.3)) is relatively simple to calculate programmatically. Conversely, the tangent modulus (Eq. (2.4)) is complicated to compute: implementing this calculation is prone to error because the tensor is fourth order, which leads to cumbersome calculations. To overcome this issue, Miehe [58] proposed an approximation that uses perturbations of the deformation gradient to simplify the implementation without compromising the convergence rate [59]. Notably, because the approximation is used only by the Newton–Raphson iterative solver, the ultimate solution will be the same (within numerical tolerance) as if the exact tangent modulus were used.

The UMAT and USERMAT subroutines must be frame invariant, which means that the stress state and other physical quantities of the system must be invariant to a change of observer [55]. Abaqus and Ansys use an intermediate reference configuration called a "co-rotated frame" [60]. This frame is defined at each time step of a simulation and is rotated from the previous time step by performing multiplication operations with the rotation matrix $\mathbf{R}$ [55].

2.5 Muscle model of Blemker et al. (2005)

In this section, we describe the muscle constitutive model proposed by Blemker et al. [31], which will serve as a reference for the muscle constitutive model developed in this thesis. The energy related to the fiber shear resistance term in the model of Blemker et al. [31] was proposed by Criscione et al. [61]. The constitutive model proposed by Blemker et al. [31] exhibits a transversely isotropic response with invariants (values that are independent of the reference system) B_1 and B_2 that are associated with the interaction between the muscle fibers and the intramuscular connective tissue. Invariant B_1 accounts for the along-fiber shear while B_2 accounts for the cross-fiber shear [31]. B_1 and B_2 are functions of the well-known invariants $\bar{I}_1$, $\bar{I}_2$, $\bar{I}_4$, and $\bar{I}_5$ (functions of the modified right

Cauchy tensor, $\bar{\mathbf{C}} = J^{-2/3}\mathbf{C}$), and are defined as follows:

$$\bar{I}_1 = \mathrm{tr}(\bar{\mathbf{C}}) \tag{2.5}$$

$$\bar{I}_2 = \frac{1}{2}\left[\mathrm{tr}(\bar{\mathbf{C}})^2 - \mathrm{tr}(\bar{\mathbf{C}}^2)\right] \tag{2.6}$$

$$\bar{I}_4 = \mathbf{a_0} \cdot \bar{\mathbf{C}}\mathbf{a_0} \tag{2.7}$$

$$\bar{I}_5 = \mathbf{a_0} \cdot \bar{\mathbf{C}}^2\mathbf{a_0} \tag{2.8}$$

$$B_1 = \sqrt{\bar{I}_5/\bar{I}_4 - 1} \tag{2.9}$$

$$B_2 = \cosh^{-1}\left(\frac{\bar{I}_1\bar{I}_4 - \bar{I}_5}{2\sqrt{\bar{I}_4}}\right) \tag{2.10}$$

where $\mathrm{tr}(\bar{\mathbf{C}})$ denotes the trace of $\bar{\mathbf{C}}$ and $\mathbf{a_0}$ is a unit vector in the direction of the fiber.

When we combine the along- and cross-fiber shear strain energies (W_1 and W_2) with the muscle fiber stretch strain energy (W_3), the isochoric strain energy (Ψ_{iso}) can be written as follows:

$$\Psi_{\mathrm{iso}}\left(\bar{I}_1, \bar{I}_2, \bar{I}_4, \bar{I}_5, \alpha\right) = W_1\left(B_1\left(\bar{I}_4, \bar{I}_5\right)\right) + W_2\left(B_2\left(\bar{I}_1, \bar{I}_4, \bar{I}_5\right)\right) + W_3\left(\lambda, \alpha\right) \tag{2.11}$$

where

$$W_1 = G_1 B_1^2 \tag{2.12}$$

$$W_2 = G_2 B_2^2 \tag{2.13}$$

The coefficients G_1 and G_2 represent the moduli of B_1 and B_2, respectively. The energy term W_3 is associated with the Cauchy stress generated by the fiber stretch (σ_{fiber}). W_3 has two components: the passive stress and the active stress. The former depends on the fiber stretch (λ) while the latter depends on both fiber stretch and activation level (α), where $0 \leq \alpha \leq 1$. The Cauchy stress is related to the muscle fiber stretch strain energy as follows:

$$\lambda\frac{\partial W_3}{\partial \lambda} = \sigma_{\mathrm{fiber}}(\lambda, \alpha) \tag{2.14}$$

The following assumptions are made to simplify the calculation of the fiber stress [2]:

1. The maximum isometric force occurs at the optimal fiber length.
2. All sarcomeres in series have the same length.
3. The active fiber stress varies proportionally with activation (λ).
4. The sarcomeres are uniformly distributed within the muscle.
5. Incompressibility can be assumed because muscles are comprised predominantly of water. This assumption leads to the relation $A(\lambda_{\text{opt}})/A(\lambda) = \lambda/\lambda_{\text{opt}}$, where $A(\lambda)$ is the cross-sectional area of the fibers for a given stretch λ. The optimal fiber stretch (λ_{opt}) is the stretch (relative to its length at the beginning of the simulation) at which a fiber reaches its optimal length.
6. The passive fiber force (f_{passive}) and active fiber force (f_{active}) are independent, each being described by a force–length curve.

These simplifications lead to the following definition of fiber stress:

$$\sigma_{\text{fiber}}(\lambda, \alpha) = \sigma_{\max} \frac{\lambda}{\lambda_{\text{opt}}} \big(f_{\text{passive}}(\lambda) + \alpha f_{\text{active}}(\lambda)\big) \tag{2.15}$$

where $\sigma_{\max}$ is the maximum isometric stress, which is the maximum isometric force divided by the cross-sectional area of the fibers. The passive force (f_{passive}) is normalized by the maximum isometric force and is modelled as a piecewise-exponential function of the fiber stretch [31]:

$$f_{\text{passive}}(\lambda) = \begin{cases} 0, & \text{if } \lambda \leq \lambda_{\text{opt}} \\ P_1 \left(e^{P_2(\lambda/\lambda_{\text{opt}} - 1)} - 1\right), & \text{if } \lambda_{\text{opt}} < \lambda < \lambda^* \\ P_3 \lambda/\lambda_{\text{opt}} + P_4, & \text{if } \lambda \geq \lambda^* \end{cases} \tag{2.16}$$

where P_1, P_2, P_3, and P_4 are constants. Parameters P_3 and P_4 must be selected to guarantee C^0 and C^1 continuity at λ^* (i.e., the stretch above which the passive function is linear). Hence, parameters P_3 and P_4 can be written as functions of P_1 and P_2 [31]:

$$P_3 = P_1 P_2 e^{P_2(\lambda^*/\lambda_{\text{opt}}-1)} \tag{2.17}$$

$$P_4 = P_1 \left((1 - P_2\lambda^*/\lambda_{\text{opt}})e^{P_2(\lambda^*/\lambda_{\text{opt}}-1)} - 1\right) \tag{2.18}$$

The second term in the fiber stress equation (Eq. (2.15)) is the active force (f_{active}). This function is also normalized by the maximum isometric force and is defined as a piecewise-quadratic function of the fiber stretch [31]:

$$f_{\text{active}}(\lambda) = \begin{cases} 9\left(\lambda/\lambda_{\text{opt}} - 0.4\right)^2, & \text{if } \lambda \leq 0.6\lambda_{\text{opt}} \\ 1 - 4\left(1 - \lambda/\lambda_{\text{opt}}\right)^2, & \text{if } 0.6\lambda_{\text{opt}} < \lambda < 1.4\lambda_{\text{opt}} \\ 9\left(\lambda/\lambda_{\text{opt}} - 1.6\right)^2, & \text{if } \lambda \geq 1.4\lambda_{\text{opt}} \end{cases} \tag{2.19}$$

Constants P_1, P_2, P_3, and P_4 that appear in Eqs. (2.16)–(2.18), and the coefficients that appear in Eq. (2.19), were obtained from Blemker et al. [31] and were determined by fitting the stress–strain relationship for muscle to data from Zajac [2].

2.6 Finite element models for residuum and prosthesis mechanics

The finite element method is a powerful numerical tool that has been widely used in the biomedical field. In simulation studies of prostheses, the main applications of finite element models have been to study the mechanics of the soft tissue in the residuum, to study residuum–prosthesis interface mechanics, to identify residuum tissue characteristics, and to analyze prosthesis components [62]. This thesis focuses on the mechanics of the residuum–prosthesis interface; thus, the review of existing work will be concentrated on this application area.

As simulation technology has advanced, more complex finite element models of the residuum–socket interface have been developed. For example, existing studies have explored various modelling techniques [63], various loading conditions [64], donning loads [65, 66], quasi-dynamic analyses [67], hyperelastic material models that are capable of representing soft tissues [68], and various socket designs [69, 70]. A summary of existing finite element analyses of the residuum–socket interface is provided in Table 2.1.

Table 2.1: Summary of existing work on finite element analysis of the residuum–socket interface. A.L.=amputation level; CAD=computer-aided design; CT=computed tomography; E=Young's modulus; FE=finite element; MRI=magnetic resonance image; N/A=not applicable or not provided; PTB=patellar tendon bearing; TF=transfemoral; TSB=total surface bearing; TT=transtibial; v=Poisson's ratio.

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Zachariah and Sanders (2000)	Comparison of two contact elements on the interface stress	TT	Socket: CAD file; bones: CT scan	800 N axial load	Rigid	$E = 965$ kPa, $v = 0.45$	N/A	$E = 1$ GPa, $v = 0.35$	$\mu = 0.675$
Jia et al. (2004)	Influence of dynamic (inertial) loads on interface stress	TT	Soft tissue and bones: MRI; socket: CAD software	Quasi-dynamic knee forces and moment from gait cycle	$E = 10$ GPa, $v = 0.3$	$E = 200$ kPa, $v = 0.49$	$E = 380$ kPa, $v = 0.39$	Rigid	$\mu = 0.5$

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Lee et al. (2004)	Influence of pre-stress on the stress interface	TT	Soft tissue and bones: MRI; socket: CAD software	Quasi-dynamic knee forces and moment from gait cycle	$E = 10$ GPa, $v = 0.3$	$E = 200$ kPa, $v = 0.49$	N/A	$E = 1.5$ GPa, $v = 0.3$	$\mu = 0.5$
Faustini et al. (2006)	Analysis of the structure response of four PTB socket designs	TT	Soft tissue, bones and socket: CT scan	Initial load equals 800 N weight. Next, forces and moments from gait cycle.	$E = 15$ GPa, $v = 0.3$	$E = 200$ kPa, $v = 0.39$	$E = 380$ kPa, $v = 0.39$	$E = 1.6$ GPa $v = 0.39$	$\mu = 0.675$

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Portney et al. (2007)	Develop a patient-specific FE parametric biomechanical model of the residual limb	TT	N/A	400 N axial load applied on the knee	$E = 7.3$ GPa, $v = 0.3$	$E = 100$ kPa, $v = 0.49$	N/A	Rigid	$\mu = 0.5$
Frillici et al. (2008)	Discussion of the simulation tools for socket design	TT	External part of the limb: 3D laser scan; internal structure: MRI	Initial load equals half of the body weight. Next, forces and moments from gait cycle.	$E = 10$ GPa, $v = 0.24$	$E = 0.2$ GPa, $v = 0.9$	N/A	$E = 1.5$ GPa, $v = 0.9$	N/A

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Lacroix and Ramírez Patiño (2011)	Time-dependent model of pre-stress from donning process	TF	Soft tissue and bones: CT scan; socket: scan of rectified cast	Donning axially along residuum at 6–9 mm/s velocity	$E = 15$ GPa, $v = 0.3$	3-parameter hyperelastic Mooney–Rivlin, $C_{21} = 4.25$ kPa, $C_{11} = 0$, $D_1 = 2.36$ MPa	N/A	$E = 1.5$ GPa, $v = 0.3$	$\mu = 0.415$
Zhang et al. (2013)	Prediction of stress on translational socket–limb interface	TF	Soft tissue, bones, and socket: CT scan	50 N pre-loading (donning), quasi-dynamic knee forces and moment from gait cycle	$E = 15$ GPa, $v = 0.3$	2-parameter hyperelastic Mooney–Rivlin, $C_{21} = 85.5$ kPa, $C_{12} = 21.4$ kPa, $v = 0.459$	N/A	$E = 15$ GPa, $v = 0.3$	$\mu = 0.5$

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Van et al. (2017)	Investigation of different socket types considering hyperplasticity	TF	Soft tissue and bones: MRI	Half and full body weight	$E =$ 17.7 GPa, $v = 0.3$	Quasi-linear viscoelastic with 3-parameter Mooney–Rivlin model; different parameters were defined for skin, fat and muscle	N/A	$E =$ 1.885 GPa, $v = 0.39$	$\mu = 0.5$
Cagle et al. (2018)	Investigation of TSB socket–limb stress interface considering skin issues as validation	TT	Soft tissue, bones and socket: MRI	Two loading profiles representing two clinical conditions	N/A	$E = 300$ kPa, $v = 0.45$	3-parameter hyperelastic Yeoh, $C_{10} = 20.14$ kPa, $C_{30} = -1.54$ kPa, $C_{50} = 0.41$ kPa	$E = 19$ GPa, $v = 0.1$	$\mu = 0.5$

Authors	**Objective**	**A.L.**	**Source**	**Loading Scenario**	**Bone Properties**	**Soft-Tissue Properties**	**Liner Properties**	**Socket Properties**	**Friction Interface**
Jamaludin et al. (2019)	Analysis of pressure between two types of sockets and a patient-specific limb model	TF	Soft tissue, bones and socket: MRI	Donning axially along residuum at 0.5 mm/s velocity after a half-body weight was applied	$E =$ 17.7 GPa, $v = 0.3$	Quasi-linear viscoelastic with Mooney–Rivlin model; different parameters were defined for skin, fat and muscle	N/A	$E =$ 1.885 GPa, $v = 0.39$	$\mu = 0.5$
Steer et al. (2021)	Best practice in FE modelling of lower-limb prostheses	TT	Soft tissue and bones: MRI	Donning by overclosure and gait loads at three instants	$E = 12$ GPa, $v = 0.3$	Neo-Hookean and linear for different regions on the limb	Neo-Hookean, $C = 37.6$ kPa, $D = 0.54$ MPa^{-1}	$E = 1.5$ GPa, $v = 0.3$	$\mu = 0.5$

The residual limb and prosthesis socket have complex geometries that are unique to each individual and may be customized in simulations. Custom geometries can be created from computed tomography (CT) scans and MRI. Cagle et al. [71] proposed two steps to define the geometry of a residual limb and prosthesis socket from MRI images. Initially, a sequence of MRI images of the limb is imported into a parametric modelling software package to be segmented with bounding splines. The spline profiles are then imported into a 3D scan data processor to be "lofted" (an operation that creates a 3D object by connecting stacked 2D profiles) and smoothed. Some studies have reported that the liner inserted between the residual limb and the socket is not visible in MRI images [67, 71, 72]. To overcome this issue, many authors have defined the liner geometry by creating a 4–6 mm offset from the residual limb. Once the geometry has been created for all parts of the model, one must define the mesh elements. For 3D models, the following elements have been used: tetrahedral [64, 72], 8-node hexahedral [66], and 10-node hexahedral [69].

Each element of a finite element model is assigned a material model according to its required mechanical response. Bones, soft tissues, the prosthesis socket, and the liner are the typical parts in a finite element model for studying the residuum–prosthesis interface. Bones and sockets are treated as linear elastic materials with high elastic modulus (Young's modulus, E). The elastic modulus for bones typically ranges from 10 GPa to 20 GPa; for sockets, an elastic modulus between 1.5 GPa and 20 GPa has been used [69–71]. Rather than create the socket structure explicitly, Lin et al. [72] fixed the exterior nodes of the liner to ground to approximate the effect of the socket. Cagle et al. [71] used a similar method to simulate the bones by fixing the internal nodes of the soft tissues to ground. Liners have generally been modelled as linear elastic materials, but with lower stiffness than bones and sockets (see Table 2.1). The typical range of elastic modulus used to represent the liner is 0.4–0.8 MPa [67, 69].

Soft tissues in the residual limb exhibit nonlinear stress–strain behaviour. Recent studies have applied hyperelastic materials to model these soft tissues [64, 66, 71]. The

strain energy models used by these authors were the Mooney–Rivlin model [64, 66, 70] and the Yeoh model [71]. Although hyperelastic models are capable of reproducing the nonlinear stress–strain behaviour of soft tissues, they require more computation and can cause convergence issues during simulations. For example, Cagle et al. [71] reported instability when they used the Mooney–Rivlin model in their simulations. Ultimately, they decided to use a linear elastic model for the soft tissues in their model to reduce the computational burden and improve convergence. The approach of considering soft tissues as linear elastic materials was also adopted by Lin et al. [72], Jia et al. [67], and Faustini et al. [69].

The appropriate boundary conditions in a finite element simulation depend on the type of virtual experiment that is being simulated. For example, previous studies have simulated loads during walking using boundary conditions and loads during standing by applying a force equivalent to one-half of the body weight. Lin et al. [72] simulated the body weight using a set of downward displacements on the superior face of the bone structure rather than using an applied load. Faustini et al. [69] applied an initial load on the knee (to simulate the standing load) and then used data from ground reaction force measurements at several instants of the stance phase: at 0% (heel strike), 15%, 35%, 55%, 78%, and 100% (toe off). Jamaludin et al. [70] applied an external load to the superior face of the residuum as a boundary condition to simulate the standing load.

Simulations must be verified and validated to ensure that they represent the behaviour of the system to sufficient accuracy in order to answer the given research question [73]. Verification tests ensure that equations have been implemented correctly (e.g., energy is conserved) while validation tests ensure that the simulations relate to reality (e.g., by comparing predictions to experimental observations). Finite element models must be calibrated by performing a convergence analysis to ensure the mesh is sufficiently dense. Finite element simulations can be verified by performing a variety of plausibility checks on the results [74]. Validation can be performed by comparing the predicted stress profiles from various regions of the residual limb with the stress profiles measured from

the corresponding regions of a real limb [72]. Another validation strategy is to compare the simulation results with previous simulation results reported in the literature [64]. Finally, Cagle et al. [71] validated their simulations by noting similarities between the regions on their simulated residual limb in which high stresses were predicted and the regions on a patient's limb where the skin was damaged.

2.7 Variation of volume and shape of the residuum

Sockets of commercial prostheses are typically designed assuming that the volume and shape of the residuum are constant [75]. Changes in the residuum volume affect the fit of the socket, consequently altering the stress-state on the residuum–socket interface. The residuum volume and shape change over time due to several factors, including postoperative liquid retention, muscle atrophy, fluid collection, fluid motion, and muscle contraction [76].

The eighteen-month period following an amputation surgery is considered the surgical recovery period. After the operation, the residuum undergoes considerable changes in shape and volume due to fluid retention immediately following the surgery; later changes occur during the recovery period as a result of muscle atrophy and fluid collection [77]. During this latter period, it is frequently necessary to change and reshape the socket because of the rapid reduction in the residual limb volume [76]. Beyond 160 days post-surgery, a volume reduction of up to 35% has been observed [78].

Even after the recovery period, daily volume changes are observed in mature residual limbs. These changes are caused by fluids moving into and out of the limb [79]. The volume changes due to fluid motion are caused by the pressure exerted on the residuum by the socket. When the pressure applied to the external surface of the residuum is greater than the internal pressure, the fluid inside the cells is pushed back into the bloodstream and the residuum volume decreases. On the other hand, when the residuum experiences a lower external pressure than its internal pressure, its volume tends to increase [80]. Because the residuum experiences an external pressure that is greater

than its internal pressure when a prosthesis is worn, the residuum volume decreases on average throughout the day [81]. Furthermore, muscle contraction causes the shape of the residuum to fluctuate during daily activities as the muscles bulge and their cross-sectional areas change [75, 82].

Experimental studies have demonstrated that the volume and shape of the residuum increase after walking. Lilja et al. [75] compared the residual limb volume when the muscles were activated with the volume of the limb when the muscles were relaxed. They considered two cases: when the participants had been wearing a liner and when a liner had not been worn. First, the relaxed residuum volume was measured using a laser scanner. The residuum was then scanned again while the participants maximally activated their gastrocnemius and tibialis anterior muscles. The average volume of the limb with activated muscles increased by 3.5% with a liner and 5.8% without a liner, when compared to the volume of the relaxed residuum. Zachariah et al. [79] reported a volume increase of 2.4–10.9% after a short period of walking (4 to 5 minutes). Sanders et al. [83] quantified the volume change observed in healthy individuals and those with comorbidities. Healthy participants presented a small volume increase after walking (0.2% to 0.9%), while a volume decrease was observed in individuals with comorbidities (–3.5% to –0.2%). Youngblood et al. [81] compared the volume change during walking, which the authors classified as a high-intensity activity, versus standing and sitting, which were classified as low-intensity activities. Their measurements revealed a smaller decrease in volume during high-intensity activity than during low-intensity activity. This difference is likely due to greater vascular flow into the muscle during high-intensity activity.

Several techniques have been proposed to measure the volume of the residual limb, including the following:

1. Water Displacement: The residuum is placed inside a cylinder completely filled with water. The mass of the displaced water is measured to calculate the volume of the residuum [84].

2. Spiral X-Ray Computed Tomography (SXCT): X-ray images are taken of the residual limb from several angles to measure the internal and external structure [85].

3. Laser Scanning: The residual limb is exposed to light. A camera is used to measure the shape created by the intersection of the light and the residuum from different angles. These measurements are used to create a computational model of the limb geometry [86].

4. Magnetic Resonance Imaging (MRI): The volume of the residual limb is calculated using commercial MRI equipment. This technique allows one to create a 3D representation of the residuum [87].

5. Bioimpedance: This method uses electrode pairs on the residual limb to apply current over a range of frequencies and measure the electrical resistance (impedance). The measured "bioimpedance" is related to the volume of extracellular and intracellular fluid. Mathematical models are then used to relate the measured resistance to a change in fluid volume. This method demonstrated good reproducibility and sensitivity [83].

Each of the techniques listed above has advantages and disadvantages. Measurement methods that involve applying force or pressure to the limb can distort the limb shape; the water displacement method is the most susceptible to this type of error. Immediately after doffing (taking off) a prosthesis, the volume of the residual limb might change substantially; the water displacement, laser scanning, and MRI methods are all susceptible to the distortion in shape of the residuum. For example, Zachariah et al. [79] reported a change in volume of between 0.2% and 17% measured 15 minutes after doffing the prosthesis. Furthermore, if the scanning equipment takes a long time to collect the measurements, the participant will tend to move, which reduces the accuracy of the measurement [87]. This issue is critical because even small movements can induce large measurement errors [76]. Finally, experimental instruments must be accurate and

provide good resolution. Accuracy is difficult to assess because of the lack of a "gold standard" test system when new instruments and measurement techniques are being developed. Therefore, some authors instead focus on the repeatability of a measurement strategy as a way to assess its accuracy [76].

Bibliography for Chapter 2

[1] T.K. Uchida and S.L. Delp. Biomechanics of Movement: The Science of Sports, Robotics, and Rehabilitation. Cambridge, MA: MIT Press, 2021.

[2] F.E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17, 359–411, 1989.

[3] A.F. Huxley. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7, 255–318, 1957.

[4] M.D. Binder, N. Hirokawa, and U. Windhorst. Cross-bridge theory. In: *Encyclopedia of Neuroscience*. Springer Berlin Heidelberg, 2009.

[5] A.M. Gordon, A.F. Huxley, and F.J. Julian. The variation in isometric tension with sarcomere length in vertebrate muscle fibres. *Journal of Physiology*, 184, 170–192, 1966.

[6] W. Herzog et al. A new paradigm for muscle contraction. *Frontiers in Physiology*, 6, 174, 2015.

[7] B.C. Abbott and X.M. Aubert. The force exerted by active striated muscle during and after change of length. *Journal of Physiology*, 117, 77–86, 1952.

[8] K.A.P. Edman, G. Elzinga, and M.I.M. Noble. Residual force enhancement after stretch of contracting frog single muscle fibers. *Journal of General Physiology*, 80, 769–784, 1982.

[9] V. Joumaa, T.R. Leonard, and W. Herzog. Residual force enhancement in myofibrils and sarcomeres. *Proceedings of the Royal Society B: Biological Sciences*, 275, 1411–1419, 2008.

[10] T.R. Leonard, M. DuVall, and W. Herzog. Force enhancement following stretch in a single sarcomere. *American Journal of Physiology Cell Physiology*, 299, 1398–1401, 2010.

[11] D.E. Rassier et al. Stretch-induced, steady-state force enhancement in single skeletal muscle fibers exceeds the isometric force at optimal fibre length. *Journal of Biomechanics*, 36, 1309–1316, 2003.

[12] V. Joumaa and W. Herzog. Energy cost of force production is reduced after active stretch in skinned muscle fibres. *Journal of Biomechanics*, 46, 1135–1139, 2013.

[13] A.E. Oskouei and W. Herzog. Observations on force enhancement in sub maximal voluntary contractions of human adductor pollicis muscle. *Journal of Applied Physiology*, 98, 2087–2095, 2005.

[14] H.D. Lee and W. Herzog. Force enhancement following muscle stretch of electrically and voluntarily activated human adductor pollicis. *The Journal of Physiology*, 545, 321–330, 2002.

[15] W. Herzog and T.R. Leonard. Force enhancement following stretching of skeletal muscle: a new mechanism. *The Journal of Experimental Biology*, 205, 1275–1283, 2002.

[16] D.L. Morgan. New insights into the behavior of muscle during active lengthening. *Biophysical Journal*, 57, 209–221, 1990.

[17] A.V. Hill. The mechanics of active muscle. *Proceedings of the Royal Society B: Biological Sciences*, 141, 104–117, 1953.

[18] T.L. Allinger, M. Epstein, and W. Herzog. Stability of muscle fibers on the descending limb of the force-length relation. A theoretical consideration. *Journal of Biomechanics*, 29, 627–633, 1996.

[19] D.E. Rassier, W. Herzog, and G.H. Pollack. Stretch-induced force enhancement and stability of skeletal muscle myofibrils, 538, 501–515, 2003.

[20] K.A.P. Edman, G. Elzinga, and M.I.M. Noble. Enhancement of mechanical performance by stretch during tetanic contractions of vertebrate skeletal muscle fibres. *The Journal of Physiology*, 281, 139–155, 1978.

[21] R. Schachar, W. Herzog, and T.R. Leonard. The effects of muscle stretching and shortening on isometric forces on the descending limb of the force-length relationship. *Journal of Biomechanics*, 37, 917–926, 2004.

[22] W. Herzog, E.J. Lee, and D.E. Rassier. Residual force enhancement in skeletal muscle. *The Journal of Physiology*, 574, 635–642, 2006.

[23] D. Labeit et al. Calcium-dependent molecular spring elements in the giant protein titin. *Proceedings of the National Academy of Sciences of the United States of America*, 100, 13716–13721, 2003.

[24] G. Schappacher-Tilp et al. A novel three-filament model of force generation in eccentric contraction of skeletal muscles. *PLoS One*, 10, e0117634, 2015.

[25] S. Dutta et al. Calcium increases titin N2A binding to F-actin and regulated thin filaments. *Scientific Reports*, 8, 14575, 2018.

[26] K.C. Nishikawa et al. Is titin a 'winding filament'? A new twist on muscle contraction. *Proceedings of the Royal Society B: Biological Sciences*, 279, 981–990, 2012.

[27] C. Rode, T. Siebert, and R. Blickhan. Titin-induced force enhancement and force depression. *Journal of Theoretical Biology*, 259, 350–360, 2009.

[28] W. Seiberl, G.A. Power, and D. Hahn. Residual force enhancement in humans: Current evidence and unresolved issues. *Journal of Electromyography and Kinesiology*, 25, 571–580, 2015.

[29] W. Seiberl et al. On the relevance of residual force enhancement for everyday human movement. *Journal of Biomechanics*, 46, 1996–2001, 2013.

[30] A.L. Hessel, S.L. Lindstedt, and K.C. Nishikawa. Physiological mechanisms of eccentric contraction and its applications: A role for the giant titin protein. *Frontiers in Physiology*, 8, 70, 2017.

[31] S.S. Blemker, P.M. Pinsky, and S.L. Delp. A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38, 657–665, 2005.

[32] M.A. Nazari, P. Perrier, and Y. Payan. The distributed lambda (λ) model (DLM): a 3-D, finite-element muscle model based on Feldman's λ model; assessment of orofacial gestures. *Journal of Speech, Language, and Hearing Research*, 56(6), 1909–1923, 2013.

[33] S.D. Uhlrich et al. Ten steps to becoming a musculoskeletal simulation expert: a half-century of progress and outlook for the future. *Journal of Biomechanics*, 154, 111623, 2023.

[34] S. Göktepe, A. Menzel, and E. Kuhl. The generalized Hill model: A kinematic approach towards active muscle contraction. *Journal of the Mechanics and Physics of Solids*, 72, 20–39, 2014.

[35] A.F. Huxley and R. Niedergerke. Structural changes in muscle during contraction. Interference microscopy of living muscle fibres. *Nature*, 173, 971–973, 1954.

[36] O. Röhrle, J.B. Davidson, and A.J. Pullan. Bridging scales: a three-dimensional electromechanical finite element model of skeletal muscle. *SIAM Journal on Scientific Computing*, 30, 2882–2904, 2008.

[37] T. Siebert and C. Rode. Computational modeling of muscle biomechanics. In: *Computational Modelling of Biomechanics and Biotribology in the Musculoskeletal System: Biomaterials and Tissues*. Woodhead Publishing Limited, 2014.

[38] S.S. Blemker et al. Image-based musculoskeletal modeling: applications, advances, and future opportunities. *Journal of Magnetic Resonance Imaging*, 25, 441–451, 2007.

[39] S.S. Blemker and S.L. Delp. Three-dimensional representation of complex muscle architectures and geometries. *Annals of Biomedical Engineering*, 33(5), 661–673, 2005.

[40] C.A. Sánchez et al. Embedding digitized fibre fields in finite element models of muscles. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization*, 2, 223–236, 2014.

[41] B.M. Damon et al. Validation of diffusion tensor MRI-based muscle fiber tracking. *Magnetic Resonance in Medicine*, 48, 97–104, 2002.

[42] R. Kinugasa et al. Influence of intramuscular fiber orientation on the Achilles tendon curvature using three-dimensional finite element modeling of contracting skeletal muscle. *Journal of Biomechanics*, 49, 3592–3595, 2016.

[43] L. Beldie et al. Finite element modelling of maxillofacial surgery and facial expressions – a preliminary study. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 6, 422–430, 2010.

[44] M. Jabareen and J. Trzewik. Development and validation of a three-dimensional finite element model of the face. *Journal of Biomedical Engineering*, 131(4), 041006, 2009.

[45] T. Wu, A. Hung, and K. Mithraratne. Generating facial expressions using an anatomically accurate biomechanical model. *IEEE Transactions on Visualization and Computer Graphics*, 20, 1519–1529, 2014.

[46] A.M. Zöllner et al. On high heels and short muscles: a multiscale model for sarcomere loss in the gastrocnemius muscle. *Journal of Theoretical Biology*, 365, 301–310, 2014.

[47] Y. Wang et al. Finite element analysis of the deformation of deep veins in the lower limb under external compression. *Medical Engineering and Physics*, 35, 515–523, 2013.

[48] A.E. Ehret, M. Böl, and M. Itskov. A continuum constitutive model for the active behaviour of skeletal muscle. *Journal of the Mechanics and Physics of Solids*, 59(3), 625–636, 2011.

[49] M. Böl and S. Reese. Micromechanical modelling of skeletal muscles based on the finite element method. *Computational Methods in Biomechanics and Biomedical Engineering*, 11, 489–504, 2008.

[50] J. Li et al. Development of a finite element musculoskeletal model with the ability to predict contractions of three-dimensional muscles. *Journal of Biomechanics*, 94, 230–234, 2019.

[51] J. Weickenmeier et al. A physically motivated constitutive model for 3D numerical simulation of skeletal muscles. *International Journal for Numerical Methods in Biomedical Engineering*, 30, 545–562, 2014.

[52] R. Lemos et al. Realistic skeletal muscle deformation using finite element analysis. In: *Proceedings XIV Brazilian Symposium on Computer Graphics and Image Processing*. Florianopolis, Brazil, 2001, pp. 192–199.

[53] R. Seydewitz, T. Siebert, and M. Böl. On a three-dimensional constitutive model for history effects in skeletal muscles. *Biomechanics Model in Mechanobiology*, 18, 1665–1681, 2019.

[54] T. Heidlauf et al. A multi-scale continuum model of skeletal muscle mechanics predicting force enhancement based on actin-titin interaction. *Biomechanics Model in Mechanobiology*, 15, 1423–1437, 2016.

[55] G.A. Holzapfel. Nonlinear Solid Mechanics: A Continuum Approach for Engineering. Chichester: John Wiley & Sons, Ltd, 2001.

[56] P. Steinmann, M. Hossain, and G. Possart. Hyperelastic models for rubber-like materials: consistent tangent operators and suitability for Treloar's data. *Archive of Applied Mechanics*, 82, 1183–1217, 2012.

[57] E. Peña et al. An anisotropic visco-hyperelastic model for ligaments at finite strains. Formulation and computational aspects. *International Journal of Solids and Structures*, 44, 760–778, 2007.

[58] C. Miehe. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Computer Methods in Applied Mechanics and Engineering*, 134, 223–240, 1996.

[59] H. Liu and W. Sun. Numerical approximation of elasticity tensor associated with Green–Naghdi rate. *Journal of Biomechanical Engineering*, 139, 139–148, 2017.

[60] ANSYS, Inc. ANSYS Mechanical APDL Technology Demonstration Guide, Release 15.0. Tech. rep. Canonsburg, PA, USA, 2013.

[61] J.C. Criscione, A.S. Douglas, and W.C. Hunter. Physically based strain invariant set for materials exhibiting transversely isotropic behavior. *Journal of the Mechanics and Physics of Solids*, 49, 871–897, 2001.

[62] A.S. Dickinson, J.W. Steer, and P.R. Worsley. Finite element analysis of the amputated lower limb: A systematic review and recommendations. *Medical Engineering and Physics*, 43, 1–18, 2017.

[63] S.G. Zachariah and J.E. Sanders. Finite element estimates of interface stress in the transtibial prosthesis using gap elements are different from those using automated contact. *Journal of Biomechanics*, 33, 895–899, 2000.

[64] L. Zhang et al. Finite element analysis of the contact interface between transfemoral stump and prosthetic socket. In: *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. 2013, pp. 1270–1273.

[65] W.C.C. Lee et al. Finite element modeling of the contact interface between transtibial residual limb and prosthetic socket. *Medical Engineering and Physics*, 26, 655–662, 2004.

[66] D. Lacroix and J.F. Ramírez Patiño. Finite element analysis of donning procedure of a prosthetic transfemoral socket. *Annals of Biomedical Engineering*, 39, 2972–2983, 2011.

[67] X. Jia et al. A quasi-dynamic nonlinear finite element model to investigate prosthetic interface stresses during walking for transtibial amputees. *Clinical Biomechanics*, 20, 630–635, 2005.

[68] F.S. Frillici et al. The role of simulation tools to innovate the prosthesis socket design process. In: *Innovative Production Machines and Systems*. 2008, pp. 214–219.

[69] M.C. Faustini, R.R. Neptune, and R.H. Crawford. The quasi-static response of compliant prosthetic sockets for transtibial amputees using finite element methods. *Medical Engineering and Physics*, 28, 114–121, 2006.

[70] M.S. Jamaludin et al. Analysis of pressure distribution in transfemoral prosthetic socket for prefabrication evaluation via the finite element method. *Bioengineering*, 6, 98, 2019.

[71] J.C. Cagle et al. A finite element model to assess transtibial prosthetic sockets with elastomeric liners. *Medical and Biology Engineering Computing*, 56, 1227–1240, 2018.

[72] C.C. Lin et al. Effects of liner stiffness for transtibial prosthesis: a finite element contact model. *Medical Engineering and Physics*, 26, 1–9, 2004.

[73] J.L. Hicks et al. Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement. *Journal of Biomechanical Engineering*, 137(2), 020905, 2015.

[74] C. Oefner et al. Reporting checklist for verification and validation of finite element analysis in orthopedic and trauma biomechanics. *Medical Engineering & Physics*, 92, 25–32, 2021.

[75] M. Lilja, S. Johansson, and T. Öberg. Relaxed versus activated stump muscles during casting for trans-tibial prostheses. *Prosthetics and Orthotics International*, 23, 13–20, 1999.

[76] J.E. Sanders and S. Fatone. Residual limb volume change: systematic review of measurement and management. *Journal of Rehabilitation Research and Development*, 48, 949–986, 2011.

[77] F. Golbranson et al. Volume changes occurring in postoperative below-knee residual limbs. *Journal of Rehabilitation Research and Development*, 25, 11–18, 1988.

[78] A.T. Tantua et al. Reduction of residual limb volume in people with transtibial amputation. *Journal of Rehabilitation Research and Development*, 51, 1119–1126, 2014.

[79] S.G. Zachariah et al. Shape and volume change in the transtibial residuum over the short term: preliminary investigation of six subjects. *Journal of Rehabilitation Research and Development*, 41, 683–694, 2004.

[80] G.M. Street. Vacuum suspension and its effects on the limb. *Orthopädie Technik*, 4, 4–7, 2007.

[81] R.T. Youngblood et al. Effects of activity intensity, time, and intermittent doffing on daily limb fluid volume change in people with transtibial amputation. *Prosthetics and Orthotics International*, 43, 28–38, 2019.

[82] S. Chatterjee et al. Review: problems with use of transtibial prosthesis. *Journal of Medical Imaging and Health Informatics*, 6, 269–284, 2016.

[83] J.E. Sanders, E.L. Rogers, and D.C. Abrahamson. Assessment of residual-limb volume change using bioimpedance. *Journal of Rehabilitation Research and Development*, 44, 525–535, 2007.

[84] G.R. Fernie, P.J. Holliday, and R.J. Lobb. An instrument for monitoring stump oedema and shrinkage in amputees. *Prosthetics and Orthotics International*, 2, 69–72, 1978.

[85] K.E. Smith et al. Validation of spiral CT and optical surface scanning for lower limb stump volumetry. *Prosthetics and Orthotics International*, 19, 97–107, 1995.

[86] G.R. Fernie et al. Shape sensing for computer aided below-knee prosthetic socket design. *Prosthetics and Orthotics International*, 9, 12–16, 1985.

[87] A.W.P. Buis et al. Magnetic resonance imaging technology in transtibial socket research: a pilot study. *Journal of Rehabilitation Research and Development*, 43, 883–890, 2006.

3

Muscle Constitutive Model with a Tangent Modulus Approximation: Ansys Implementation and Verification

The first main contribution of this thesis is the implementation and verification study of an approximation for the tangent modulus applied to a muscle constitutive model. One of the specific aims of this thesis was to develop a new muscle material model, which we implemented in Ansys using a USERMAT subroutine. Deriving and programming the tangent modulus, which is a fourth-order tensor, is challenging for nonlinear material models such as the muscle model we planned to implement. The strategy proposed by Sun et al. [1] for approximating the tangent modulus could simplify our muscle material model implementation, but the strategy had not yet been validated for muscle models. Initially, we applied the approximation to Mooney–Rivlin, Yeoh, and Exponential Anisotropic material models in uniaxial, shear, torsion, and equitriaxial test cases. We presented the results from those simulations at the Congress of the International Society of Biomechanics (Sampaio de Oliveira, M.L., Uchida, T.K. Approximation method to

calculate the elasticity tensor for hyperelastic finite element models, XXVIII Congress of the International Society of Biomechanics, July 25–29, 2021, Stockholm, Sweden).

We then applied the approximation to the muscle material model proposed by Blemker et al. [2]. This model is a nearly incompressible hyperplastic model that includes both isochoric and volumetric strain energy components. The approximation algorithm is convenient as it requires only the calculated stress and its implementation is independent of the specific muscle model that is being used. Most existing muscle models provide fiber force–length curves, which can be transformed into stress–strain curves; therefore, the tangent modulus can be readily calculated using this approximation algorithm, without the need for additional computations. We validated our Ansys implementation using three geometrical test models and compared the simulated stress and strain with those obtained from analogous simulations that we developed in FEBio, which uses an exact calculation of the tangent modulus. Our results using the approximation showed good agreement with FEBio for stress and strain, though discrepancies arose in regions of high stress concentration and where elements were highly skewed. Among the three test models, the root-mean-square error ranged from approximately 0.00% to 6.75%. This work facilitates broader adoption of advanced muscle models in Ansys by reducing implementation complexity. We shared our implementation so that others can reproduce and extend our results, and we presented this work at the North American Congress of Biomechanics (Sampaio de Oliveira, M.L., Uchida, T.K. Elasticity tensor applied to a finite element muscle tensor, North American Congress of Biomechanics, August 21–25, 2022, Ottawa, Canada). This work has been published in the ASME Journal of Biomechanical Engineering (Sampaio de Oliveira, M.L., Uchida, T.K. Muscle constitutive model with a tangent modulus approximation: Ansys implementation and verification. ASME Journal of Biomechanical Engineering, 145(7): 071002, 2023, doi: 10.1115/1.4056948). The content of this paper is provided in the remainder of this chapter.

3.1 Abstract

Sophisticated muscle material models are required to perform detailed finite element simulations of soft tissue; however, state-of-the-art muscle models are not among the built-in materials in popular commercial finite element software packages. Implementing user-defined muscle material models is challenging for two reasons: deriving the tangent modulus tensor for a material with a complex strain energy function is tedious and programming the algorithm to compute it is error-prone. These challenges hinder widespread use of such models in software that employs implicit, nonlinear, Newton-type finite element methods. We implement a muscle material model in Ansys using an approximation of the tangent modulus, which simplifies its derivation and implementation. Three test models were constructed by revolving a rectangle (RR), a right trapezoid (RTR), and a generic obtuse trapezoid (RTO) around the muscle's centerline. A displacement was applied to one end of each muscle, holding the other end fixed. The results were validated against analogous simulations in FEBio, which uses the same muscle model but with the exact tangent modulus. Overall, good agreement was found between our Ansys and FEBio simulations, though some noticeable discrepancies were observed. For the elements along the muscle's centerline, the root-mean-square-percentage error in the Von Mises stress was 0.00%, 3.03%, and 6.75% for the RR, RTR, and RTO models, respectively; similar errors in longitudinal strain were observed. We provide our Ansys implementation so that others can reproduce and extend our results.

Keywords: Ansys; elasticity tensor; finite element; hyperelastic material; muscle model; tangent modulus.

3.2 Introduction

Computational models of skeletal muscle can be used to understand and predict the behavior of muscle during human movement. For example, muscle models have been used to predict the force and stress distribution on the tissues of the lower limb during external compression [3] and to explore why, among the hamstring muscles, the biceps femoris long head is the most susceptible to large strain [4]. Most three-dimensional muscle constitutive models are developed using custom finite element (FE) codes. Models with various levels of complexity have been proposed [2, 5–8], but the difficulty of implementing even simple user-defined muscle models has hindered the widespread adoption of state-of-the-art muscle material models when using commercial software packages. Consequently, research on muscle material models is lacking in general and has suffered from poor reproducibility, both of which have hindered progress in developing new models of skeletal muscle [9].

With the steadily improving processing power of computers, FE models have become popular tools to study muscle behavior. Muscle models can be divided into two broad categories: phenomenological models and biophysical models. Phenomenological muscle models attempt to fit equations to experimental data, without attempting to describe the biological mechanisms by which the observed behaviors emerge. Hill [10] developed the first phenomenological muscle model from experiments that characterized the relationship between the shortening velocity of muscle fibers and the force they produce. This model was later extended into the popular "Hill-type" muscle model commonly used today [11]. In contrast, biophysical muscle models represent the response of muscle by simulating their intrinsic physiological properties [9, 12, 13]. For example, the underlying aspects of muscle contraction were only understood after Huxley [14] proposed a microstructural model based on the cross-bridge theory. Incorporating the physiological aspects of muscle into a constitutive model can be a complex task. For example, introducing electromechanical dynamics into a hyperelastic material model requires more

equations as well as more parameters, both of which increase the complexity of the model [12].

The force generated by muscle is often described as consisting of two components: an active component that depends on the neural activation and a passive component that is independent of activation. Muscles can be simulated using only the passive component if their passive behavior predominates. Linear elastic materials [4], nonlinear hyperelastic materials [15–20], and nonlinear elastic–viscoplastic materials [21] have been used. On the other hand, some simulations require one to consider the effect of muscle activation (e.g., to study the causes of the nonuniform strain in the biceps brachii [2]). The active component of muscle force generation has been considered by introducing an additional term to the strain energy calculation in a hyperelastic transversely isotropic material model [2, 5–7, 22–25]. Another strategy is to mix different element types in a finite element model. For example, Li et al. [26] used solid elements and beam elements to represent the passive and active behavior of muscle, respectively. While popular software packages such as Ansys® and Abaqus® have built-in material models to simulate the passive behavior of muscle (e.g., Neo-Hookean and Mooney–Rivlin models), they lack built-in functions to simulate the active behavior. FEBio, an open-source finite element software package [27, 28], has a built-in muscle model capable of simulating the active component of muscle. Introducing advanced muscle modeling capabilities into Ansys and Abaqus would benefit researchers and engineers as these software packages have many powerful features, broad supporting infrastructure, and large user bases in fields beyond biomechanics. For example, engineers are increasingly interested in virtual prototyping and optimization of robotic devices that interact with the body [29, 30], and Ansys or Abaqus would be a natural choice for designing the robotic components. Providing freely available implementations of sophisticated muscle material models in these software packages would facilitate adoption and accelerate future development.

Muscle material models can be implemented in Ansys and Abaqus by writing a user-defined material subroutine (USERMAT or UMAT, respectively). The Cauchy stress and

tangent modulus tensor (also referred to as the elasticity tensor or material Jacobian) are the outputs of these subroutines. The tangent modulus is used as an iterative operator for a Newton-type method to solve initial- and boundary-value problems in FE simulations [31]. For example, the governing equilibrium equations used in finite element methods can be constructed using the principle of virtual work. The resulting nonlinear equations can then be solved using Netwon's method, which involves first linearizing these equations. The second Piola–Kirchhoff stress that appears in the equilibrium equations is linearized using the tangent modulus [31, 32]. USERMAT subroutines also use the tangent modulus associated with the Jaumann rate of the Kirchhoff stress [33]. Computing the Cauchy stress is relatively straightforward because one must calculate only the first derivative of the strain energy and the result is only a second-order tensor (i.e., a matrix). On the other hand, deriving the tangent modulus tensor for a material with a complex strain energy function is tedious because it depends on the second derivative of the strain energy. Furthermore, programming the algorithm to compute the tangent modulus is error-prone because the tangent modulus is a fourth-order tensor and requires one to also implement routines to perform tensor algebra. Additionally, Ansys does not provide extensive tools to support USERMAT development.

Approximation methods have been proposed as an alternative to calculating the tangent modulus. Miehe [34] proposed a forward-difference approximation of the second Piola–Kirchhoff stress to calculate the tangent modulus. Sun et al. [1] proposed an approximation based on the forward difference of the associated Kirchhoff stress by incrementally perturbing the deformation gradient. Tanaka and Fujikawa [35] proposed a higher-order approximation scheme, using the second-order central difference and the complex-step derivative approximation (the latter being a technique to obtain the numerical value of the first derivative using complex arithmetic). All the aforementioned strategies demonstrated good agreement with the exact calculations, but their convergence and accuracy are not guaranteed for all materials and elements [36].

An approximation for the tangent modulus has been applied to several materials,

among which are the Neo-Hookean model, Holzapfel's exponential model [1], the Mooney–Rivlin model [35], and the Gasser–Ogden–Holzapfel model [37–40]. However, to the best of the authors' knowledge, an approximation method for the tangent modulus has not yet been applied to a muscle material model. Such an approximation can greatly facilitate the implementation of muscle material models (whose strain energy functions are nontrivial).

In this work, we apply the approximation strategy of Sun et al. [1] to the muscle material model proposed by Blemker et al. [2]. We implement this model as an Ansys user-defined material subroutine. We use several test cases to evaluate the convergence properties and the error in stress and strain when using the approximation. We validate our results through comparison to analogous simulations in FEBio, which uses the same muscle model but with the exact value of the tangent modulus. We provide our Ansys implementation at https://simtk.org/projects/musc_fe_approx so that others can reproduce and extend our results, use the model in their FE simulations, and develop new muscle material models in Ansys more quickly.

3.3 Methods

For completeness, we begin this section with a summary of some necessary kinematics concepts in continuum mechanics. We then present the formulation to calculate the stress and tangent modulus of a material, which are required to implement a user-defined material model in Ansys. Next, we present the tangent modulus approximation proposed by Sun et al. [1] and the hyperelastic muscle material model proposed by Blemker et al. [2]. Finally, we describe the models and simulations we used to validate our Ansys implementation of this muscle material model with a tangent modulus approximation.

3.3.1 Kinematics

The deformation gradient ($\mathbf{F}$) is the primary measure of deformation used in nonlinear continuum mechanics:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \tag{3.1}$$

where $\mathbf{x}$ is the position in the current configuration (Ω) at a given time t and $\mathbf{X}$ is the position in the reference configuration (Ω_0) [31]. The deformation gradient contains information about the rotational and stretching parts of the deformation. From the deformation gradient, the left ($\mathbf{B}$) and right ($\mathbf{C}$) Cauchy–Green deformation tensors are calculated as follows:

$$\mathbf{B} = \mathbf{F}\mathbf{F}^{\mathrm{T}} \tag{3.2}$$

$$\mathbf{C} = \mathbf{F}^{\mathrm{T}}\mathbf{F} \tag{3.3}$$

For materials that are nearly incompressible, such as muscles, the deformation gradient is often decomposed into its volumetric component ($J^{1/3}\mathbf{I}$) and isochoric component ($\bar{\mathbf{F}}$):

$$\mathbf{F} = \left(J^{1/3}\mathbf{I}\right)\bar{\mathbf{F}} \tag{3.4}$$

where $J = |\mathbf{F}|$ is the volumetric ratio and $\mathbf{I}$ is an identity matrix. Consequently, the left and right Cauchy–Green deformation tensors become $\bar{\mathbf{B}}$ and $\bar{\mathbf{C}}$, respectively, and are expressed as follows:

$$\bar{\mathbf{B}} = J^{-2/3}\mathbf{B} \tag{3.5}$$

$$\bar{\mathbf{C}} = J^{-2/3}\mathbf{C} \tag{3.6}$$

3.3.2 Stress measures

Several stress measures are used in nonlinear problems; the second Piola–Kirchhoff, Cauchy, and Kirchhoff stresses are the most common. The second Piola–Kirchhoff stress

($\mathbf{S}$) is defined from the derivative of the strain energy function (Ψ) with respect to the right Cauchy–Green deformation tensor:

$$\mathbf{S} = 2\frac{\partial \Psi(\mathbf{C})}{\partial \mathbf{C}} \tag{3.7}$$

For materials that are nearly incompressible, the strain energy is a function of the modified right Cauchy–Green deformation tensor (i.e., $\Psi(\bar{\mathbf{C}})$). Thus, using the chain rule and separating the strain energy into the isochoric and volumetric components, we can calculate the isochoric ($\mathbf{S}_{\text{iso}}$) and volumetric ($\mathbf{S}_{\text{vol}}$) components of the second Piola–Kirchhoff stress as follows:

$$\mathbf{S}_{\text{iso}} = J^{-2/3}\left(\mathbb{I} - \frac{1}{3}\mathbf{C}^{-1} \otimes \mathbf{C}\right) : \bar{\mathbf{S}} \tag{3.8}$$

$$\mathbf{S}_{\text{vol}} = Jp\mathbf{C}^{-1} \tag{3.9}$$

where $\mathbb{I}$ is a unit fourth-order tensor; "$\otimes$" and ":" are the tensor product and double-contraction operators, respectively. $\bar{\mathbf{S}}$ is the "fictitious" second Piola–Kirchhoff stress:

$$\bar{\mathbf{S}} = 2\frac{\partial \Psi(\bar{\mathbf{C}})}{\partial \bar{\mathbf{C}}} \tag{3.10}$$

and p is the hydrostatic pressure, which can be interpreted as a Lagrange multiplier that enforces near-incompressibility:

$$p = \frac{d\Psi_{\text{vol}}(J)}{dJ} \tag{3.11}$$

The Cauchy and Kirchhoff stresses are calculated using the push-forward operator, which transforms a vector or tensor from the material configuration to the spatial configuration. The Kirchhoff stress ($\boldsymbol{\tau}$) is calculated as the push-forward transformation of the second Piola–Kirchhoff stress ($\mathbf{S}$):

$$\boldsymbol{\tau} = \mathbf{F}\mathbf{S}\mathbf{F}^{\mathrm{T}} \tag{3.12}$$

The Cauchy stress ($\boldsymbol{\sigma}$) is related to the Kirchhoff stress as follows:

$$\boldsymbol{\sigma} = J^{-1}\boldsymbol{\tau} \tag{3.13}$$

3.3.3 Tangent modulus and its approximation

Ansys uses a co-rotational frame to express the stress and tangent modulus [33]. Thus, one must use an objective rate of stress to account for the material rotation [38]. Ansys uses the objective Jaumann rate of Cauchy stress ($\overset{\nabla}{\boldsymbol{\sigma}}$) and the associated tangent modulus ($\mathbb{C}^J$), which is a fourth-order tensor [33]:

$$\overset{\nabla}{\boldsymbol{\sigma}} = \mathbb{C}^J : \mathbf{D} \tag{3.14}$$

where $\mathbf{D}$ (the symmetric part of the spatial velocity gradient $\mathbf{L} = \mathbf{F}\mathbf{F}^{-1}$) is the rate-of-deformation tensor. The Jaumann stress rate can also be expressed in terms of the spatial tangent modulus ($\mathbb{C}^S$), the rate-of-deformation tensor ($\mathbf{D}$), and the Cauchy stress ($\boldsymbol{\sigma}$) [39]:

$$\overset{\nabla}{\boldsymbol{\sigma}} = \mathbb{C}^S : \mathbf{D} + \mathbf{D}\boldsymbol{\sigma} + \boldsymbol{\sigma}\mathbf{D}^{\mathrm{T}} \tag{3.15}$$

Upon subtracting Eq. (3.14) from Eq. (3.15), we find that $\mathbb{C}^J$ can be calculated by adding a term to $\mathbb{C}^S$. The index notation of $\mathbb{C}^J$ is as follows:

$$\mathbb{C}^J_{ijk\ell} = \mathbb{C}^S_{ijk\ell} + \frac{1}{2}\left(I_{ik}\sigma_{j\ell} + \sigma_{ik}I_{j\ell} + I_{i\ell}\sigma_{jk} + \sigma_{i\ell}I_{jk}\right) \tag{3.16}$$

where i, j, k, and ℓ are the free indices in $\{1, 2, 3\}$. $\mathbb{C}^S$ can be found from the push-forward transformation of the material tangent modulus ($\mathbb{C}$) divided by the volumetric ratio (J):

$$\mathbb{C}^S_{ijk\ell} = J^{-1}F_{iA}F_{jB}F_{kC}F_{\ell D}\mathbb{C}_{ABCD} \tag{3.17}$$

Finally, $\mathbb{C}$ can be calculated from the second derivative of the strain function:

$$\mathbb{C} = 4\frac{\partial^2 \Psi(\mathbf{C})}{\partial \mathbf{C} \partial \mathbf{C}} \tag{3.18}$$

Deriving the tangent modulus (Eqs. (3.16) and (3.17)), a fourth-order tensor, is tedious for a material with a complex strain energy function and programming the algorithm to compute it is error-prone. These challenges can limit the widespread use of state-of-the-art hyperelastic material models. Sun et al. [1] proposed a numerical approximation for the tangent modulus that addresses these challenges. The method of Sun et al. is particularly convenient for the present application as the approximation is performed in the co-rotational frame, which is the frame with respect to which USERMAT outputs must be expressed. The formulation begins by calculating matrices $\Delta\mathbf{F}^{(i,j)}$ for $i, j \in \{1, 2, 3\}$, which are the perturbations of the deformation gradient $\mathbf{F}$ upon perturbing only its (i, j) element:

$$\Delta\mathbf{F}^{(i,j)} = \frac{\epsilon}{2}\left(e_i \otimes e_j\mathbf{F} + e_j \otimes e_i\mathbf{F}\right) \tag{3.19}$$

where $\epsilon > 0$ is the (small) perturbation parameter and $\{e_1, e_2, e_3\}$ are the basis vectors in the spatial reference frame. Next, the perturbed deformation gradient $\hat{\mathbf{F}}^{(i,j)}$ is computed for each perturbation:

$$\hat{\mathbf{F}}^{(i,j)} = \mathbf{F} + \Delta\mathbf{F}^{(i,j)} \tag{3.20}$$

The elements of the approximated tangent modulus are then calculated as the forward difference of the Cauchy stress:

$$\mathbb{C}^J_{ijk\ell} = \frac{1}{J\epsilon}\left[J_p^{(i,j)}\boldsymbol{\sigma}\left(\hat{\mathbf{F}}^{(i,j)}\right)_{k\ell} - J\boldsymbol{\sigma}\left(\mathbf{F}\right)_{k\ell}\right] \tag{3.21}$$

where $J_p^{(i,j)} = \left|\hat{\mathbf{F}}^{(i,j)}\right|$. This approximation has the advantage that Eqs. (3.19) to (3.21) are independent of the material model—that is, they depend only on the Cauchy stress, the closed form of which is a required calculation in any Ansys USERMAT subroutine

and is, therefore, readily available.

3.3.4 Hyperelastic muscle material model

We applied the approximation of the tangent modulus proposed by Sun et al. [1] (described in Sec. 3.3.3) to the hyperelastic muscle material model of Blemker et al. [2]; for completeness, we briefly describe the latter in this section. The isochoric component of the strain energy (Ψ_{iso}) is formed by adding three terms: the along-fiber and cross-fiber shear strain energies (W_1 and W_2) and the muscle fiber stretch strain energy (W_3):

$$\Psi_{\text{iso}}\left(\bar{I}_1, \bar{I}_2, \bar{I}_4, \bar{I}_5, \alpha\right) = W_1\left(B_1\left(\bar{I}_4, \bar{I}_5\right)\right) + W_2\left(B_2\left(\bar{I}_1, \bar{I}_4, \bar{I}_5\right)\right) + W_3\left(\lambda, \alpha\right) \tag{3.22}$$

where $\bar{I}_1$, $\bar{I}_2$, $\bar{I}_4$, and $\bar{I}_5$ are invariants of the modified right Cauchy–Green tensor ($\bar{\mathbf{C}}$). (Note that muscle is modeled as a nearly incompressible material, hence the use of $\bar{\mathbf{C}}$.) In Eq. (3.22), functions W_1, W_2, B_1, and B_2 are defined as follows:

$$W_1 = G_1 B_1^2 \tag{3.23}$$

$$W_2 = G_2 B_2^2 \tag{3.24}$$

$$B_1 = \sqrt{\bar{I}_5/\bar{I}_4 - 1} \tag{3.25}$$

$$B_2 = \cosh^{-1}\left(\frac{\bar{I}_1\bar{I}_4 - \bar{I}_5}{2\sqrt{\bar{I}_4}}\right) \tag{3.26}$$

where G_1 and G_2 are the along-fiber and cross-fiber shear moduli. The volumetric component of the strain energy (Ψ_{vol}) is a function of only the volumetric ratio (J):

$$\Psi_{\text{vol}}(J) = \frac{K}{2}\ln^2(J) \tag{3.27}$$

where K is the effective bulk modulus. The total strain energy is simply the sum of the isochoric and volumetric components thereof:

$$\Psi = \Psi_{\text{iso}} + \Psi_{\text{vol}} \tag{3.28}$$

The muscle fiber stretch strain energy (W_3) in Eq. (3.22) is associated with the Cauchy stress generated by the fiber stretch (σ_{fiber}). W_3 has two components: the passive stress and the active stress. The former depends on the fiber stretch (λ) while the latter depends on both the fiber stretch and the muscle activation (α, where $0 \leq \alpha \leq 1$). It is assumed that all sarcomeres in a fiber have the same length, as do all fibers. The Cauchy stress is related to the strain energy as follows:

$$\lambda \frac{\partial W_3}{\partial \lambda} = \sigma_{\text{fiber}}(\lambda, \alpha) = \sigma_{\max} \frac{\lambda}{\lambda_{\text{opt}}} \left(\text{f}_{\text{passive}}(\lambda) + \alpha \text{f}_{\text{active}}(\lambda) \right) \tag{3.29}$$

where $\sigma_{\max}$ is the maximum isometric stress, which is the maximum isometric force of the muscle divided by its physiological cross-sectional area, and λ_{opt} is the optimal fiber stretch, which corresponds to the length of the fiber at which the sarcomeres are at their optimal length (i.e., the length at which they can generate the maximum active force). The function $f_{\text{passive}}(\lambda)$ is the passive force normalized by the maximum isometric force and it is modeled as a piecewise exponential function of the fiber stretch [2]:

$$f_{\text{passive}}(\lambda) = \begin{cases} 0, & \text{if } \lambda \leq \lambda_{\text{opt}} \\ P_1 \left(e^{P_2(\lambda/\lambda_{\text{opt}} - 1)} - 1 \right), & \text{if } \lambda_{\text{opt}} < \lambda < \lambda^* \\ P_3 \lambda / \lambda_{\text{opt}} + P_4, & \text{if } \lambda \geq \lambda^* \end{cases} \tag{3.30}$$

where P_1, P_2, P_3, and P_4 are constants. Note that the values for P_3 and P_4 must preserve C^0 and C^1 continuity at $\lambda = \lambda^*$, where λ^* is the stretch above which the passive function is linear. The last term in Eq. (3.29) is the active force component. Specifically, $f_{\text{active}}(\lambda)$ is the active force normalized by the maximum isometric force and it is defined

as a piecewise quadratic function of the fiber stretch [2]:

$$f_{\text{active}}(\lambda) = \begin{cases} 9\left(\lambda/\lambda_{\text{opt}} - 0.4\right)^2, & \text{if } \lambda \leq 0.6\lambda_{\text{opt}} \\ 1 - 4\left(1 - \lambda/\lambda_{\text{opt}}\right)^2, & \text{if } 0.6\lambda_{\text{opt}} < \lambda < 1.4\lambda_{\text{opt}} \\ 9\left(\lambda/\lambda_{\text{opt}} - 1.6\right)^2, & \text{if } \lambda \geq 1.4\lambda_{\text{opt}} \end{cases} \tag{3.31}$$

3.3.5 Models to verify the approximation

Five models were created to validate our Ansys USERMAT subroutine that applies the approximation of the tangent modulus described in Sec. 3.3.3 to the hyperelastic muscle material model described in Sec. 3.3.4. Three "quarter-muscle" models were constructed by revolving a rectangle, a right trapezoid, and a generic obtuse trapezoid around the centerline of the muscle (Fig. 3.1); we refer to these models as RR, RTR, and RTO, respectively. (The dimensions of these models were based on the "Variation I" and "Variation II" models illustrated in Blemker et al. [2].) The fibers were oriented along the X-axis. All meshes were created using the "multizone" method, which automatically decomposes the geometry into sweepable and free regions (i.e., regions in which the meshes are structured and unstructured, respectively) [41]. All models used eight-node hexahedral elements (following the methods of Blemker et al. [2] for the "Variation I" and "Variation II" models), with size 0.003 m and the following constitutive parameters: $G_1 = G_2 = 50$ kPa, $\sigma_{\max} = 100$ kPa, $\lambda^* = \lambda_{\text{opt}} = 1.4$, $\alpha = 0.15$, $P_1 = 0.05$, and $P_2 = 6.6$. Unless otherwise specified, the value of the effective bulk modulus was $K = 5.0 \times 10^5$ kPa. These parameters are similar to those used elsewhere [4, 42].

To investigate the effect of element skewness along the centerline, we created two additional versions of the RTR model. Specifically, we generated models with inclination angles (θ) of 5.05° and 54.58° in addition to the "nominal" RTR model shown in Fig. 3.1 (where $\theta = 19.37°$) by changing the length of the anterior line, keeping the length of the centerline constant. (Note that all models, including these variants of the RTR model,

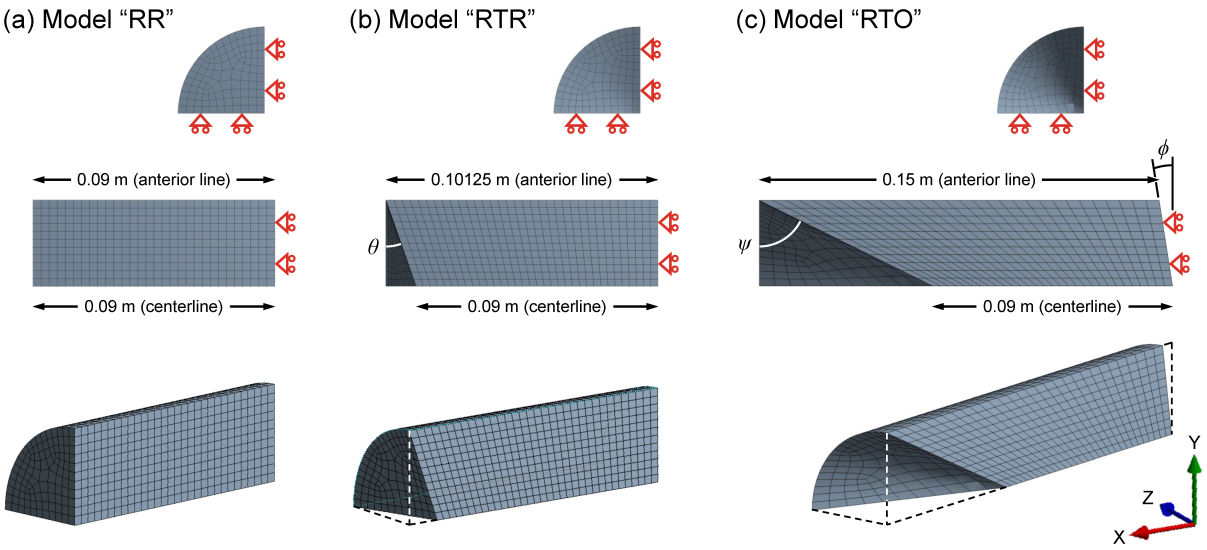

Figure 3.1: Models used to verify our Ansys implementation. 3D models were generated by revolving (a) a rectangle (RR), (b) a right trapezoid (RTR), and (c) an obtuse trapezoid (RTO) around the centerline. Cross-sections in the YZ-plane (top) and XY-plane (middle); 3D view (bottom). Small triangles indicate constraints on the displacement. All models have radius 0.032 m. The angles labeled in the RTR and RTO models are $\theta = 19.37°$, $\psi = 63.79°$, and $\phi = 8.88°$.

have the same centerline length.) Increasing the inclination angle increases the skewness of the elements along the centerline near the slanted face. The number of elements and nodes in each model are provided in Table 3.1.

3.3.6 Simulations to verify the approximation

The models described in Sec. 3.3.5 were simulated by performing static analyses in Ansys. A displacement of 0.04 m was applied to the left face of each model, as shown

Table 3.1: Number of elements and nodes in each model. Models were constructed by revolving a rectangle (RR), a right trapezoid (RTR), and a generic obtuse trapezoid (RTO) around the centerline of the muscle. In the RTR model, θ is the angle of the slanted face relative to vertical.

	RR	RTR ($\theta = 5.05°$)	RTR ($\theta = 19.37°$)	RTR ($\theta = 54.58°$)	RTO
Elements	3270	3379	4046	8415	3270
Nodes	4030	4160	4935	9936	4030

in the middle row of Fig. 3.1, directed in the positive X-direction (i.e., stretching the fibers). The motion of the opposite face was constrained in the X-direction (triangles in Fig. 3.1, middle row). Furthermore, the motion of the faces in the XY- and XZ-planes was constrained in the direction normal to each plane (triangles in Fig. 3.1, top row). In Ansys, the "large deformation" option was selected and the convergence options "auto time stepping," "force convergence," and "displacement convergence" were set to "program controlled."

Simulations of the RR model were used to determine an appropriate value for the perturbation parameter (ϵ) in the approximation method. Specifically, the rate of convergence was evaluated by noting the number of iterations required to reach equilibrium when $\epsilon = 10^k$ for $k \in \{-12, -10, -8, -6, -4\}$.

To validate our results, we compared the stress and strain obtained using our implementation in Ansys with the results from analogous simulations performed in FEBio, which uses the same muscle model but with the exact value of the tangent modulus. Specifically, we compared the stress and longitudinal strain for the elements along the anterior line and centerline of each model. The stress was a direct output from Ansys and FEBio; the longitudinal strain (s) was calculated from the longitudinal deformation of each element:

$$s = \frac{L - L_0}{L_0} \tag{3.32}$$

where L_0 and L are the initial and final lengths of the element, respectively. Note that all models described in Sec. 3.3.5 have the same centerline length and are subjected to the same displacement, thus the total strain along the centerline is the same in each simulation.

We used two metrics to quantify the error between the Ansys and FEBio results. The first metric was the relative percent error (RPErr):

$$\text{RPErr} = \left| \frac{\zeta_{\text{FEBio}} - \zeta_{\text{Ansys}}}{\zeta_{\text{FEBio}}} \right| \times 100\% \tag{3.33}$$

where ζ is either stress or longitudinal strain. The largest RPErr for any element along either the anterior line or the centerline of the muscle (maxRPErr) was used to quantify the largest deviation between the Ansys and FEBio results. The second metric we used was the root-mean-square-percentage error (RMSPE), which quantifies the average error:

$$\text{RMSPE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left[\left(\frac{\zeta_{\text{FEBio},k} - \zeta_{\text{Ansys},k}}{\zeta_{\text{FEBio},k}} \times 100\%\right)^2\right]} \tag{3.34}$$

where N is the number of elements along either the anterior line or the centerline.

Several authors have reported an anomalous, nonphysical growth of stress in uniaxial tests when the volumetric and isochoric components of strain energy were decoupled in isotropic materials [43] and anisotropic materials [6, 44] (Eq. (3.28)). Therefore, we anticipated similar behavior in our simulations. We used the $\theta = 54.58°$ variant of the RTR model to explore how the effective bulk modulus (K) affects the stress and strain computed using our Ansys implementation. Specifically, we compared the longitudinal stress and longitudinal strain from simulations using four values of K: $10G_1$, 10^2G_1, 10^3G_1, and 10^4G_1.

3.4 Results and discussion

3.4.1 Selection of the perturbation parameter

The number of iterations required to reach equilibrium at each simulation step when using the approximation of the tangent modulus is shown in Table 3.2. The algorithm required more iterations to achieve convergence when $\epsilon = 10^{-4}$ because, for large values of ϵ, the error of the forward difference approximation increased (Eq. (3.20)). On the other hand, when ϵ was very small, the perturbation approached machine precision and numerical error polluted the approximation calculation.

The Von Mises stress and the strain in the longitudinal direction were insensitive to the perturbation parameter ϵ. Stress and strain were nearly perfectly uniform across

Table 3.2: Number of iterations required to reach equilibrium at each simulation step for different values of perturbation parameter ϵ, using the RR model. The simulation was inefficient when using the largest value of ϵ and did not converge ("DNC") when using the smallest value.

	Perturbation parameter (ϵ)				
Simulation step	10^{-12}	10^{-10}	10^{-8}	10^{-6}	10^{-4}
1	DNC	2	2	2	2
2 to 18	DNC	1	1	1	1
19	DNC	1	1	1	2
20	DNC	1	1	1	11
Total	DNC	21	21	21	32

Table 3.3: Sensitivity of Von Mises stress and longitudinal strain to perturbation parameter ϵ, using the RR model. Data are shown for the element along the centerline at the free end of the muscle.

	Perturbation parameter (ϵ)			
	10^{-10}	10^{-8}	10^{-6}	10^{-4}
Stress (Pa)	16494	16494	16495	16495
Strain	0.443	0.443	0.443	0.443

all elements of the RR model; data for one representative element are reported in Table 3.3. Our results agree with those reported by Sun et al. [1], who observed that the approximation for a Neo-Hookean material demonstrated the best convergence when ϵ was between 10^{-10} and 10^{-6}. Based on the observations of Sun et al. and the results shown in Tables 3.2 and 3.3, we used $\epsilon = 10^{-8}$ in the remainder of our study.

3.4.2 Simulations of RR, RTR, and RTO models

The deformed models, shaded to visualize fiber stretch, are shown in Fig. 3.2. The Von Mises stress along the anterior line and centerline of each model is shown in Figs. 3.3(a) and 3.3(b), respectively. In the RR model, the Von Mises stress was constant along the entire length of the muscle, along both the anterior line and the centerline. The relative percent error (RPErr) of stress between the Ansys and FEBio solutions was 0.00% for all elements along the anterior line and centerline; the root-mean-square-percentage error (RMSPE) calculated along these lines was therefore 0.00% as well (Table 3.4). In the RTR model, the stress remained relatively constant for most of the length of the muscle. Specifically, for all elements from the fixed end to approximately 71% of the distance along the anterior line and centerline, the stress deviated by no more than 1.23% and 3.58% of the stress at the fixed end, respectively. Beyond 71% of the distance from the fixed end, the stress decreased by up to 21.06% and increased by up to 20.35% of the stress at the fixed end along the anterior line and centerline, respectively. The maximum RPErr (maxRPErr) of the stress over all elements along the anterior line was 3.68% (Table 3.4) and occurred at the free end; along the centerline, the maxRPErr was 15.09% and occurred at approximately 97% of the distance from the fixed end. In the RTO model, the stress varied along the entire anterior line. Overall, we observed good agreement between the Von Mises stress from our Ansys implementation and the analogous simulations in FEBio, except in the region near the free end along the centerline in the RTO model (Fig. 3.3(b)). The stress in this region increased dramatically toward the free end in both Ansys and FEBio simulations. In general, similar observations were noted for longitudinal stress, as shown in Figs. 3.3(c) and 3.3(d).

The strain from our Ansys simulations was also in good agreement with the results from FEBio; the longitudinal strain along the anterior line and centerline of each model is shown in Figs. 3.3(e) and 3.3(f), respectively. As was the case for the Von Mises and longitudinal stresses, the error in longitudinal strain was least in the RR model and greatest in the RTO model (Table 3.4). The peak errors in strain generally occurred in

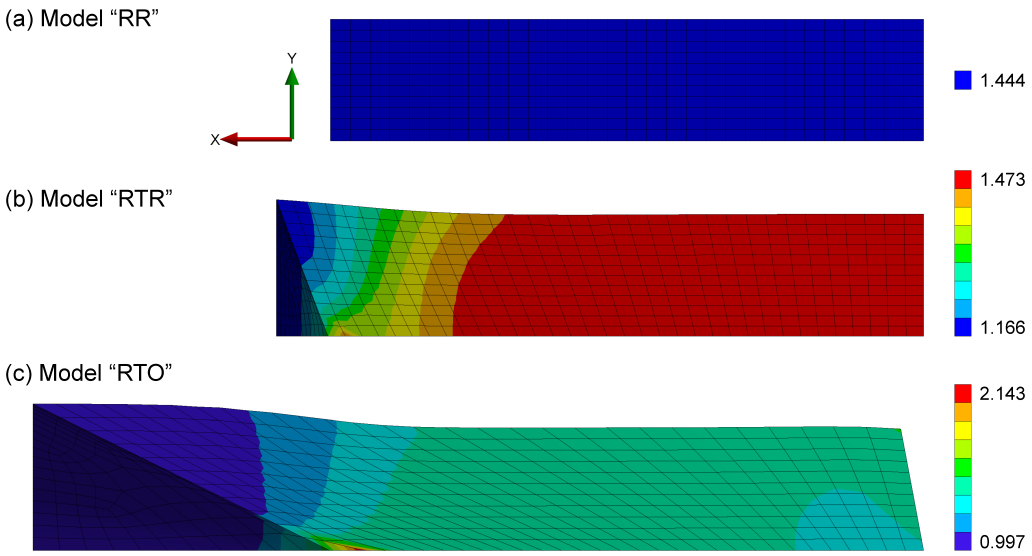

Figure 3.2: Visualization of fiber stretch after deformation of (a) the RR model, (b) the RTR model, and (c) the RTO model.

the same regions as the peak errors in stress. In the RTO model, the strain exhibited non-smooth behavior near the free end. Note that this behavior was observed in the Ansys model as well as in the FEBio model, the latter of which uses the same muscle model but with the exact value of the tangent modulus.

Table 3.4: Error in Von Mises stress, longitudinal stress, and longitudinal strain between Ansys and FEBio simulations for the RR, RTR, and RTO models. The maximum relative percent error (maxRPErr) and root-mean-square-percentage error (RMSPE) were calculated over the elements along the anterior line and centerline in each model.

		RR		RTR ($\theta = 19.37°$)		RTO	
		Anterior	Center	Anterior	Center	Anterior	Center
Von Mises	maxRPErr	0.00%	0.00%	3.68%	15.09%	7.47%	25.55%
Stress	RMSPE	0.00%	0.00%	0.77%	3.03%	3.08%	6.75%
Longitudinal	maxRPErr	0.00%	0.00%	2.39%	12.35%	7.01%	42.93%
Stress	RMSPE	0.00%	0.00%	0.88%	2.52%	4.88%	8.52%
Longitudinal	maxRPErr	0.00%	0.00%	3.86%	7.87%	25.63%	28.32%
Strain	RMSPE	0.00%	0.00%	1.48%	1.71%	9.52%	6.07%

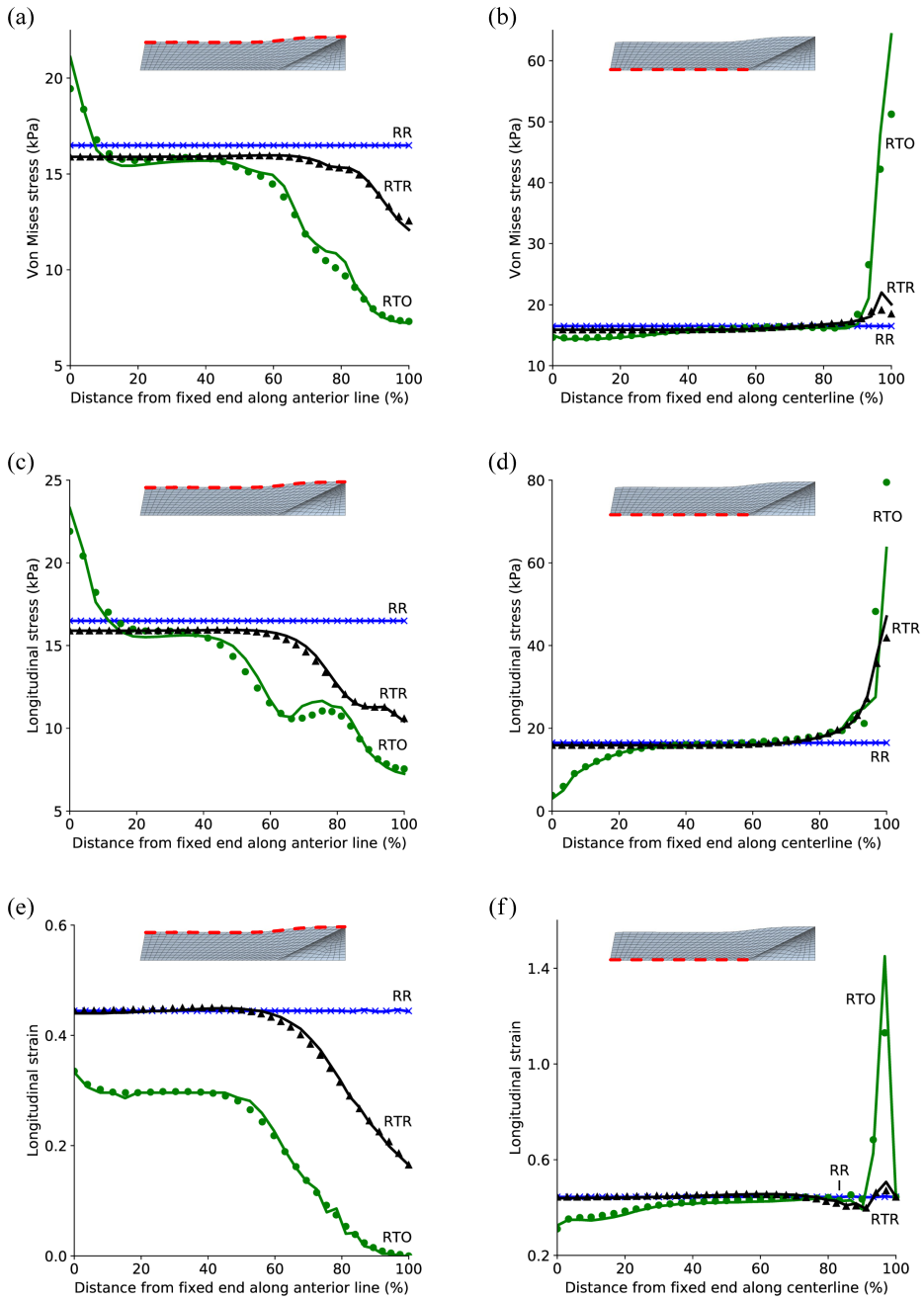

Figure 3.3: Stress and strain from Ansys and FEBio simulations, for elements along the length of the RR, RTR, and RTO models. Von Mises stress along (a) the anterior line and (b) the centerline; longitudinal stress along (c) the anterior line and (d) the centerline; longitudinal strain along (e) the anterior line and (f) the centerline. Results are shown from Ansys using the approximation method (solid lines) and FEBio (markers).

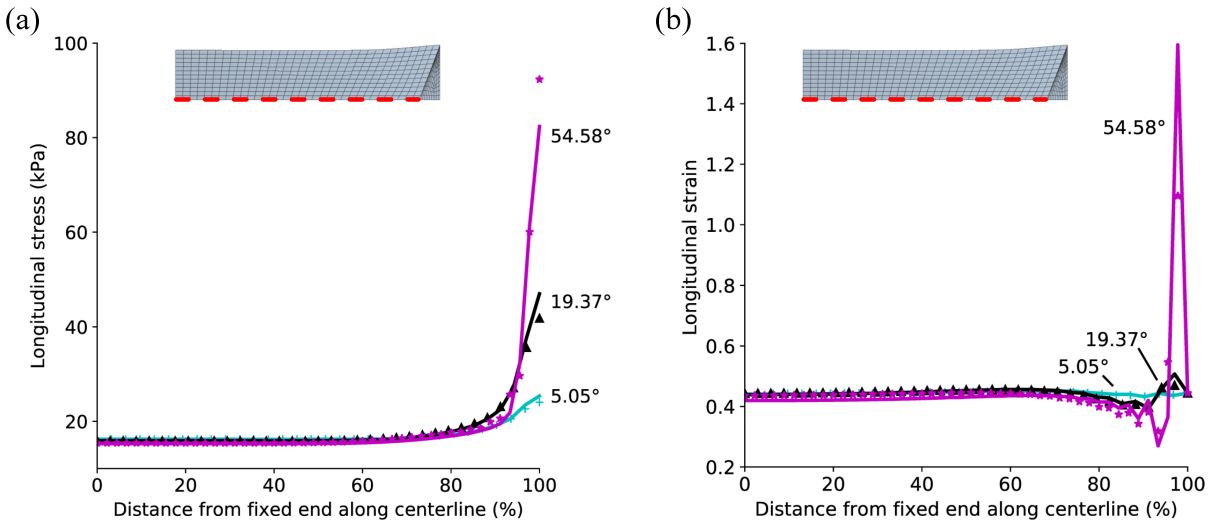

Figure 3.4: Longitudinal stress (a) and longitudinal strain (b) from Ansys and FEBio simulations, for elements along the centerline of the RTR model with three inclination angles (θ, as labeled). Results are shown from Ansys using the approximation method (solid lines) and FEBio (markers).

3.4.3 Simulations of RTR model variants

We created two additional versions of the RTR model to investigate the effect of element skewness along the centerline. The longitudinal stress along the centerline of the RTR model is shown in Fig. 3.4(a) for three values of inclination angle (θ; see Fig. 3.1): 5.05°, 19.37°, and 54.58°. The stress in the region near the fixed end was constant for all three inclination angles. At the free end of the muscle, where the elements had the greatest skewness and were closest to the applied displacement, a sharp increase in the stress was observed. This sharp increase in stress was more pronounced for higher inclination angles in both the Ansys and FEBio simulations. For the inclination angle $\theta = 54.58°$, the stress was 6.0 times larger at the free end (92.3 kPa) than at the fixed end (15.4 kPa) in the FEBio model, and was 5.4 times larger at the free end (82.3 kPa) than at the fixed end (15.2 kPa) in the Ansys model. We observed the greatest difference in stress between the Ansys and FEBio simulations with the $\theta = 54.58°$ RTR model (Table 3.5).

The longitudinal strain along the centerline of the RTR model is shown in Fig. 3.4(b)

Table 3.5: Error in longitudinal stress and longitudinal strain between Ansys and FEBio simulations for the three variants of the RTR model. The maximum relative percent error (maxRPErr) and root-mean-square-percentage error (RMSPE) were calculated over the elements along the centerline.

		RTR ($\theta = 5.05°$)	RTR ($\theta = 19.37°$)	RTR ($\theta = 54.58°$)
Longitudinal	maxRPErr	1.23%	12.35%	14.87%
Stress	RMSPE	0.71%	2.52%	3.87%
Longitudinal	maxRPErr	0.53%	7.87%	45.60%
Strain	RMSPE	0.29%	1.71%	9.37%

for the same three values of inclination angle (θ). The strain in the region near the fixed end was constant, with a value of 0.43 for all three inclination angles, and began decreasing at approximately 67% of the distance along the centerline. A sharp peak in strain was observed near the free end of the muscle for inclination angles of 19.37° and 54.58° in both Ansys and FEBio simulations. The largest peak occurred in the model with the greatest inclination angle, with strain reaching 3.7 and 2.6 times the strain at the fixed end in the Ansys and FEBio simulations, respectively. The difference in strain between the Ansys and FEBio simulations was largest in the $\theta = 54.58°$ RTR model (Table 3.5).

The results shown in Fig. 3.4 support the hypothesis that element skewness contributed to the undesirable behavior observed in longitudinal stress and longitudinal strain near the free end. Recall that high longitudinal strain corresponds to high fiber stretch in this model because the fibers are oriented in the longitudinal direction. Helfenstein et al. [44] studied the behavior of fiber-reinforced, nearly incompressible materials, and noted that nonphysical responses may occur when fiber stretch is high. Specifically, they noted that elements in which the fibers dominate the calculation of total deformation energy will tend to deform in order to reduce the fiber contribution. We observed the same phenomenon in our simulations: when fiber stretch was high, the volumetric component of stress increased to nonphysical levels. As shown in Fig. 3.4, the peaks in stress and non-smooth behavior of strain were exacerbated by increasing element skewness in the RTR model as inclination angle increased.

3.4.4 Sensitivity to the effective bulk modulus

The $\theta = 54.58°$ variant of the RTR model was used to explore how the effective bulk modulus (K) affects the stress and strain computed using our Ansys implementation. As shown in Fig. 3.5, varying K from $10G_1$ to 10^4G_1 had little effect on the longitudinal stress or longitudinal strain along the centerline from the fixed end to approximately 90% of the distance to the free end, in all simulations. In both Ansys and FEBio, the increase in stress toward the free end of the muscle was greatest when $K = 10^4G_1$ (Figs. 3.5(a) and 3.5(c)). In Ansys, the peak stress was 82.3 kPa when $K = 10^4G_1$ and 55.0 kPa when $K = 10G_1$; in FEBio, these peaks were 92.3 kPa and 71.1 kPa, respectively. The non-smooth, oscillatory behavior observed in the strain when $K = 10^4G_1$ was improved when $K = 10G_1$ in both Ansys and FEBio (Figs. 3.5(b) and 3.5(d)). In Ansys, the peak strain was 1.58 when $K = 10^4G_1$ and 1.78 when $K = 10G_1$; in FEBio, these peaks were 1.10 and 1.23, respectively. The results from Ansys and FEBio compared favorably, with maxRPErr less than 1% for stress and less than 4% for strain for the elements along the centerline from the fixed end to 90% of the distance to the free end; greater differences were observed near the free end.

Smaller values of K resulted in a lower peak longitudinal stress and smoother longitudinal strain behavior in the region near the free end. However, one must be cautious when using small values for K because the assumption of near-incompressibility may be violated. In our simulations, the volumetric ratio (J) of the element with largest peak stress and strain was 1.000, 1.002, 1.018, and 1.167 when K was 10^4, 10^3, 10^2, and 10 times the shear moduli ($G_1 = G_2$), respectively. Volumetric ratios that are substantially different from 1 suggest that the solution may be unrealistic. Thus, one must select K judiciously to find a compromise between the competing objectives of avoiding unrealistic peaks in stress and non-smooth strain behavior on one hand while respecting the incompressibility assumption on the other hand.

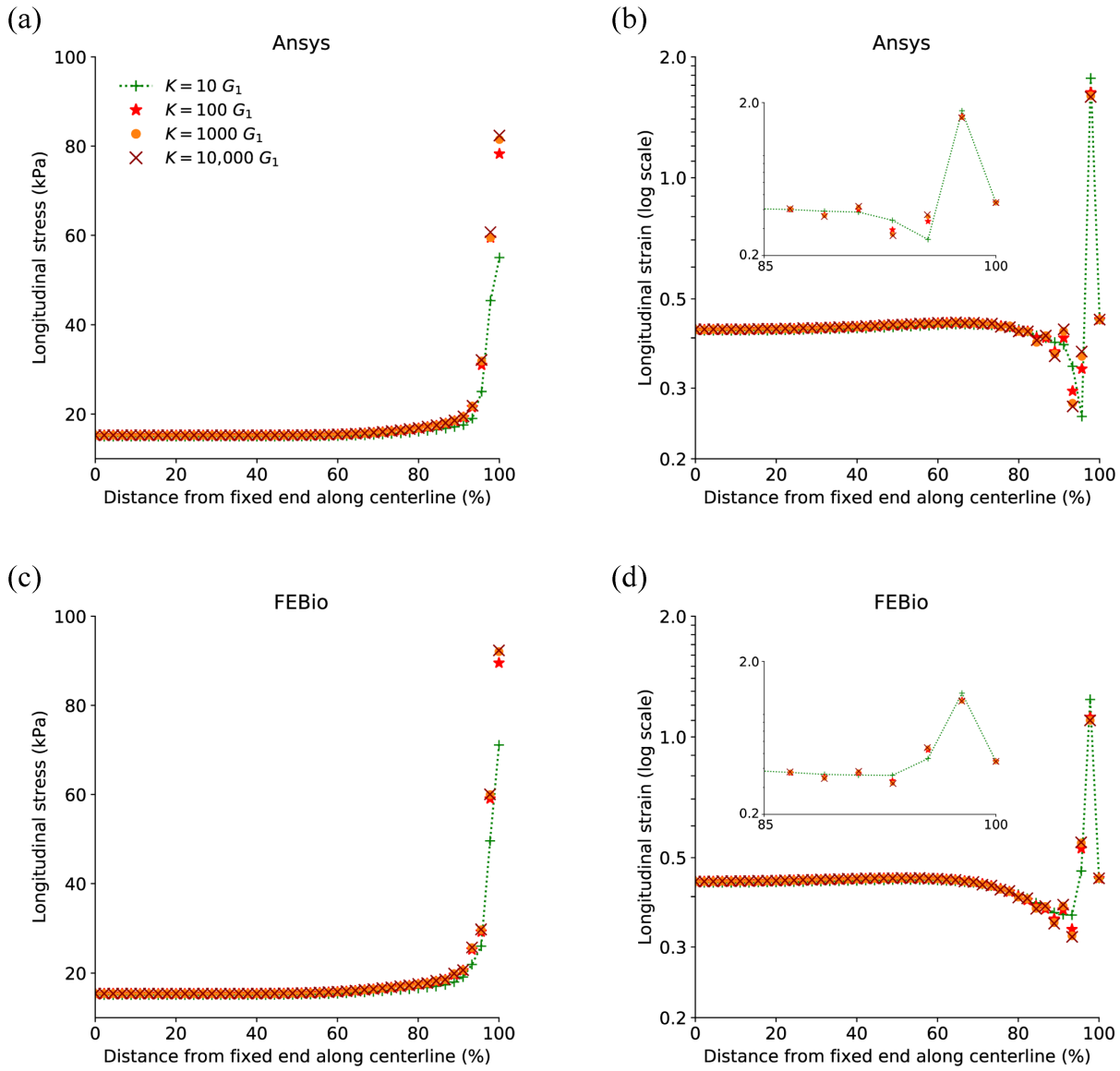

Figure 3.5: Longitudinal stress and longitudinal strain from Ansys ((a) and (b)) and FEBio ((c) and (d)) for elements along the centerline of the RTR model ($\theta = 54.58°$), using four values of parameter K. Insets in panels (b) and (d) magnify the plots near the free end of the muscle.

3.5 Conclusions

This work presents the application of an approximation for the tangent modulus based on the forward difference of the Cauchy stress [1] to a muscle material model [2]. The material model was implemented as an Ansys user-defined material subroutine and validated through comparison to analogous simulations in FEBio, which uses the same muscle model but with the exact value of the tangent modulus. Upon determining an appropriate perturbation parameter for the approximation method ($\epsilon = 10^{-8}$), we observed generally good agreement in stress and strain between Ansys and FEBio across several test scenarios, with some exceptions as noted below. Three test models were constructed by revolving a rectangle (RR), a right trapezoid (RTR), and a generic obtuse trapezoid (RTO) around the muscle's centerline. For the elements along the muscle's centerline, the root-mean-square-percentage error in the Von Mises stress was 0.00%, 3.03%, and 6.75% for the RR, RTR, and RTO models, respectively, in simulations of muscle lengthening; the corresponding errors in longitudinal strain were 0.00%, 1.71%, and 6.07%.

The effect of element skewness was tested by creating two additional versions of the RTR model with different inclination angles at the free end. The stress and strain exhibited the highest error relative to the FEBio simulations when the inclination angle was highest. Sharp increases in longitudinal stress and non-smooth behavior of longitudinal strain were observed along the muscle's centerline in the region near the free end, in both Ansys and FEBio. The largest errors in stress were also observed in this region. Smaller values of K resulted in a lower peak stress and smoother strain behavior; however, selecting a value of K that is too small will violate the assumption of near-incompressibility. Furthermore, small values of K were observed to increase the peak strain and did not entirely eliminate the nonphysical behavior of stress, which increased dramatically in the region near the free end.

The presented approximation for the tangent modulus can greatly facilitate the use of user-defined material models, as it simplifies the derivation as well as the computational

implementation. In particular, programming the calculation of stress in the co-rotational frame is much easier using the approximation method than directly evaluating the exact tangent modulus tensor. In this work, we observed an expected but mild reduction in accuracy when using the approximation method, with the error increasing as element skewness increased. Based on these results, and noting that the approximation method is independent of the specific material model that is used, we recommend applying the approximation to other materials to simplify implementation.

There are two key limitations of this work. First, the test cases we explored use relatively simple geometry. Using simple geometry facilitated thorough analysis and will allow others to easily reproduce our test cases. In future work, more complex geometry will be used to explore the effect of the tangent modulus approximation in more realistic scenarios. Second, we compared the stress and strain in our Ansys simulations to analogous simulations in FEBio. Some of the error in our test cases is likely to have resulted from inherent differences between Ansys and FEBio, such as differences in convergence criteria and nonlinear solver settings, rather than from the approximation of the tangent modulus itself—though we do not expect software-dependent differences to play a substantial role in all conditions since the relative error in stress and strain was approximately 0.00% for the RR model. We have provided our Ansys implementation and usage notes at https://simtk.org/projects/musc_fe_approx so that others can reproduce and extend our results, use the model in their own finite element simulations, and develop new muscle material models in Ansys more quickly.

Bibliography for Chapter 3

[1] W. Sun, E.L. Chaikof, and M.E. Levenston. Numerical approximation of tangent moduli for finite element implementations of nonlinear hyperelastic material models. *Journal of Biomechanical Engineering*, 130(6), 061003, 2008.

[2] S.S. Blemker, P.M. Pinsky, and S.L. Delp. A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38(4), 657–665, 2005.

[3] Y. Wang et al. Finite element analysis of the deformation of deep veins in the lower limb under external compression. *Medical Engineering & Physics*, 35(4), 515–523, 2013.

[4] M.R. Rehorn and S.S. Blemker. The effects of aponeurosis geometry on strain injury susceptibility explored with a 3D muscle model. *Journal of Biomechanics*, 43(13), 2574–2581, 2010.

[5] M. Böl and S. Reese. Micromechanical modelling of skeletal muscles based on the finite element method. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(5), 489–504, 2008.

[6] J. Weickenmeier et al. A physically motivated constitutive model for 3D numerical simulation of skeletal muscles. *International Journal for Numerical Methods in Biomedical Engineering*, 30(5), 545–562, 2014.

[7] O. Röhrle, J.B. Davidson, and A.J. Pullan. Bridging scales: a three-dimensional electromechanical finite element model of skeletal muscle. *SIAM Journal on Scientific Computing*, 30(6), 2882–2904, 2008.

[8] C.W.J. Oomens et al. Finite element modelling of contracting skeletal muscle. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 358(1437), 1453–1460, 2003.

[9] T.T. Dao and M.-C.H.B. Tho. A systematic review of continuum modeling of skeletal muscles: current trends, limitations, and recommendations. *Applied Bionics and Biomechanics*, 2018, 7631818, 2018.

[10] A.V. Hill. The heat of shortening and the dynamic constants of muscle. *Proceedings of the Royal Society B: Biological Sciences*, 126(843), 136–195, 1938.

[11] T.K. Uchida and S.L. Delp. Biomechanics of Movement: The Science of Sports, Robotics, and Rehabilitation. Cambridge, MA, USA: The MIT Press, 2021.

[12] B. Hernández-Gascón et al. A 3D electro-mechanical continuum model for simulating skeletal muscle contraction. *Journal of Theoretical Biology*, 335, 108–118, 2013.

[13] M. Böl, R. Weikert, and C. Weichert. A coupled electromechanical model for the excitation-dependent contraction of skeletal muscle. *Journal of the Mechanical Behavior of Biomedical Materials*, 4(7), 1299–1310, 2011.

[14] A.F. Huxley. Muscle structure and theories of contraction. *Progress in Biophysics and Biophysical Chemistry*, 7, 255–318, 1957.

[15] L. Beldie et al. Finite element modelling of maxillofacial surgery and facial expressions – a preliminary study. *The International Journal of Medical Robotics and Computer Assisted Surgery*, 6(4), 422–430, 2010.

[16] B. Calvo et al. Passive nonlinear elastic behaviour of skeletal muscle: experimental results and model formulation. *Journal of Biomechanics*, 43(2), 318–325, 2010.

[17] A.E. Ehret and M. Itskov. A polyconvex hyperelastic model for fiber-reinforced materials in application to soft tissues. *Journal of Materials Science*, 42(21), 8853–8863, 2007.

[18] S. Hedenstierna and P. Halldin. How does a three-dimensional continuum muscle model affect the kinematics and muscle strains of a finite element neck model compared to a discrete muscle model in rear-end, frontal, and lateral impacts. *Spine*, 33(8), E236–E245, 2008.

[19] A.M. Zöllner et al. On high heels and short muscles: a multiscale model for sarcomere loss in the gastrocnemius muscle. *Journal of Theoretical Biology*, 365, 301–310, 2015.

[20] P. Büchler et al. A finite element model of the shoulder: application to the comparison of normal and osteoarthritic joints. *Clinical Biomechanics*, 17(9–10), 630–639, 2002.

[21] G.G. Barbarino et al. Development and validation of a three-dimensional finite element model of the face. *Journal of Biomechanical Engineering*, 131(4), 041006, 2009.

[22] A.E. Ehret, M. Böl, and M. Itskov. A continuum constitutive model for the active behaviour of skeletal muscle. *Journal of the Mechanics and Physics of Solids*, 59(3), 625–636, 2011.

[23] J. Grasa et al. A 3D active-passive numerical skeletal muscle model incorporating initial tissue strains. Validation with experimental results on rat tibialis anterior muscle. *Biomechanics and Modeling in Mechanobiology*, 10(5), 779–787, 2011.

[24] J. Li et al. Development of a finite element musculoskeletal model with the ability to predict contractions of three-dimensional muscles. *Journal of Biomechanics*, 94, 230–234, 2019.

[25] C.Y. Tang, G. Zhang, and C.P. Tsui. A 3D skeletal muscle model coupled with active contraction of muscle fibres and hyperelastic behaviour. *Journal of Biomechanics*, 42(7), 865–872, 2009.

[26] F. Li et al. Simulation of muscle activation with coupled nonlinear FE models. *Journal of Mechanics in Medicine and Biology*, 16(6), 1650082, 2016.

[27] S.A. Maas et al. FEBio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering*, 134(1), 011005, 2012.

[28] S.A. Maas, G.A. Ateshian, and J.A. Weiss. FEBio: history and advances. *Annual Review of Biomedical Engineering*, 19, 279–299, 2017.

[29] D. Scherb, S. Wartzack, and J. Miehling. Modelling the interaction between wearable assistive devices and digital human models—a systematic review. *Frontiers in Bioengineering and Biotechnology*, 10, 1044275, 2023.

[30] M.B. Yandell et al. Physical interface dynamics alter how robotic exosuits augment human movement: implications for optimizing wearable assistive devices. *Journal of Neuroengineering and Rehabilitation*, 14(1), 40, 2017.

[31] G.A. Holzapfel. Nonlinear Solid Mechanics: A Continuum Approach for Engineering. Chichester, England: John Wiley & Sons, 2000.

[32] J. Cheng and L.T. Zhang. A general approach to derive stress and elasticity tensors for hyperelastic isotropic and anisotropic biomaterials. *International Journal of Computational Methods*, 15(4), 1850028, 2018.

[33] ANSYS, Inc. ANSYS Mechanical APDL Technology Demonstration Guide, Release 15.0. Tech. rep. Canonsburg, PA, USA, 2013.

[34] C. Miehe. Numerical computation of algorithmic (consistent) tangent moduli in large-strain computational inelasticity. *Computer Methods in Applied Mechanics and Engineering*, 134(3–4), 223–240, 1996.

[35] M. Tanaka and M. Fujikawa. Numerical approximation of consistent tangent moduli using complex-step derivative and its application to finite deformation problems. *Transactions of the Japan Society of Mechanical Engineers. A*, 77(773), 27–38, 2011.

[36] C. Suchocki. A finite element implementation of Knowles stored-energy function: theory, coding and applications. *The Archive of Mechanical Engineering*, 58(3), 319–346, 2011.

[37] T.C. Gasser, R.W. Ogden, and G.A. Holzapfel. Hyperelastic modelling of arterial layers with distributed collagen fibre orientations. *Journal of the Royal Society Interface*, 3(6), 15–35, 2006.

[38] H. Liu and W. Sun. Numerical approximation of elasticity tensor associated with Green-Naghdi rate. *Journal of Biomechanical Engineering*, 139(8), 081007, 2017.

[39] H. Fehervary et al. How to implement user-defined fiber-reinforced hyperelastic materials in finite element software. *Journal of the Mechanical Behavior of Biomedical Materials*, 110, 103737, 2020.

[40] H. Liu and W. Sun. Computational efficiency of numerical approximations of tangent moduli for finite element implementation of a fiber-reinforced hyperelastic material model. *Computer Methods in Biomechanics and Biomedical Engineering*, 19(11), 1171–1180, 2016.

[41] ANSYS, Inc. ANSYS Meshing User's Guide, Release 13.0. Tech. rep. Canonsburg, PA, USA, 2010.

[42] N.M. Fiorentino and S.S. Blemker. Musculotendon variability influences tissue strains experienced by the biceps femoris long head muscle during high-speed running. *Journal of Biomechanics*, 47(13), 3325–3333, 2014.

[43] W. Ehlers and G. Eipper. The simple tension problem at large volumetric strains computed from finite hyperelastic material laws. *Acta Mechanica*, 130(1), 17–27, 1998.

[44] J. Helfenstein et al. On non-physical response in models for fiber-reinforced hyperelastic materials. *International Journal of Solids and Structures*, 47(16), 2056–2061, 2010.

4

Phenomenological Muscle Constitutive Model With Actin–Titin Binding for Simulating Active Stretching

The second main contribution of this thesis is the development of a new phenomenological muscle material model that exhibits force enhancement. This work is an important contribution in the research area because it enables the development of finite element models that consider muscle contraction, including our own models for studying the residuum–socket interface (Chapter 5). Our muscle material model is one of the few models available that exhibits force enhancement, which we implemented by considering the interaction between the actin and titin filaments. Including force enhancement in the material model improved the accuracy of the predictions of fiber force while also improving numerical stability.

We addressed the shortcomings of existing muscle constitutive models by ensuring numerical stability during active stretching simulations. We explored two approaches. Initially, we attempted to address the numerical instability of the muscle constitutive model

proposed by Blemker et al. (2005) by adopting the theory of sarcomere non-uniformity. Following this theory, we represented the fiber length during active stretching simulations as a weighted average of up to three different fiber lengths, where each of these lengths yielded the same force according to the force–length curve. (Because of the negative stiffness region, there is a range of total fiber forces that can be generated by fibers of three different lengths.) The coefficients used in the weighted average calculation were defined such that the eigenvalues of the spatial elasticity tensor were always greater than zero; however, the resulting fiber lengths were unrealistic. Following this investigation, we decided to include force enhancement into our constitutive model. Our model incorporates actin–titin binding and calcium-induced stiffness changes in the titin filament, which occur during active stretching. We developed a transversely isotropic hyperelastic material model by improving an existing muscle model [1]. Specifically, we replaced the active stress–stretch curve with an asymmetric Weibull curve and we adopted a passive stress–stretch curve derived from experiments of titin filament stretch, which exhibits force enhancement.

We tested our model by evaluating its numerical stability and validating the results against experimental data reported in the literature. The numerical stability was tested using a cubic model during a series of uniaxial stretches; the simulations demonstrated that the proposed material model is numerically stable and the spatial elasticity tensor maintains positive eigenvalues during active stretching. A 3D benchmark model was used to validate the results of our proposed muscle model. The stress produced during active stretching was greater than the stress produced isometrically at the same fiber length, which is consistent with experiments. Also, the fiber length distribution from the 3D benchmark model improved agreement with measurements of fiber lengths during experiments of active stretching, compared to the distribution predicted by the material model proposed by Blemker et al. [1].

We implemented our model in FEBio (and, later, in Ansys) and demonstrated its effectiveness in simulations where active stretching occurs. Our model could be used

to study the interactions between muscle and the surrounding tissue during eccentric contractions, or to evaluate prosthetic and orthotic devices under realistic muscle activation conditions. Our proposed model also demonstrated numerical stability, which can improve robustness in finite element simulations.

We presented this work at the Canadian Society of Mechanical Engineering International Congress (Sampaio de Oliveira, M.L., Uchida, T.K. Muscle constitutive model with actin–titin binding demonstrates force enhancement, Canadian Society of Mechanical Engineering International Congress, May 26–29, 2024, Toronto, Canada). We published this work in the ASME Journal of Biomechanical Engineering (Sampaio de Oliveira, M.L., Uchida, T.K. Phenomenological muscle constitutive model with actin–titin binding for simulating active stretching. ASME Journal of Biomechanical Engineering, 147(1): 011002, 2025, doi: 10.1115/1.4066564). The content of this paper is provided in the remainder of this chapter.

4.1 Abstract

The force produced by a muscle depends on its contractile history, yet human movement simulations typically employ muscle models that define the force–length relationship from measurements of fiber force during isometric contractions. In these muscle models, the total force–length curve can have a negative slope at fiber lengths greater than the fiber length at which peak isometric force is produced. This region of negative stiffness can cause numerical instability in simulations. Experiments have found that the steady-state force in a muscle fiber following active stretching is greater than the force produced during a purely isometric contraction. This behaviour is called residual force enhancement. We present a constitutive model that exhibits force enhancement, implemented as a hyperelastic material in the FEBio finite element software. There is no consensus on the mechanisms responsible for force enhancement; we adopt the assumption that the passive fiber force depends on the sarcomere length at the instant that the muscle is activated above a threshold. We demonstrate the numerical stability of our model using

an eigenvalue analysis and by simulating a muscle whose fibers are of different lengths. We then use a 3D muscle geometry to verify the effect of force enhancement on the development of stress and the distribution of fiber lengths. Our proposed muscle material model is one of the few models available that exhibits force enhancement and is suitable for simulations of active lengthening. We provide our implementation in FEBio so that others can reproduce and extend our results.

Keywords: elasticity tensor; finite element; force enhancement; skeletal muscle; titin; worm-like chain.

4.2 Introduction

Finite element simulations of skeletal muscles have provided insight into the mechanical interaction between muscle and internal structures such as soft tissues and bones [2], as well as external structures such as prostheses [3, 4]. However, most finite element simulations use muscle material models that generate only passive force [5] or that generate active force based on experimental measurements of purely isometric contractions (i.e., muscle fibers that are activated while being held at a constant length) [6]. These modelling simplifications may affect answers to research questions that are sensitive to muscle behaviour. Furthermore, muscle material models have often been used in finite element simulations without first establishing that they are numerically stable [7]. Biomechanists generating finite element simulations would benefit from muscle material models that are both numerically stable and can be used in situations where muscles are not acting purely isometrically.

Phenomenological muscle models fit equations to experimental data without attempting to describe the biological mechanisms responsible for the observed behaviour. Hill developed the first phenomenological muscle model from experiments that characterized the relationship between the shortening velocity of muscle fibers and the force they

produce. This model was later extended into the popular "Hill-type" muscle model commonly used today [8]. In addition to shortening velocity, the force produced by a muscle also depends on its activation, the length of its fibers, and other factors that are often ignored in muscle models, such as the temperature and the contractile history of the muscle [2, 9]. Active force is produced only when activation is greater than zero. The maximum possible active force is typically a function of only the length and velocity of the fibers, and this value is scaled linearly with the activation level. Passive force is produced whenever the fibers are stretched beyond a particular length, regardless of activation, and increases as the fibers lengthen [10]. Many muscle models compute the total generated force as the sum of active and passive components, which is called the active-stress approach in the context of continuum mechanics; other models employ the active-strain approach, where the deformation gradient tensor is decomposed into active strain and elastic strain components to account for, respectively, active distortion and the storage of elastic energy [11, 12].

Sarcomeres are the smallest functional units of skeletal muscle. Sarcomeres are arranged in series to form myofibrils, myofibrils are arranged in parallel to form muscle fibers, muscle fibers are arranged in bundles called muscle fascicles, and muscle fascicles pack together to form whole muscle. Because whole muscle is comprised of sarcomeres arranged in series and in parallel, the active force–length (FL) curve used in muscle models is often derived from the active FL curve of a single sarcomere—specifically, the latter is scaled up in force and length to derive the former. Another approach is to use the FL curves measured from whole muscle with a model comprised of series and parallel elastic and contractile elements to identify the active and passive FL curves [13]. A sarcomere's active FL curve has three regions: the ascending limb, where the force increases with increasing sarcomere length; the plateau region, where the force is at its maximum over a small range of sarcomere lengths; and the descending limb, where the force decreases as the sarcomere lengthens (i.e., the curve has a negative slope) [8]. In Hill-type models, although the passive FL curve has a positive slope when the fibers are stretched, the

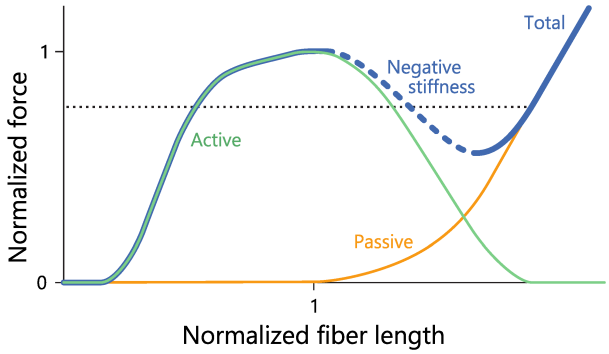

Figure 4.1: Typical shapes of force–length curves used in Hill-type models. The sum of the active and passive curves produces a total curve with a region of negative stiffness, which can cause numerical instability. Further, a fiber may generate a given total force at up to three distinct lengths (dotted horizontal line).

negative slope of the active FL curve is steeper from approximately 1.1 to 1.5 times the optimal fiber length (the length at which the fiber can generate maximum isometric active force) [7]; thus, the total FL curve has a negative slope over a range of fiber lengths (Fig. 4.1). This region of the total FL curve is referred to as the negative stiffness region [14, 15]. Unlike most elastic objects with which we are familiar, a material operating in a region of negative stiffness generates less force as it lengthens. In finite element simulations in which a muscle is active and lengthening, negative stiffness corresponds to a negative change in strain energy [16], which makes the material model numerically unstable.

Points on the total FL curve can be determined experimentally by passively stretching a muscle to a desired length, holding it at this length as it is maximally activated, and measuring the generated force. The total FL curve is obtained by repeating this experiment over a range of fiber lengths. This method assumes that the relationship between active force and length that is observed when the muscle is held at a constant length (isometric) remains valid when the muscle is active and shortening (concentric contraction) and when it is active and lengthening (eccentric contraction). However, the forces measured during active stretching experiments always indicate that the muscle has positive stiffness, and the steady-state force measured following an active stretch is

always greater than the force measured at the same length but where the muscle had been stretched passively and then activated.

During active stretching, the increase in force that is observed is called force enhancement [17]. Following an active stretch, the increase in steady-state force is called residual force enhancement and has been observed at all structural levels of a muscle: in single sarcomeres [18, 19], myofibrils [20, 21], single fibers [22, 23], and whole muscle [24, 25]. The steady-state force following an active stretch depends on the length of the fibers at the onset of activation [25, 26] but is insensitive to the stretch velocity [24, 25, 27]. The observed increase in force following an active stretch is greater on the descending limb of the active FL curve than on the ascending limb [28]. Physiologically, force enhancement enables our muscles to generate and sustain high forces during active stretching [23, 29], which protects them from injury due to overstretching. Mechanically, force enhancement provides stability when our muscles are operating on the descending limb of the active FL curve [17]. Although the macroscopic effects of force enhancement are well characterized, there is no consensus on the underlying mechanisms that are responsible [30].

One of the first attempts to explain force enhancement was the sarcomere length non-uniformity hypothesis [14]. During active stretching of a myofibril on the descending limb of the active FL curve, the hypothesis was that some sarcomeres will stretch more than others, resulting in non-uniform sarcomere lengths. If the average sarcomere length during this active stretch experiment is equal to the average length measured in an analogous isometric experiment, where all the sarcomeres are of equal length, the force produced during the active stretch experiment will be greater due to the contribution of the passive forces in the longer-than-average sarcomeres [15]. However, this theory does not explain why residual force enhancement can be observed in a single sarcomere [19].

Another possible explanation for residual force enhancement is that the stiffness of the structural protein titin changes during active stretching. This hypothesis is consistent with experimental observations. In skeletal muscle, titin comprises two immunoglobulin (Ig) domains (proximal and distal regions, which are adjacent to the Z-disc and myosin,

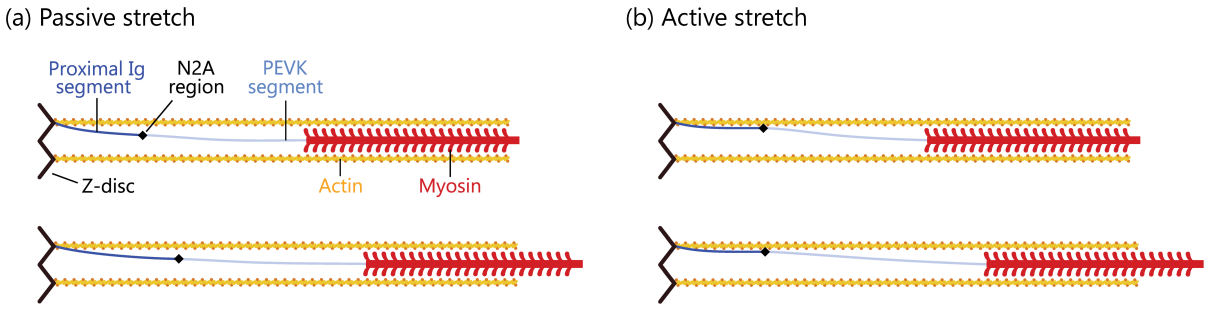

Figure 4.2: Model of a half-sarcomere during passive and active stretches. (a) When stretched passively, the titin and actin filaments do not interact and the titin acts as a nonlinear spring. (b) It has been hypothesized that the N2A region binds to the actin filament when a muscle is activated; thus, during active stretching, only the PEVK segment of titin changes length.

respectively); a PEVK segment, which is a region rich in the amino acids proline (P), glutamate (E), valine (V), and lysine (K); and the N2A region [31], which connects the proximal Ig domain and PEVK segment [32] (Fig. 4.2). Titin acts as a molecular spring that connects the myosin filament to the Z-disc, and its stiffness can change in two ways. First, when muscle is activated, calcium ions bind to specific sites on the PEVK segment which increases the inherent stiffness of titin [21, 33, 34]. Second, it is believed that titin binds to the actin filament when the muscle becomes activated, thereby decreasing the free length of the titin [35].

How to model the actin–titin binding remains a topic of debate. Rode et al. [36] proposed a "sticky-spring" mechanism in which the PEVK segment attaches to actin as a piece of adhesive tape would attach to a flat surface. Nishikawa et al. [37] proposed the "winding filament" hypothesis, where the titin binds to actin at the N2A region and the PEVK segment winds around the actin filament. Schappacher-Tilp et al. [38] also hypothesized that the titin binds to actin at the N2A region, but their model does not consider any interaction between the PEVK segment and the actin filament (Fig. 4.2(b)). Millard et al. [39] proposed a viscoelastic actin–titin model where the binding site is located within the PEVK segment. It is likely that titin binds to actin; however, further study is required to determine how and where the binding occurs [40].

Although residual force enhancement was first observed over 70 years ago [24, 41], most simulations using finite element methods neglect this effect. The first phenomenological constitutive model that attempted to consider force enhancement replaced the nonlinear active FL curve with a linear relationship [42]. This model did not suffer from instability on the descending limb of the active FL curve, but it produced unrealistic behaviour. Seydewitz et al. [43] proposed a phenomenological model to simulate force enhancement that considers both velocity and history-dependent effects; however, this model does not consider the length of the fiber at the onset of activation. Heidlauf et al. [44] proposed a model that demonstrates force enhancement, but the forces produced by actin–titin interaction were not validated. In the context of continuum mechanics, muscles are often modelled as nonlinear or "hyperelastic" materials [2]. Hyperelastic materials are described by a strain energy function that defines the relationship between stress and strain [45]. Muscles are unique in that the forces they produce are directed along the fibers.

In this work, we propose a transversely isotropic, hyperelastic muscle material model that exhibits force enhancement and has a positive definite spatial elasticity tensor so that it is numerically stable. We model the fiber behaviour at the microscopic level and consider the other components of the muscle tissue at the macroscopic level by adding the along- and cross-fiber shear behaviour to the strain energy function. The active and passive FL curves are obtained from experimental data that were collected from a rabbit psoas myofibril [26]. The actin–titin interaction is modelled using the model of force enhancement from Schappacher-Tilp et al. [38], which adjusts the passive FL curve depending on the fiber length at the onset of activation. The numerical stability of our model is demonstrated by observing that the spatial elasticity tensor remains positive definite and by simulating a muscle filament in which each element has a slightly different fiber length, the latter being a situation that can elicit numerical instability. Furthermore, we perform a benchmark simulation using a 3D muscle geometry undergoing an active stretch and show that our model predicts fiber lengths whose

distribution is similar to the measurements reported by Moo et al. [46]. We provide our material model implementation in the open-source finite element software FEBio [47] at https://simtk.org/projects/muscle_fe_titin so that others can reproduce and extend our results.

4.3 Methodology

To capture the force enhancement behaviour observed at the sarcomere level, we adopted the "three-filament" model of titin proposed by Schappacher-Tilp et al. [38]. In this section, we describe the three-filament model as well as our modifications to achieve two objectives: (1) avoid running Monte Carlo simulations as was done by Schappacher-Tilp et al. [38], and (2) improve agreement with experiments on actively stretched myofibrils [26]. We also describe our use of optimization to calculate the parameters for the passive FL curves in the three-filament model. Finally, we present our proposed constitutive model, which extends the model proposed by Blemker et al. [1] by introducing the force enhancement behaviour.

4.3.1 Micromechanical model of actin–titin binding

Titin is a protein that can be modelled as three serially connected segments: a proximal immunoglobulin (Ig) segment that is attached to the Z-disc, a distal Ig segment that is attached to the myosin filament, and a PEVK segment that joins the two Ig segments [48]. Each Ig segment is comprised of serially linked Ig domains. Each Ig domain is a self-stable region in the protein chain that can undergo conformational changes (i.e., folding or unfolding) independently of the rest of the chain.

Worm-like chain (WLC) models are used in the three-filament model of titin to simulate the proximal Ig and PEVK segments. (The third filament, the distal Ig segment, is relatively inelastic and is, therefore, modelled as being rigid.) WLC models have been used to simulate the elastic behaviour of polymers [49, 50] as well as the titin filament [35, 38]. The relationship between the applied force (F_{Ig}) and elongation (x_{Ig}) of one

domain in the proximal Ig segment is approximated from a WLC model as follows:

$$F_{\mathrm{Ig}} = \frac{k_B T}{p_{\mathrm{Ig}}} \left(\frac{1}{4\left(1 - \frac{x_{\mathrm{Ig}}}{L_{\mathrm{Ig}}}\right)^2} - \frac{1}{4} + \frac{x_{\mathrm{Ig}}}{L_{\mathrm{Ig}}} \right) \tag{4.1}$$

where k_B is the Boltzmann constant, T is the absolute temperature, p_{Ig} is the persistence length (quantifying the bending stiffness) of one Ig domain [48], and L_{Ig} is the contour length of one Ig domain (i.e., the theoretical end-to-end length of a single Ig domain when stretched with infinite force, assuming there is no fracture or change in the structure of the chain). To model the PEVK segment, we used a variation of the WLC model with one additional parameter. Specifically, we followed the approach of Linke et al. [51], who introduced a stretch modulus parameter (K_0) to improve agreement with measurements of force (F_{PEVK}) and elongation (x_{PEVK}) of the PEVK segment:

$$F_{\mathrm{PEVK}} = \frac{k_B T}{p_{\mathrm{PEVK}}} \left(\frac{1}{4\left(1 - \frac{x_{\mathrm{PEVK}}}{L_{\mathrm{PEVK}}} + \frac{F_{\mathrm{PEVK}}}{K_0}\right)^2} - \frac{1}{4} + \frac{x_{\mathrm{PEVK}}}{L_{\mathrm{PEVK}}} - \frac{F_{\mathrm{PEVK}}}{K_0} \right) \tag{4.2}$$

where p_{PEVK} and L_{PEVK} are, respectively, the persistence length and contour length of the PEVK segment.

Given the length of each segment of titin, the half-sarcomere length (ℓ_{HS}) can be calculated as follows:

$$\ell_{\mathrm{HS}} = N^{\mathrm{prox}} \ell_{\mathrm{Ig}}^{\mathrm{prox}} + \ell_{\mathrm{PEVK}} + \ell_{\mathrm{Ig}}^{\mathrm{dist}} + d \tag{4.3}$$

where N^{prox} is the number of proximal Ig domains (i.e., the number of Ig domains comprising the proximal Ig segment), $\ell_{\mathrm{Ig}}^{\mathrm{prox}}$ is the length of each proximal Ig domain, ℓ_{PEVK} is the length of the PEVK segment, $\ell_{\mathrm{Ig}}^{\mathrm{dist}}$ is the length of the distal Ig segment, and d is half the width of the A-band region. Both the distal Ig segment and the A-band region are assumed to be inextensible [38].

Proximal Ig domains transit between two energetic states that have different lengths. The transition from the low-energy (folded) state to the high-energy (unfolded) state

is called unfolding. The number of unfolded proximal Ig domains depends on the force applied to the proximal Ig segment (F_{Ig}). When under high tension, proximal Ig domains from rabbit psoas muscle unfold at an approximate sarcomere length of 3.6 µm [52]. Unfolding increases the length of a proximal Ig domain by 26 nm [38]. Therefore, for sarcomere lengths greater than 3.6 µm, Eq. (4.3) was adapted to account for the number of folded and unfolded proximal Ig domains:

$$\ell_{\mathrm{HS}} = N_u^{\mathrm{prox}} \ell_{\mathrm{Ig,u}}^{\mathrm{prox}} + N_f^{\mathrm{prox}} \ell_{\mathrm{Ig,f}}^{\mathrm{prox}} + \ell_{\mathrm{PEVK}} + \ell_{\mathrm{Ig}}^{\mathrm{dist}} + d \tag{4.4}$$

where N_u^{prox} and N_f^{prox} are the number of unfolded and folded proximal Ig domains, and $\ell_{\mathrm{Ig,u}}^{\mathrm{prox}}$ and $\ell_{\mathrm{Ig,f}}^{\mathrm{prox}}$ are the length of one proximal Ig domain when unfolded and when folded, respectively.

The statistical folding and unfolding of Ig domains can be represented by a Monte Carlo simulation [50, 53]; however, running Monte Carlo simulations and averaging the predicted forces would incur substantial computational expense if implemented within a muscle material model. Instead, we calculated the number of unfolded and folded domains in the proximal Ig segment as follows:

$$N_u^{\mathrm{prox}} = P_u N^{\mathrm{prox}} \tag{4.5a}$$

$$N_f^{\mathrm{prox}} = (1 - P_u)\, N^{\mathrm{prox}} \tag{4.5b}$$

where P_u is the probability of unfolding when a force F_{Ig} is applied. This probability can be calculated as follows [54]:

$$P_u = \frac{\alpha\left(F_{\mathrm{Ig}}\right)}{\alpha\left(F_{\mathrm{Ig}}\right) + \beta\left(F_{\mathrm{Ig}}\right)} \tag{4.6}$$

where $\alpha\left(F_{\mathrm{Ig}}\right)$ is the unfolding rate and $\beta\left(F_{\mathrm{Ig}}\right)$ is the folding rate. The unfolding and

folding rates can be calculated as follows:

$$\alpha\left(F_{\mathrm{Ig}}\right)=\alpha_0 \exp\left(\frac{F_{\mathrm{Ig}} b_u}{k_B T}\right) \tag{4.7a}$$

$$\beta\left(F_{\mathrm{Ig}}\right)=\beta_0 \exp\left(\frac{F_{\mathrm{Ig}} b_f}{k_B T}\right) \tag{4.7b}$$

where α_0 and β_0 are the unfolding and folding rates at zero force, and b_u and b_f are the width of the activation barrier for an unfolding or folding event to occur [50].

4.3.2 Parameters for PEVK segment during active stretching

During active stretching, it has been suggested that titin binds to actin, thereby reducing the free length of the titin and increasing the force it generates when elongated [30, 40]. We assumed that each titin filament binds to the nearby actin filament at the N2A region of the titin, and that this binding occurs at the instant that the muscle is activated above a threshold [38]. Once the titin has bound to the actin, only the PEVK segment will elongate when the titin is stretched.

To calculate the elongation of the PEVK segment during active stretching, we must first determine the elongation of one proximal Ig domain at the onset of activation. We can then compute the length of the portion of the titin filament that remains fixed relative to the actin (Fig. 4.2(b)). The length of the sarcomere at this instant will be referred to as the initial sarcomere length (ISL). To compute the elongation of one proximal Ig domain at this instant ($x_{\mathrm{Ig}}^{\mathrm{ISL}}$), we solve Eqs. (4.1), (4.2), and (4.3) at the ISL if the sarcomere is shorter than 3.6 µm; otherwise, we solve Eqs. (4.1), (4.2), and (4.4) at the ISL (see Sect. 4.3.1). The elongation of the PEVK segment is then given as follows:

$$x_{\mathrm{PEVK}}=x_{\mathrm{titin}}-N^{\mathrm{prox}} x_{\mathrm{Ig}}^{\mathrm{ISL}} \tag{4.8}$$

The force produced by titin is then calculated by substituting x_{PEVK} into Eq. (4.2).

The parameters defining the behaviour of the PEVK segment were assumed to be

functions of the ISL. This assumption enabled a closer match between our simulation results and the experimental data collected from a myofibril stretch [26]. This modelling assumption can be justified by noting two sources of uncertainty: (1) how the stiffness of titin changes upon activation, and (2) the exact locations of the binding sites between actin and titin. We varied the persistence length of the PEVK segment (p_{PEVK}) and the stretch modulus parameter (K_0) linearly depending on the ISL:

$$p_{\mathrm{PEVK,ISL}} = (p_{\mathrm{PEVK,active}} - p_{\mathrm{PEVK,passive}}) \frac{\ell_{\mathrm{sar}}^{\mathrm{max-act}} - \mathrm{ISL}}{\ell_{\mathrm{sar}}^{\mathrm{max-act}} - \ell_{\mathrm{sar}}^{\mathrm{opt}}} + p_{\mathrm{PEVK,passive}} \tag{4.9a}$$

$$K_{0,\mathrm{ISL}} = (K_{0,\mathrm{active}} - K_{0,\mathrm{passive}}) \frac{\ell_{\mathrm{sar}}^{\mathrm{max-act}} - \mathrm{ISL}}{\ell_{\mathrm{sar}}^{\mathrm{max-act}} - \ell_{\mathrm{sar}}^{\mathrm{opt}}} + K_{0,\mathrm{passive}} \tag{4.9b}$$

where $\ell_{\mathrm{sar}}^{\mathrm{max-act}}$ is the sarcomere length at which the descending limb of the active FL curve reaches zero force and $\ell_{\mathrm{sar}}^{\mathrm{opt}}$ is the optimal sarcomere length (the length at which the sarcomere can generate maximum isometric active force). The subscripts "active" and "passive" refer to the parameters corresponding to the active and passive states. For stretches starting from sarcomere lengths greater than $\ell_{\mathrm{sar}}^{\mathrm{max-act}}$, the passive parameters are used. Consequently, in our model, there is no difference between active and passive stretches starting from sarcomere lengths greater than $\ell_{\mathrm{sar}}^{\mathrm{max-act}}$, following the conclusions of Leonard and Herzog [26].

4.3.3 Parameters for worm-like chain model

We used parameters derived from experimental measurements reported in the literature. We assumed $N^{\mathrm{prox}} = 50$ domains in the proximal Ig segment and $N^{\mathrm{PEVKr}} = 800$ PEVK residues (repeated units in the polymeric chain) [55]. The contour length of one Ig domain was $L_{\mathrm{Ig}} = 4.5$ nm and the contour length of one PEVK residue was $L_{\mathrm{PEVKr}} = 0.36$ nm [55]. The total contour length of the PEVK segment (L_{PEVK}) was computed as follows:

$$L_{\mathrm{PEVK}} = N^{\mathrm{PEVKr}} L_{\mathrm{PEVKr}} \tag{4.10}$$

We used different values for the PEVK segment parameters during active and passive stretching. This was required because the titin stiffness increases due to binding of calcium ions to the PEVK segment when the muscle is activated [33]. We used experimental data from a passively stretched myofibril [26] to estimate the persistence lengths for the proximal Ig and PEVK segments, as well as the stretch modulus for the PEVK segment. These parameters were used when the muscle was stretching passively. Experiments on actively stretched myofibrils indicate that the passive force increases by approximately 25% due to calcium-induced titin stiffness [56]. Therefore, we assumed an increase of 25% in the passive force to estimate the persistence length and stretch modulus for the PEVK segment during active stretching.

The passive FL curve can be divided into three intervals based on the sarcomere length. When the sarcomeres are shorter than their optimal length of 2.4 µm, no passive force is generated. For sarcomere lengths between 2.4 µm and 3.6 µm, the passive force–length relationship is described by a system of three nonlinear equations (Eqs. (4.1), (4.2), and (4.3)). This system of equations can be solved to determine the passive force at a given sarcomere length. We assumed that, during stretching (whether passive or active), proximal Ig domains begin to unfold at sarcomere lengths greater than 3.6 µm. Thus, for sarcomere lengths greater than 3.6 µm, the passive force–length relationship is described by a different system of nonlinear equations (Eqs. (4.1), (4.2), and (4.4)) to account for the difference in the persistence length of an Ig domain when in its folded and unfolded states. By solving these systems of equations, we obtained the forces generated by the PEVK and proximal Ig segments (F_{PEVK} and F_{Ig}, which are equal since the two segments are arranged in series) as well as their lengths (ℓ_{PEVK} and $\ell_{\mathrm{Ig}}^{\mathrm{prox}}$).

We used the constrained optimization routine "fmincon" in MATLAB® to find values for p_{Ig}, p_{PEVK}, and K_0. Equations (4.1) and (4.2) yield forces and are used as constraints in the optimization problem. To relate these forces to the stresses reported by Leonard and Herzog [26], we assumed a myofibril diameter of 0.9 µm based on the estimation proposed by Linke et al. [51]. For sarcomere lengths between 2.4 µm and

3.6 µm, experimental data from passively stretched myofibrils were used to solve the following optimization problem:

$$\text{minimize} \qquad J_1\left(p_{\text{Ig}}^f, p_{\text{PEVK}}, K_0\right) = \left(F_{\text{exp}} - F_{\text{calc}}\right)^2 \tag{4.11a}$$

$$\text{subject to} \qquad \text{Eqs. (4.1), (4.2), and (4.3)} \tag{4.11b}$$

where p_{Ig}^f is the persistence length of one folded Ig domain, F_{exp} is the force obtained from experimental measurements [26], and F_{calc} is the calculated titin force. For sarcomere lengths greater than 3.6 µm, where proximal Ig domains begin to unfold, we used the parameters α_0, β_0, b_u, and b_f from Li et al. [54] in the unfolding rate equations (Eqs. (4.6), (4.7a), and (4.7b)); we then solved a similar optimization problem to calculate the persistence length of one unfolded Ig domain (p_{Ig}^u):

$$\text{minimize} \qquad J_2\left(p_{\text{Ig}}^u\right) = \left(F_{\text{exp}} - F_{\text{calc}}\right)^2 \tag{4.12a}$$

$$\text{subject to} \qquad \text{Eqs. (4.1), (4.2), and (4.4)} \tag{4.12b}$$

Upon solving Eqs. (4.11) and (4.12), we obtained a set of parameters for simulations of passive stretching. Finally, we obtained parameters for simulations of active stretching by solving the same optimization problems (i.e., Eqs. (4.11) and (4.12)), but where the myofibril force data from Leonard and Herzog [26] were increased by 25% [56].

4.3.4 Macro-scale constitutive model with force enhancement

We extended the muscle material model proposed by Blemker et al. [1]. In their model, muscle is considered to be a transverse isotropic, nearly incompressible, hyperelastic material. Therefore, the strain energy function is additively separated (uncoupled) into volumetric and isochoric behaviours. Our proposed modifications affect the model's behaviour in the fiber direction only. Specifically, to include force enhancement in the constitutive model, we replaced the normalized passive force–length (FL) curve in the muscle

model proposed by Blemker et al. [1] with a new normalized passive FL curve generated from a WLC model with actin–titin binding. This modification affected the calculation of the Cauchy stress in only the fiber direction. We also replaced the piecewise-quadratic function proposed by Blemker et al. to describe the active force–length relationship with an asymmetric Weibull distribution, as described below and shown in Supplementary Material (Sect. 4.7).

The isochoric strain energy function proposed by Blemker et al. [1] is defined as the sum of three terms: the along-fiber shear strain energy (W_1 with material parameter G_1), the cross-fiber shear strain energy (W_2 with material parameter G_2), and the fiber-stretch strain energy (W_3). These strain energies are functions of the invariants ($\bar{I}_1$, $\bar{I}_2$, $\bar{I}_4$, and $\bar{I}_5$) of the modified right Cauchy–Green tensor ($\bar{\mathbf{C}}$). The isochoric strain energy function (Ψ_{iso}) is defined as follows:

$$\Psi_{\text{iso}} = W_1\left(\bar{I}_4, \bar{I}_5\right) + W_2\left(\bar{I}_1, \bar{I}_4, \bar{I}_5\right) + W_3\left(\lambda(\bar{I}_4)\right) \tag{4.13a}$$

$$W_1\left(\bar{I}_4, \bar{I}_5\right) = G_1\left(\frac{\bar{I}_5}{\bar{I}_4^2} - 1\right) \tag{4.13b}$$

$$W_2\left(\bar{I}_1, \bar{I}_4, \bar{I}_5\right) = G_2\left[\cosh^{-1}\left(\frac{\bar{I}_1\bar{I}_4 - \bar{I}_5}{2\sqrt{\bar{I}_4}}\right)\right]^2 \tag{4.13c}$$

where $\lambda = \sqrt{\bar{I}_4}$ is the fiber stretch. The fictitious isochoric second Piola–Kirchhoff stress tensor ($\bar{\mathbf{S}}_{\text{iso}}$) and the fictitious material elasticity tensor ($\bar{\mathbb{C}}_{\text{iso}}$) can be computed as follows:

$$\bar{\mathbf{S}}_{\text{iso}} = \bar{\mathbf{S}}_1 + \bar{\mathbf{S}}_2 + \bar{\mathbf{S}}_3 \tag{4.14a}$$

$$\bar{\mathbf{S}}_3 = 2\frac{\partial W_3}{\partial \bar{I}_4}\mathbf{a}_0 \otimes \mathbf{a}_0 \tag{4.14b}$$

$$\bar{\mathbb{C}}_{\text{iso}} = \bar{\mathbb{C}}_1 + \bar{\mathbb{C}}_2 + \bar{\mathbb{C}}_3 \tag{4.14c}$$

$$\bar{\mathbb{C}}_3 = 2J^{-4/3}\frac{\partial \bar{\mathbf{S}}_3}{\partial \bar{\mathbf{C}}} \tag{4.14d}$$

where $\bar{\mathbf{S}}_k$ and $\bar{\mathbb{C}}_k$ are, respectively, the fictitious second Piola–Kirchhoff stress tensor

and the fictitious material elasticity tensor calculated with respect to each term of the isochoric strain energy function ($k \in \{1, 2, 3\}$); $\mathbf{a}_0$ is a unit vector in the direction of the fiber in the referential configuration; J is the volumetric ratio; and "$\otimes$" is the tensor product operator. Here, we have shown the calculation of only $\bar{\mathbf{S}}_3$ and $\bar{\mathbb{C}}_3$ because our modifications to the original formulation affect the stress in only the fiber direction; descriptions of $\bar{\mathbf{S}}_1$, $\bar{\mathbf{S}}_2$, $\bar{\mathbb{C}}_1$, and $\bar{\mathbb{C}}_2$ have been provided by Blemker et al. [1] and Holzapfel [45]. Upon calculating $\bar{\mathbf{S}}_{\text{iso}}$ and $\bar{\mathbb{C}}_{\text{iso}}$, the second Piola–Kirchhoff stress tensor and the material elasticity tensor can be calculated following the procedure described by Holzapfel [45].

Our muscle material model includes a variable α, $0 \leq \alpha \leq 1$, representing the muscle activation. Force enhancement has been observed during maximal as well as submaximal activation [57, 58]. To the best of our knowledge, no study has correlated force enhancement with levels of submaximal activation. In our model, we assume that the muscle's passive force (f_{passive}) is produced entirely by titin and varies linearly with activation [44]:

$$f_{\text{passive}} = (1 - \alpha)\, f_{\text{titin}}^{\alpha=0} + \alpha f_{\text{titin}}^{\alpha=1} \tag{4.15}$$

where $f_{\text{titin}}^{\alpha=0}$ and $f_{\text{titin}}^{\alpha=1}$ are the titin force that would be produced during a purely passive stretch and during a fully active stretch (while considering actin–titin binding), respectively.

Two additional parameters in our finite element simulations are the optimal fiber stretch (λ_{opt}) and the fiber stretch at the onset of activation (λ_{act}). Specifically, λ_{opt} is the ratio of the fiber length at which the maximum isometric stress (σ_{max}) occurs to the fiber length at the beginning of the simulation. The titin force that would be produced during a fully active stretch ($f_{\text{titin}}^{\alpha=1}$) is defined as a function of fiber stretch (λ). This force–stretch relationship ($f_{\text{titin}}^{\alpha=1}$) also depends on λ_{act} because the force produced by titin depends on its free length, which is determined by the sarcomere length at the onset of activation (ISL). Because sarcomeres are arranged in series to form a muscle fiber, we assumed that all sarcomeres have the same stretch, optimal stretch, and stretch

at the onset of activation; thus, these quantities are equal to the corresponding quantities defined for the whole fiber (i.e., λ, λ_{opt}, and λ_{act}, respectively). We used the optimal sarcomere length ($\ell_{\text{sar}}^{\text{opt}}$), λ_{opt}, and λ_{act} to compute the sarcomere length at the beginning of and during an active stretch (ISL and ℓ_{sar}, respectively), as well as their elongations relative to the optimal sarcomere length ($x_{\text{sar}}^{\text{initial}}$ and x_{sar}):

$$\text{ISL} = \ell_{\text{sar}}^{\text{opt}} \frac{\lambda_{\text{act}}}{\lambda_{\text{opt}}} \tag{4.16a}$$

$$x_{\text{sar}}^{\text{initial}} = \text{ISL} - \ell_{\text{sar}}^{\text{opt}} \tag{4.16b}$$

$$\ell_{\text{sar}} = \ell_{\text{sar}}^{\text{opt}} \frac{\lambda}{\lambda_{\text{opt}}} \tag{4.16c}$$

$$x_{\text{sar}} = \ell_{\text{sar}} - \ell_{\text{sar}}^{\text{opt}} \tag{4.16d}$$

These calculations were required because Eqs. (4.8), (4.9a), (4.9b), and (4.2) (which are used to compute the passive force) require lengths and elongations while finite element simulations report lengths in terms of stretch (λ).

We used a normalized fiber FL curve [59] comprised of passive and active components. Specifically, we normalized $f_{\text{titin}}^{\alpha=0}$ and $f_{\text{titin}}^{\alpha=1}$ by the maximum isometric force and added these terms to the normalized active fiber force (f_{active}):

$$f_{\text{total}}(\lambda, \alpha) = (1-\alpha) f_{\text{passive}}^{\alpha=0}(\lambda) + \alpha f_{\text{passive}}^{\alpha=1}(\lambda) + \alpha f_{\text{active}}(\lambda) \tag{4.17}$$

where f_{total} is the total fiber force (which is a function of fiber stretch λ and activation α), and $f_{\text{passive}}^{\alpha=0}$ and $f_{\text{passive}}^{\alpha=1}$ are the normalized versions of $f_{\text{titin}}^{\alpha=0}$ and $f_{\text{titin}}^{\alpha=1}$, respectively. The algorithms used to calculate $f_{\text{passive}}^{\alpha=0}$ and $f_{\text{passive}}^{\alpha=1}$ have been described in more detail in Supplementary Material (Sect. 4.8).

We used an asymmetric Weibull distribution to describe the normalized active FL relationship. We chose the Weibull curve, which has been used in other muscle material models [60], because it is C^1 continuous and it fits the total stress–stretch curve we

sought to approximate. The normalized active fiber force (f_{active}) is defined as follows:

$$f_{\text{active}} = \begin{cases} \frac{\lambda-\lambda_{\min}}{\lambda_{\text{opt}}-\lambda_{\min}} \exp\left[\frac{(2\lambda_{\min}-\lambda-\lambda_{\text{opt}})(\lambda-\lambda_{\text{opt}})}{2(\lambda_{\text{opt}}-\lambda_{\min})^2}\right], & \text{if } \lambda > \lambda_{\min} \\ 0, & \text{otherwise} \end{cases} \tag{4.18}$$

where $\lambda_{\min}$ is the minimum fiber stretch at which cross-bridges can form. With the assumption that the volume of the muscle remains constant, the Cauchy stress in the fibers (σ_{total}) can be written as follows:

$$\sigma_{\text{total}}(\lambda, \alpha) = \sigma_{\max} f_{\text{total}}(\lambda, \alpha) \frac{\lambda}{\lambda_{\text{opt}}} \tag{4.19}$$

The Cauchy stress in the fibers and the strain energy they produce (W_3) are related as follows:

$$\lambda \frac{\partial W_3}{\partial \lambda} = \sigma_{\text{total}}(\lambda, \alpha) \tag{4.20}$$

Equation (4.20) can be used to calculate the fictitious second Piola–Kirchhoff stress tensor ($\bar{\mathbf{S}}_3$, Eq. (4.14b)) using the chain rule.

To create a user-defined material in a finite element software (e.g., Ansys®, Abaqus®, or FEBio), one must implement a function to calculate the spatial elasticity tensor. To calculate the passive stress component of the spatial elasticity tensor, we first calculated the material elasticity tensor (Eq. (4.14d)); we then changed the reference frame to the spatial description. Equation (4.14d) is computed by calculating the first derivative of the WLC (Eq. (4.2)) with respect to the fiber stretch. This calculation is nontrivial because Eq. (4.2) is implicit and its derivative yields an ordinary differential equation that would need to be solved at each time step during a simulation. To avoid this computational burden, we calculated the spatial elasticity tensor using an approximation technique described previously [61]. (Alternatively, one may interpolate the passive stress–stretch curve such that the first derivative can be easily calculated.)

4.4 Numerical examples

In this section, we describe four tests to verify and validate our proposed model [62]. The first two tests investigated the numerical stability of our proposed model. In Test 1A, we used a cube model to show that the eigenvalues of the spatial elasticity tensor remain positive in a variety of active stretch conditions, providing evidence that the spatial elasticity tensor is positive definite and that our model is numerically stable. The spatial elasticity tensor can be represented by a symmetric 6×6 matrix using Voigt notation having six eigenvalues, each providing a measure of stiffness in the direction of the associated eigenvector [63]. Test 1B studied convergence using a 3D filament model in which each element had a slightly different optimal fiber length, which is a scenario that is prone to instability. In Test 2, we simulated our model with a 3D muscle geometry that has been presented in the literature [64]. Finally, in Test 3, we used this 3D muscle geometry to validate the lengths of the fibers during an isometric contraction on the descending limb of the active FL curve.

4.4.1 Parameters and curves used in numerical examples

The optimization problems defined in Sect. 4.3.3 were solved to find the parameters used in the WLC equations. In our model, the passive force is produced only by titin; the parameters are provided in Table 4.1. The active FL curve is described phenomenologically using Eq. (4.18), with the parameters $\lambda_{\text{opt}} = 1.0$ and $\lambda_{\text{min}} = 0.51$. We compared the total stress produced by our model (Eq. (4.19)) with the mean stress measured from actively stretched myofibrils [26]. As shown in Fig. 4.3, the stress predicted by our model was generally within 1 standard deviation of the experimental data.

Table 4.1: Parameters used in the worm-like chain (WLC) equations for titin during passive and active stretches. The values in the first four rows were computed; the remaining values were obtained from the literature as indicated. To compute the parameters for active stretching, the sarcomere was assumed to be at its optimal length at the onset of activation; the initial sarcomere length (ISL) is shown in parentheses. The parameter values in column 1 were obtained by solving the optimization problems stated in Eqs. (4.11) and (4.12). The parameters p_{Ig}^{f} and p_{Ig}^{u} have the same values regardless of activation. The parameters p_{PEVK} and K_0 during active stretching were obtained by solving the optimization problem stated in Eqs. (4.11) and (4.12) with an assumed 25% increase in the experimentally measured force (column 2) and using Eq. (4.9) (column 3).

Parameter	Value		
	Passive Stretch	Active Stretch (ISL: 2.4 µm)	Active Stretch (ISL: 3.4 µm)
p_{Ig}^{f}, persistence length of one folded proximal Ig domain (µm)	1.759×10^{-4}	1.759×10^{-4}	1.759×10^{-4}
p_{Ig}^{u}, persistence length of one unfolded proximal Ig domain (µm)	1.26×10^{-5}	1.26×10^{-5}	1.26×10^{-5}
p_{PEVK}, persistence length of PEVK segment (µm)	5.23×10^{-5}	5.75×10^{-5}	5.43×10^{-5}
K_0, stretch modulus of PEVK segment (nN)	0.090	0.170	0.120
L_{Ig}, contour length of one Ig domain (nm) [55]	4.5	4.5	4.5
N^{prox}, number of proximal Ig domains [55]	50	50	50
L_{PEVKr}, contour length of one PEVK residue (nm) [55]	0.36	0.36	0.36
N^{PEVKr}, number of PEVK residues [55]	800	800	800
α_0, unfolding rate of proximal Ig domain at zero force (s^{-1}) [54]	3.3×10^{-3}	3.3×10^{-3}	3.3×10^{-3}
β_0, folding rate of proximal Ig domain at zero force (s^{-1}) [54]	3.3×10^{-1}	3.3×10^{-1}	3.3×10^{-1}
b_u, activation barrier width for an unfolding event to occur (nm) [54]	0.25	0.25	0.25
b_f, activation barrier width for a folding event to occur (nm) [54]	2.2	2.2	2.2

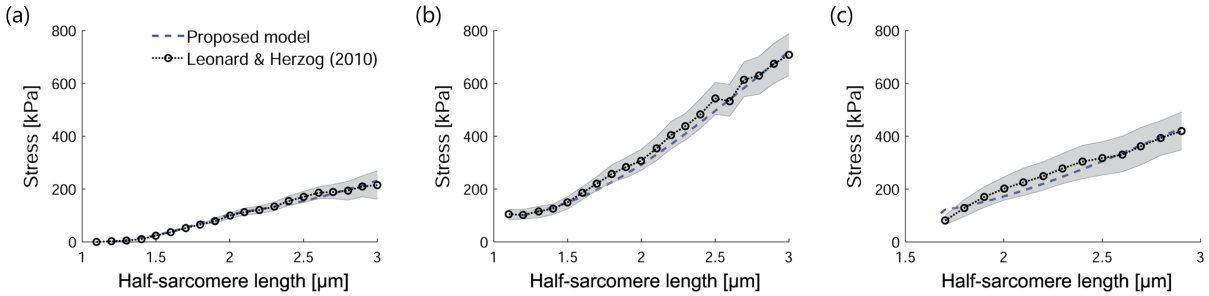

Figure 4.3: Stress vs. half-sarcomere length calculated by the proposed worm-like chain (WLC) model with experimental data from stretching myofibrils under three conditions: (a) passive stretch, (b) active stretch from the optimal sarcomere length (2.4 µm), and (c) active stretch from a sarcomere length on the descending limb of the active FL curve (3.4 µm). The experimental data (circles) shown in panels (a) and (b) were used to identify the parameters of the WLC model through optimization (Eqs. (4.11) and (4.12)); the data shown in panel (c) were used to validate our method of adjusting PEVK parameters based on the length of the PEVK segment at the onset of activation (Eq. (4.9)). The simulations (dashed lines) were generated using the parameters provided in Table 4.1. Circles: experimental data digitized from Leonard and Herzog [26]; shaded regions: ±1 standard deviation.

After finding the parameters for the WLC model, the passive stress–stretch curve can be determined when the sarcomere length at the onset of activation is known. The relationship between normalized total stress and normalized fiber length is shown in Fig. 4.4 when maximal activation was applied at five normalized fiber lengths between 1.0 and 1.4. Regardless of the fiber length at the onset of activation, further elongation produced stresses that were substantially greater than the stress predicted for isometric contraction. Because we consider calcium-ion binding to occur instantaneously, the stiffness of the PEVK segment increases instantaneously and the stress–length curves shown in Fig. 4.4 are discontinuous at the fiber length at which activation was applied. At a given fiber length, the stress was greater if the activation began at a shorter fiber length.

4.4.2 Numerical stability (Tests 1A and 1B)

In Test 1A, we modelled a unit cube comprising a single element and subjected it to a uniaxial displacement. Boundary conditions were defined such that strictly axial stretches were applied at one end (which we refer to as the "displaced" end) with the lateral sur-

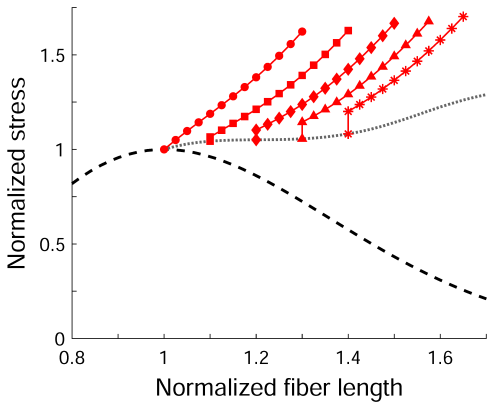

Figure 4.4: Normalized stress vs. normalized fiber length predicted by our simulations of active stretching. Dashed line: normalized active stress ignoring force enhancement; dotted line: normalized total stress ignoring force enhancement; symbols: normalized total stress considering force enhancement when activation was applied at normalized fiber lengths of 1.0 (circles), 1.1 (squares), 1.2 (diamonds), 1.3 (triangles), and 1.4 (stars).

faces traction free; the opposite end was fixed in all directions. The fiber direction was parallel to the applied stretch. The simulation was defined with one step, with a (pseudotime) duration of 1 s to ramp up the boundary conditions. We used parameter values $G_1 = G_2 = 50$ kPa in the strain energy functions W_1 and W_2, $\sigma_{\max} = 100$ kPa, $\lambda_{\text{opt}} = 1.0$, $\alpha = 1.0$, and bulk modulus $K = 5 \times 10^4$ kPa.

We calculated the eigenvalues of the spatial elasticity tensor while a variety of stretches were applied to the cube. Specifically, we analyzed the eigenvalues associated with the eigenvectors along the diagonal of the cube ($\mathbf{q}^{(1)}$), in the fiber direction ($\mathbf{q}^{(2)}$), and in the plane perpendicular to the fibers ($\mathbf{q}^{(3)}$). (The remaining eigenvectors are associated with shear modes and were immaterial for this analysis of a uniaxial stretch.) We compared the behaviour of three models: the model of Blemker et al. [1] in its original form, a model using the active FL curve from Blemker et al. with force enhancement added, and our proposed model.

In Test 1B, we modelled a 3D filament of length 0.01 m and square cross-section of height 0.001 m, meshed with 20 hexahedral elements (Fig. 4.5). We assumed that the elements had non-uniform optimal fiber lengths. Specifically, we randomly selected the parameter λ_{opt} associated with each element in the range $[0.99, 1.01]$. Thus, the fiber

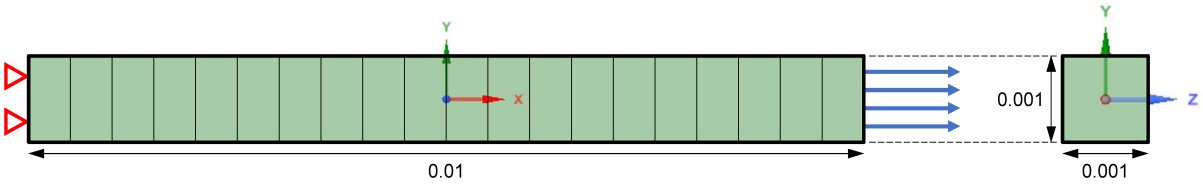

Figure 4.5: Geometry of 3D filament model used in the stability analysis. The geometry was meshed into 20 hexahedral elements arranged in series, each with dimensions $0.001 \times 0.001 \times 0.0005$ m. The red triangles indicate the fixed end, where the displacement along the x-axis was constrained; the blue arrows on the opposite end indicate the applied displacement. The fibers were directed along the x-axis.

length associated with each element at the beginning of the simulation was either on the ascending limb, at the peak, or on the descending limb of the active FL curve. The other parameter values were defined as above. A displacement of 0.003 m (i.e., 30% of the undeformed length, or a stretch of 1.3) was applied to the displaced end. We compared the behaviour of the model of Blemker et al. [1] to that of our proposed model.

4.4.3 Benchmark using 3D muscle geometry (Test 2)

In Test 2, we simulated the 3D muscle geometry shown in Fig. 4.6 under active stretching with submaximal activation. The geometry was created by revolving a 2D curve around the longitudinal axis. The radius along the length of the muscle ($r(x)$) was proposed by Böl and Reese [64]:

$$r(x) = r_2 \exp\left[-\frac{x^2}{h^2} \ln\left(\frac{r_2}{r_1}\right)\right], \quad x \in [-h, h] \tag{4.21}$$

where $r_1 = 0.009$ m is the minimum radius (at $x = \pm h$), $r_2 = 0.0175$ m is the maximum radius (at $x = 0$), and $h = 0.050$ m is the half-length of the muscle. The fibers were directed along the longitudinal axis.

A convergence analysis was performed to determine the appropriate refinement of the mesh. We applied fixed-displacement boundary conditions in all directions to the face at $x = -h$ and a stretch of 20% of the total muscle length (i.e., a stretch of $0.4h$) to the opposite face (the "displaced" end), with the muscle fully activated. A single point

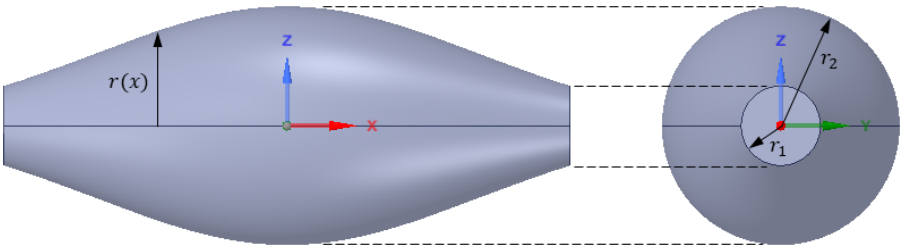

Figure 4.6: 3D muscle geometry used to demonstrate force enhancement. The fibers are directed along the x-axis. The radius is defined by a function proposed by Böl and Reese [64], with minimum radius r_1 occurring at $x = \pm h$ and maximum radius r_2 occurring at $x = 0$.

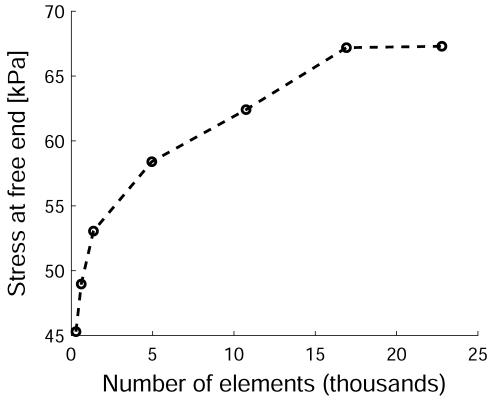

Figure 4.7: Convergence analysis for mesh refinement of the 3D muscle geometry. Longitudinal stress converged with 16,915 elements.

P at the displaced end with coordinates $(0.05, 0.0, 0.009)$ m was selected to measure the longitudinal stress during the convergence analysis. The average size of the elements in the mesh was reduced by 0.002 m until a difference of less than 1% in the longitudinal stress was observed at point P when compared to the stress observed in the previous, coarser mesh. The final, converged mesh had 16,915 hexahedral elements with an average side length of 0.0012 m (Fig. 4.7).

We used the 3D muscle geometry to simulate active stretching with five fiber lengths at the onset of activation. The simulation was divided into three steps, each with a (pseudotime) duration of 1 s: passive stretching, activation, and active stretching. During passive stretching, the muscle was stretched to a predetermined length with activation set to zero; the muscle was then activated with the length held constant. Finally, the muscle

was stretched while activated until a final length of 0.14 m was reached (i.e., in total, a 40% stretch was applied). In the passive-stretching step, the muscle was lengthened by $\Delta\ell$, where $\Delta\ell \in \{0, 0.01, 0.02, 0.03, 0.04\}$ m. In the active-stretching step, the muscle was lengthened by $0.04\text{ m} - \Delta\ell$. Thus, the fiber stretches at the onset of activation (λ_{act}) were, respectively, 1.0 (fully active stretch), 1.1, 1.2, 1.3, and 1.4 (fully passive stretch). These five simulations were performed at two activation levels: $\alpha = 0.2$ and $\alpha = 0.85$.

4.4.4 Fiber length distribution in an isometric contraction (Test 3)

In Test 3, we used the 3D muscle geometry shown in Fig. 4.6 to test the robustness of our model to non-uniform fiber lengths. The muscle was first passively stretched by 10%, then activated while being held at a constant length. This simulation was performed twice: once using our proposed model and once using the model of Blemker et al. [1]. The simulation results were validated through comparison to experimental measurements from a rat tibialis anterior muscle [46]. Specifically, we compared the distribution of fiber lengths before and after activation.

To perform this comparison, we determined the parameters for our simulations as follows. First, the mean sarcomere length after passive stretching and after applying activation must be 2.52 µm ±5% and 2.68 µm ±5%, respectively, as reported by Moo et al. [46]. Second, the force produced at the displaced end of our model (F_{disp}) must be equal to the experimentally measured force of 1.5 N ±5% [46]. We estimated F_{disp} as follows:

$$F_{\text{disp}} = \sigma_{\text{disp}} A_{\text{disp}} \frac{A_{\text{mid}}}{\text{PCSA}_{\text{rat}}} \tag{4.22}$$

where σ_{disp} and A_{disp} are the stress and area at the displaced end of our model, A_{mid} is the mid-belly area of the 3D muscle geometry (i.e., the cross-sectional area at $x = 0$), and $\text{PCSA}_{\text{rat}} = 0.38\text{ cm}^2$ is the physiological cross-sectional area of a typical rat tibialis anterior muscle [65]. Our final model parameters were $\lambda_{\text{opt}} = 1.1$, $\sigma_{\text{max}} = 40$ kPa, and $\alpha = 0.12$.

4.5 Results and discussion

In Test 1A, we subjected a unit cube to a uniaxial displacement, where the fiber direction was parallel to the applied stretch. Negative eigenvalues were observed in the spatial elasticity tensor during an active stretch simulation when using the model of Blemker et al. [1] (Fig. 4.8(a)). Specifically, the eigenvalue associated with the fiber direction was negative over a large range of fiber stretch ($1.147 \leq \lambda \leq 1.574$). This range of λ lies within the descending limb region of the active FL curve; the negative eigenvalues are caused by the negative stiffness in this region. Negative eigenvalues in the spatial elasticity tensor indicate that the strain energy function is negative [16] and the material model is unstable [66]. The eigenvalues corresponding to a model using the active FL curve from Blemker et al. with force enhancement added are shown in Fig. 4.8(b). A region with negative eigenvalues persisted, albeit over a narrower range of fiber stretch ($1.4 \leq \lambda \leq 1.412$). Negative eigenvalues were still observed in this model because the active FL curve in the original model was not compatible with the passive FL curves produced from the WLC model (when using the parameters in Table 4.1). Moreover, the active FL curve proposed by Blemker et al. [1] is only C^0 continuous at a fiber stretch of 1.4, which coincides with the discontinuity observed in Fig. 4.8(b). Using our proposed model, which uses the normalized active force–stretch curve defined in Eq. (4.18) with force enhancement, all the eigenvalues were positive and varied smoothly over the entire range of fiber stretch (Fig. 4.8(c)). Because the model of Blemker et al. with force enhancement added still resulted in negative eigenvalues in the spatial elasticity tensor, we did not consider this variant of the model further.

In Test 1B, we simulated a 3D filament model where the fibers within each element had the same optimal length but had slightly different lengths at the beginning of the simulation (and, therefore, different λ_{opt}). Because active force is a function of normalized fiber length, the elements did not generate equal forces at the beginning of the simulation: the system was not in static equilibrium. When using the model of Blemker

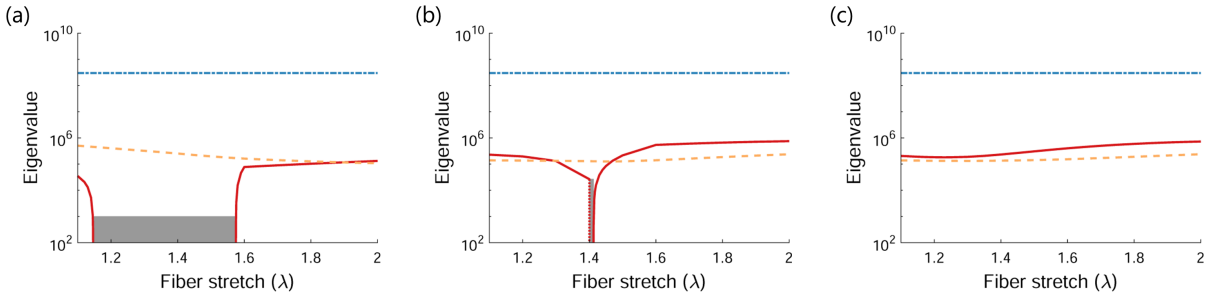

Figure 4.8: Eigenvalues of the spatial elasticity tensor during an active stretch of a unit cube using three muscle material models: (a) the model of Blemker et al. [1], (b) the model of Blemker et al. augmented with our model of force enhancement, and (c) our proposed model. One eigenvalue became negative in models (a) and (b) (grey rectangles); in model (c), all eigenvalues remained positive and varied smoothly as the fiber was actively stretched. Blue dash-dotted, red solid, and yellow dashed lines correspond, respectively, to the eigenvalues associated with the eigenvectors directed along the main diagonal, in the fiber direction, and in the plane perpendicular to the fiber direction. Vertical axis is a logarithmic scale.

et al. [1], the simulation did not converge, likely due to the negative stiffness observed in Fig. 4.8(a). Using our proposed model, the simulation converged and all fibers stretched by approximately the same amount. As shown in Fig. 4.9, non-uniform fiber stretches persisted for the first 0.07 s of the simulation; thereafter, the fiber stretch was approximately equal in all elements. Our proposed model demonstrated numerical stability and robustness to perturbations in the fiber lengths, and the simulation converged even when operating on the descending limb of the active FL curve.

In Test 2, we tested our proposed model with the 3D muscle geometry shown in Fig. 4.6. We applied a passive stretch of between 0% and 40% of the undeformed length, then activated the muscle, and finally applied an active stretch such that the final length of the muscle was 140% of its undeformed length (Fig. 4.10(a)). The stress at the displaced end of the muscle is shown in Fig. 4.10(b) for activation $\alpha = 0.2$ and in Fig. 4.10(c) for $\alpha = 0.85$. The line $\lambda = 1.4$ indicates a strictly passive stretch of 40% followed by an isometric contraction. When $\alpha = 0.2$, the stress measured at the displaced end of the muscle ranged from 170.01 kPa to 205.28 kPa depending on the fiber length at the onset of activation. When $\alpha = 0.85$, the stress measured at the displaced end of

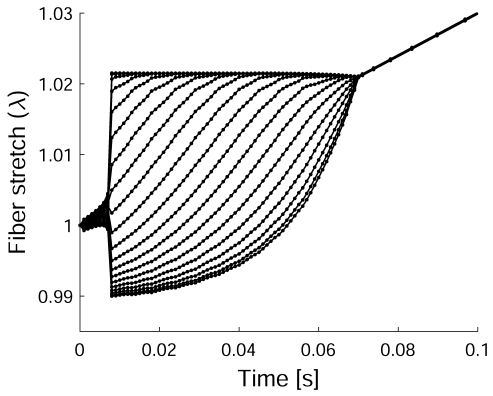

Figure 4.9: Fiber stretch in each element of a 3D filament model during an active stretch using our proposed model, where the fibers were initially of unequal normalized lengths. A displacement of 30% of the undeformed length was applied to the displaced end of the filament over 1 s. The first 0.1 s of this simulation is shown; all fibers stretched linearly with time over the remaining 0.9 s. The fiber stretch in all elements converged after approximately 0.07 s.

the muscle ranged from 399.19 kPa to 525.83 kPa. The stress following a purely active stretch was 20.7% greater than that following a purely passive stretch when $\alpha = 0.2$, and was 31.7% greater when $\alpha = 0.85$.

In Test 3, we validated the lengths of the fibers during an isometric contraction on the descending limb of the active FL curve, again using the 3D muscle geometry shown in Fig. 4.6. The muscle was passively stretched by 10%, then activated while being held at a constant length. As shown in Table 4.2, our model and the model of Blemker et al. [1] predicted similar mean sarcomere lengths, and the results of both models compared favourably to the measurements of mean sarcomere lengths in a rat tibialis anterior muscle [46]. However, the two models predicted dramatically different values for the length of the longest sarcomere. Following activation, the length of the longest sarcomere increased by 53.6% when using the model of Blemker et al. and increased by only 4.6% when using our proposed model. In this respect, our proposed model compared more favourably to the experimental measurements, where no statistically significant difference was observed between the longest sarcomere length before and after activation [46]. A possible explanation for the discrepancy between the two models is the negative slope

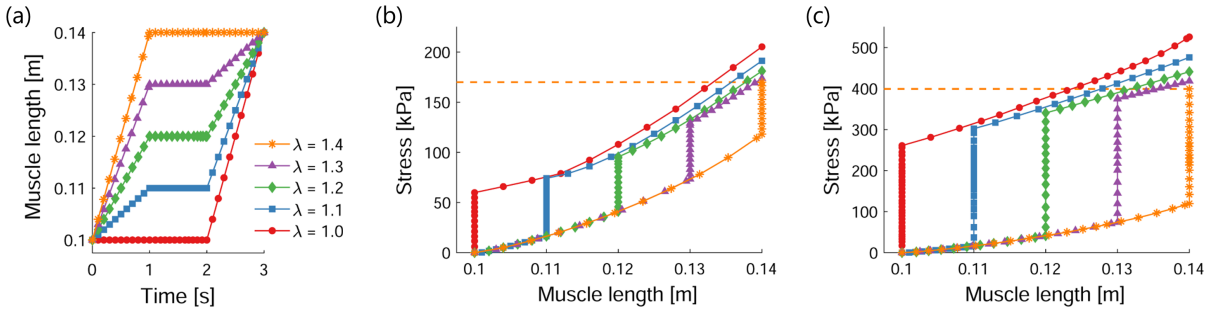

Figure 4.10: Three-step simulation of active stretching using the 3D muscle geometry with our proposed model. The muscle was passively stretched between 0% and 40% of the undeformed length, activated while held at a constant length, and then actively stretched. Activation was ramped up between pseudotimes of 1 s and 2 s, and remained constant thereafter. (a) Muscle length over time for each test scenario; and stress at the displaced end of the muscle with activation (b) $\alpha = 0.2$ and (c) $\alpha = 0.85$. Dashed horizontal lines in panels (b) and (c): isometric stress following a purely passive stretch.

of the total FL curve in the model of Blemker et al.: in this region, a given force can be generated by a muscle fiber at up to three distinct lengths (see dotted horizontal line in Fig. 4.1). In contrast, our proposed model has a strictly increasing stress–stretch curve due to force enhancement and, consequently, computes exactly one fiber length corresponding to a given force, which results in a narrower distribution of sarcomere lengths in the muscle. The coefficient of variation of sarcomere length was greater in the passive state than in the active state for both models, but greater in the active state than in the passive state in the experiments of Moo et al. This discrepancy may be due to the uncertainty in the material properties used in our simulations, the use of a 3D muscle geometry with a uniform and uniaxial fiber direction, and the fact that all fibers in the model were of the same length before the passive stretch. However, the shortest and longest sarcomere lengths, the range of lengths, and the coefficient of variation of sarcomere length were all closer to the experimental values when force enhancement was considered.

Table 4.2: Validation of sarcomere lengths during simulation of 3D muscle geometry. The muscle was passively stretched by 10% of its initial length ("Passive") and then activated while held at a constant length ("Active"). Following isometric contraction, our proposed model with force enhancement predicted a range of sarcomere lengths that compared favourably with the experiments of Moo et al. [46] (estimated marginal mean ± standard error).

	Moo et al. (2017)		Proposed model		Blemker et al. (2005)	
	Passive	Active	Passive	Active	Passive	Active
Mean sarcomere length (µm)	2.68 ± 0.03	2.52 ± 0.06	2.64	2.41	2.64	2.44
Length of shortest sarcomere (µm)	2.34 ± 0.05	2.13 ± 0.04	2.40	2.03	2.40	1.76
Length of longest sarcomere (µm)	2.97 ± 0.03	2.92 ± 0.06	2.84	2.97	2.91	4.47
Range of sarcomere lengths (µm)	0.67 ± 0.03	0.81 ± 0.03	0.44	0.94	0.51	2.71
Coefficient of variation of sarcomere length	4.1%	5.4%	10.4%	2.4%	29.7%	3.4%

4.6 Conclusions

This work presents our implementation and testing of a constitutive muscle material model that exhibits force enhancement. Our material model was developed based on the model proposed by Blemker et al. [1], which is available in the open-source finite element software FEBio. In our proposed model, the passive FL curve was created by considering both actin–titin binding and the increase in stiffness caused by the binding of calcium ions to the PEVK segment of the titin [38]; the active FL curve was created using a Weibull curve. The parameters for the passive and active FL curves were obtained by fitting them to data from stretch experiments on rabbit psoas myofibrils [26]. The numerical stability of our material model was verified by demonstrating that the eigenvalues of the spatial elasticity tensor remained positive during an active stretch and by simulating the active stretch of a muscle filament with unequal fiber lengths. The simulation converged when our proposed model was used, but failed to converge when using the model of Blemker et al. [1], which has a negative stiffness region in the total FL curve. We applied our proposed material model to a 3D muscle geometry to perform further benchmarking. In simulations of passive stretching followed by active stretching

with a fixed total amount of stretch, we observed greater force enhancement when the activation was applied at a shorter fiber length. Simulating the same muscle geometry during an isometric contraction but with non-uniform fiber lengths, we observed sarcomere lengths that were in better agreement with experimental measurements than with the model of Blemker et al. [1], which does not consider force enhancement.

The are four noteworthy limitations of our proposed material model. First, the parameters that were used to construct the active and passive FL curves were obtained using a single set of data from experiments on rabbit psoas muscles; different sets of parameters may be required to accurately predict the behaviour of different muscles. Second, only one experimental curve of stress vs. half-sarcomere length was used to validate the proposed change in PEVK parameters in scenarios where the sarcomere length at the onset of activation was on the descending limb of the active FL curve during active stretching. Thus, there is uncertainty in how to determine the passive FL curve as a function of the sarcomere length at the onset of activation. Third, the micromechanical behaviour of the fibers was directly applied at the macroscopic level, a simplification that is adopted by many muscle models. In the proposed model, the energy terms W_1 and W_2 include contributions at the macroscopic level from the extracellular matrix (ECM, the medium within which the fibers reside) as well as the passive titin response. Because the parameters W_1 and W_2 are calculated with a phenomenological approach, it is not possible to isolate the contribution of the ECM within W_1 and W_2. In future work, we will account for the mechanical properties of the ECM and muscle fibers as two separate media, and perform an analytical homogenization [67, 68] to derive the macro-scale model from the micro-scale model, ensuring that the overall stiffness remains positive in the fiber direction to preserve numerical stability. A fourth limitation is that the model was not tested for force depression, where the isometric force generated by a muscle following active shortening is less than that observed if active shortening had not occurred beforehand [69]. The WLC model is capable of simulating force depression [38]; however, implementing and testing this situation in our model remains future work.

Our proposed muscle material model is one of the few models available that demonstrates force enhancement. In our model, force enhancement occurs because of two mechanisms occurring within each sarcomere: actin–titin binding, which reduces the free length of the titin, and the binding of calcium ions to the PEVK segment. Our model also uses parameters in the WLC model that are derived from experiments on myofibrils reported in the literature. To this end, two experimental curves of stress vs. half-sarcomere length (Figs. 4.3(a) and 4.3(b)) were used for identifying the model parameters and a third curve (Fig. 4.3(c)) was used for validation. The eigenvalues of the spatial elasticity tensor remained positive when using our model to simulate uniaxial stretches. Numerical stability was also demonstrated when muscle fibers were of non-uniform length. Our proposed muscle material model can be used to study the interaction between muscles and the nearby bones and soft tissues during eccentric contraction. The model may also be useful for studying the interaction between the body and wearable devices such as orthoses and prostheses in situations where muscles are not acting purely isometrically. We have provided our FEBio implementation and our three-step simulation with the 3D muscle geometry at https://simtk.org/projects/muscle_fe_titin so that others can reproduce and extend our results.

4.7 Supplementary Material: Normalized active force–length curve

The normalized active force–length curves discussed in Subsect. 4.3.4 are illustrated in Fig. 4.11.

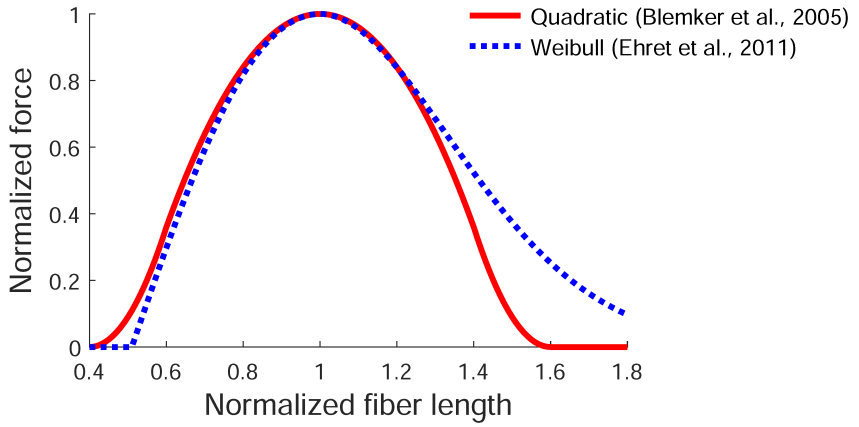

Figure 4.11: To model the normalized active force–length relationship, we replaced the piecewise-quadratic function proposed by Blemker et al. [1] (solid red line) with a Weibull distribution [60] (dotted blue line). The piecewise-quadratic function is symmetric about the optimal fiber length (i.e., normalized fiber length of 1) and nonzero from 0.4 to 1.6 normalized fiber lengths. The Weibull distribution is asymmetric: the ascending limb begins at a normalized fiber length of $\lambda_{\text{min}} = 0.51$ and the descending limb approaches zero asymptotically.

4.8 Supplementary Material: Calculation of normalized passive force–length curve

This section summarizes the process used to calculate the normalized passive force–length (FL) curve during passive and active stretching. During *passive* stretching (i.e., $a = 0$), the process shown in Fig. 4.12 is used to calculate the normalized passive FL curve ($f_{\text{passive}}^{a=0}$). The passive FL curve is nonzero only for sarcomere lengths greater than the optimal sarcomere length ($\ell_{\text{sar}}^{\text{opt}} = 2.4$ µm); thus, the half-sarcomere length (ℓ_{HS}) is initialized to 1.2 µm. If ℓ_{HS} is less than 1.8 µm, the threshold for the unfolding of proximal Ig domains, then Eqs. (4.1), (4.2), and (4.3) are solved; otherwise, Eqs. (4.1), (4.2), and (4.4) are solved, using Subprocess "A" (Fig. 4.13) to compute the number of folded and unfolded proximal Ig domains (N_u^{prox} and N_f^{prox}, respectively). The normalization factor c is calculated as follows:

$$\ell_{\text{sar}} = 2\ell_{\text{HS}} \tag{4.23}$$

$$A_{\text{sar}} = \frac{\pi\, d_{\text{sar}}^2}{4} \tag{4.24}$$

$$c = \frac{\ell_{\text{sar}}^{\text{opt}}}{A_{\text{sar}}\, \sigma_{\max}\, \ell_{\text{sar}}} \tag{4.25}$$

where diameter $d_{\text{sar}} = 0.36$ µm [51] and $\sigma_{\max} = 100$ kPa. The parameter sets that appear in Figs. 4.12 and 4.13 are listed in Table 4.3.

During *active* stretching at maximal activation (i.e., $a = 1$), the processes shown in Figs. 4.14 and 4.15 are used to calculate the normalized passive FL curve ($f_{\text{passive}}^{a=1}$). The elongation of one proximal Ig domain at the onset of activation ($x_{\text{Ig}}^{\text{ISL}}$) is calculated using the process shown in Fig. 4.14. Note that 3.6 µm is the sarcomere length beyond which proximal Ig domains unfold. The process shown in Fig. 4.15 is then used to calculate $f_{\text{passive}}^{a=1}$. The parameter sets that appear in Figs. 4.14 and 4.15 are listed in Table 4.3.

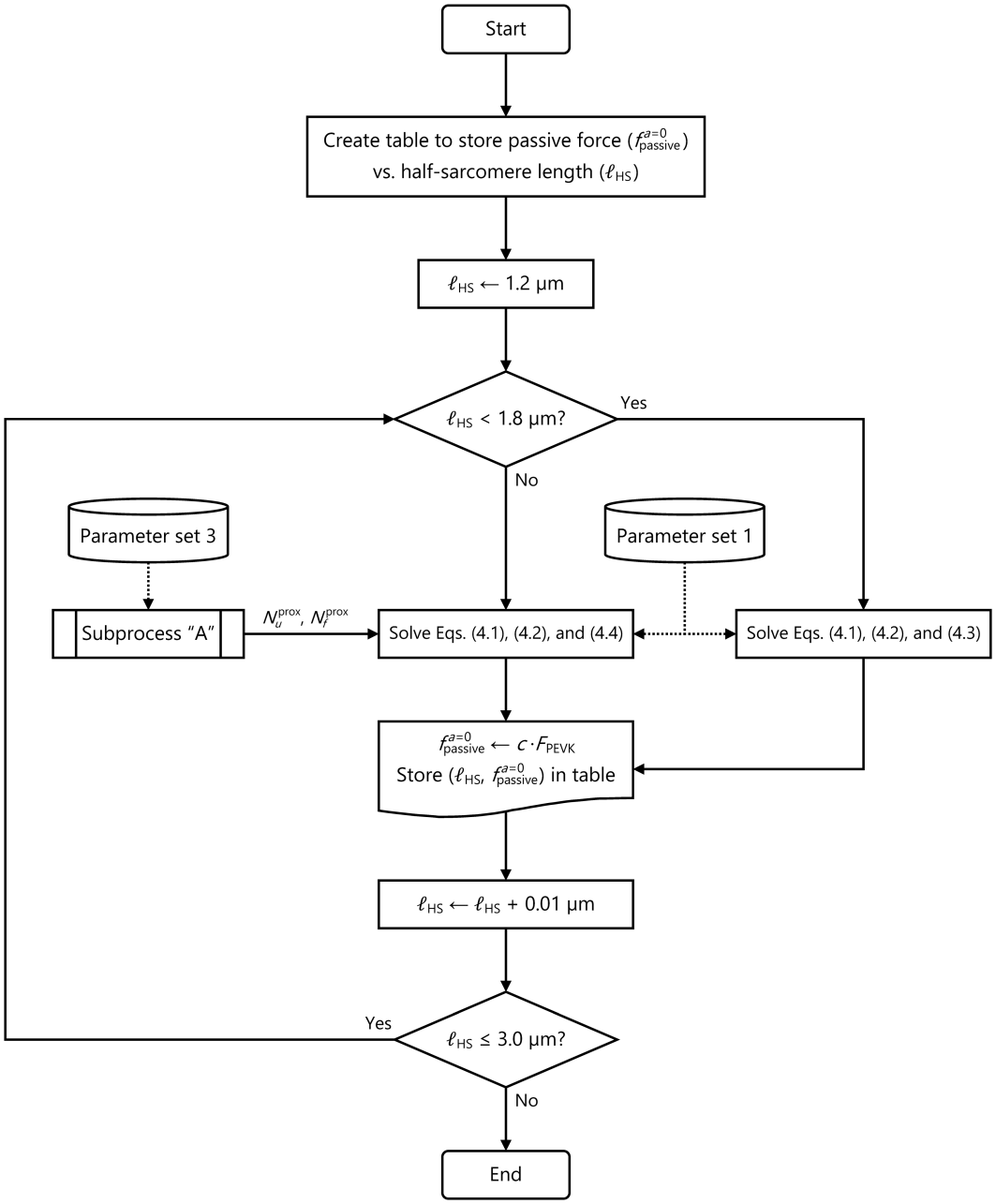

Figure 4.12: Process to calculate the normalized passive FL curve during passive stretching.

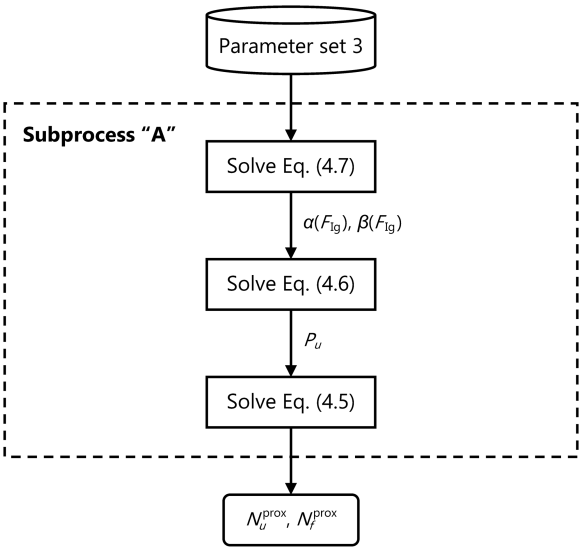

Figure 4.13: Subprocess to calculate the number of folded and unfolded proximal Ig domains.

Table 4.3: Sets of parameters used to calculate the normalized passive force–length curve.

Set	Parameter	Description
1	$K_{0,\text{active}}$, $K_{0,\text{passive}}$	Stretch modulus of PEVK segment in active and passive states
	k_B	Boltzmann constant
	L_{Ig}	Contour length of one Ig domain
	L_{PEVK}	Contour length of PEVK segment
	N^{prox}	Number of proximal Ig domains
	p^f_{Ig}, p^u_{Ig}	Persistence length of one Ig domain when folded and unfolded
	$p_{\text{PEVK,active}}$, $p_{\text{PEVK,passive}}$	Persistence length of PEVK segment in active and passive states
	T	Absolute temperature
2	ISL	Length of sarcomere at the onset of activation ("initial sarcomere length")
	$K_{0,\text{active}}$, $K_{0,\text{passive}}$	Stretch modulus of PEVK segment in active and passive states
	$\ell_{\text{sar}}^{\text{max-act}}$	Sarcomere length at which the descending limb of the active force–length curve reaches zero force
	$\ell_{\text{sar}}^{\text{opt}}$	Optimal sarcomere length
	$p_{\text{PEVK,active}}$, $p_{\text{PEVK,passive}}$	Persistence length of PEVK segment in active and passive states
3	α_0	Unfolding rate of proximal Ig domain at zero force
	β_0	Folding rate of proximal Ig domain at zero force
	b_f	Width of the activation barrier for a folding event to occur
	b_u	Width of the activation barrier for an unfolding event to occur
	k_B	Boltzmann constant
	T	Absolute temperature
4	L_{PEVKr}	Contour length of one PEVK residue
	N^{PEVKr}	Number of PEVK residues

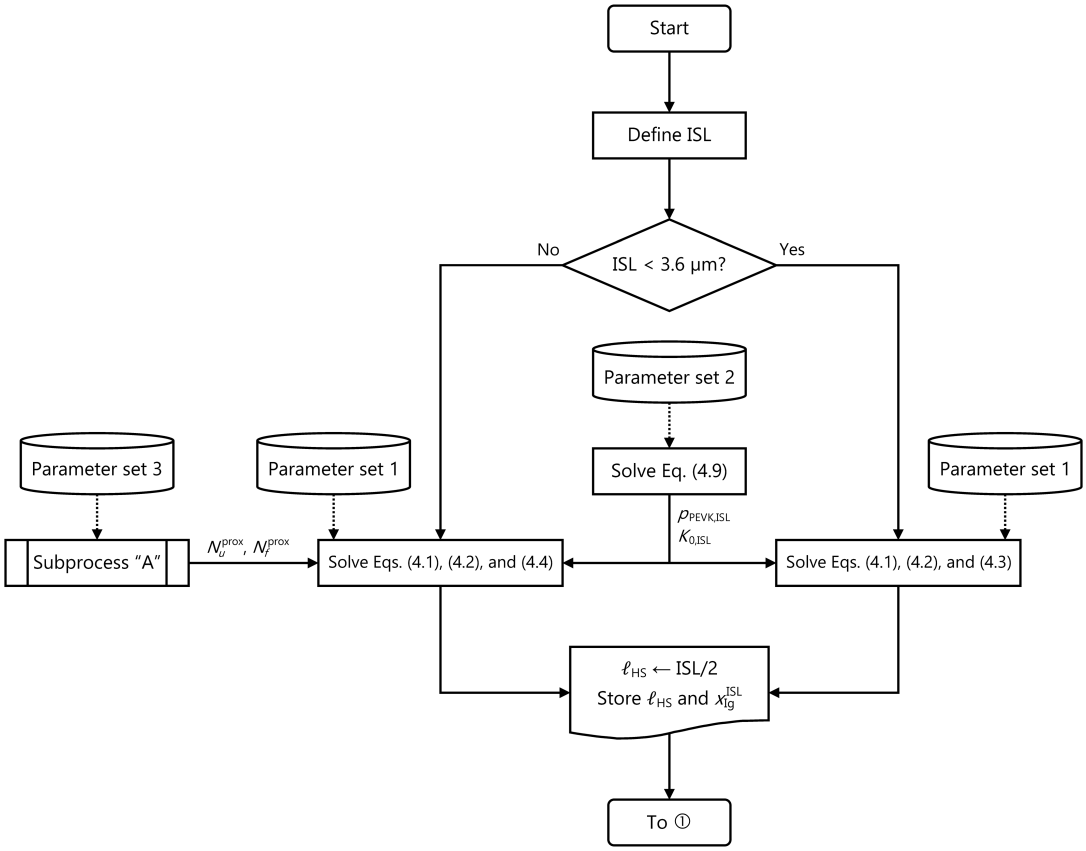

Figure 4.14: Process to calculate the elongation of one proximal Ig domain at the onset of activation.

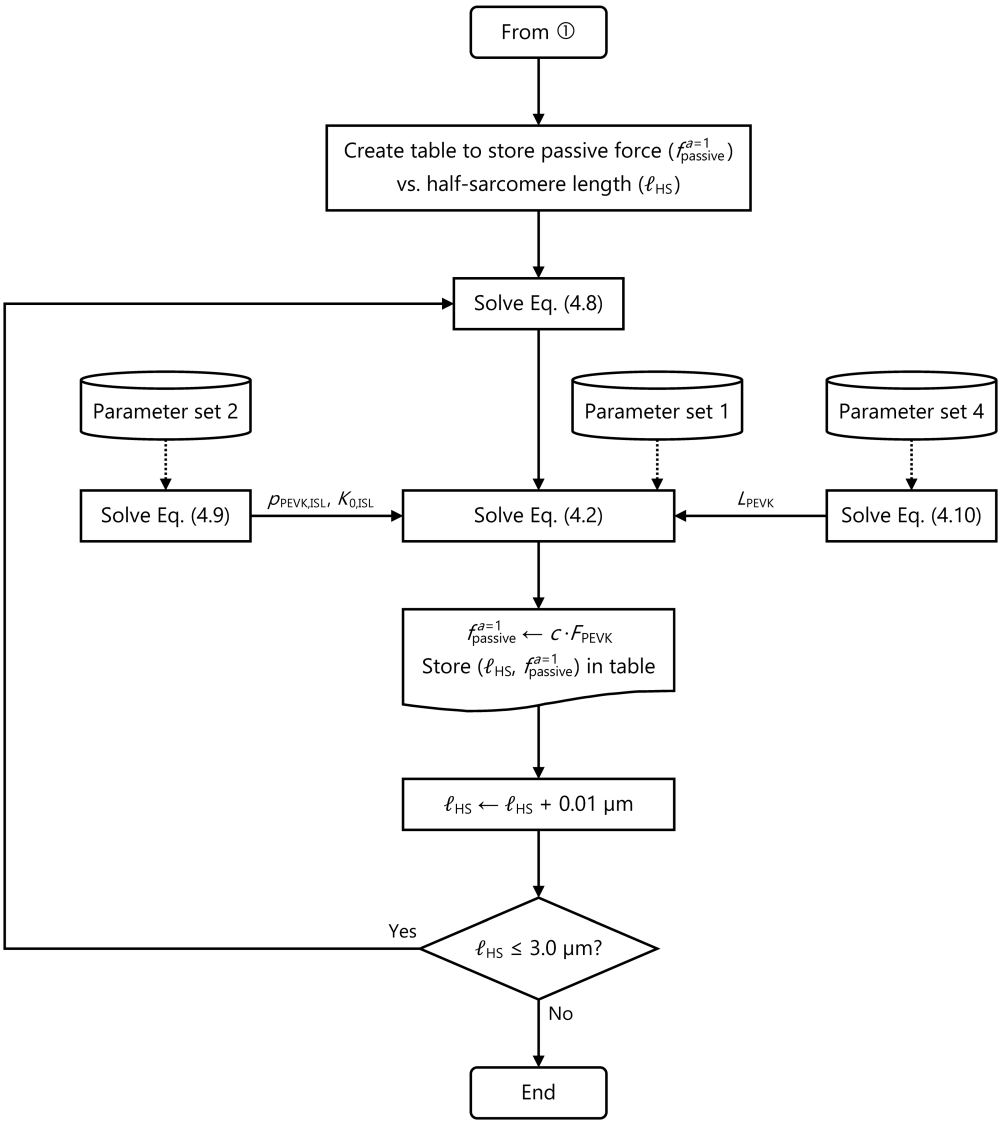

Figure 4.15: Process to calculate the normalized passive FL curve during active stretching at maximal activation.

Bibliography for Chapter 4

[1] S.S. Blemker, P.M. Pinsky, and S.L. Delp. A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38(4), 657–665, 2005.

[2] T.T. Dao and M.-C.H.B. Tho. A systematic review of continuum modeling of skeletal muscles: current trends, limitations, and recommendations. *Applied Bionics and Biomechanics*, 2018, 7631818, 2018.

[3] M. Lilja, S. Johansson, and T. Öberg. Relaxed versus activated stump muscles during casting for trans-tibial prostheses. *Prosthetics and Orthotics International*, 23(1), 13–20, 1999.

[4] J.C. Cagle et al. A finite element model to assess transtibial prosthetic sockets with elastomeric liners. *Medical & Biological Engineering & Computing*, 56(7), 1227–1240, 2018.

[5] S. Avril et al. Mixed experimental and numerical approach for characterizing the biomechanical response of the human leg under elastic compression. *Journal of Biomechanical Engineering*, 132(3), 031006, 2010.

[6] W. Zeng et al. Modeling of active skeletal muscles: a 3D continuum approach incorporating multiple muscle interactions. *Frontiers in Bioengineering and Biotechnology*, 11, 1153692, 2023.

[7] S.H. Yeo et al. Numerical instability of Hill-type muscle models. *Journal of the Royal Society Interface*, 20(199), 20220430, 2023.

[8] T.K. Uchida and S.L. Delp. Biomechanics of Movement: The Science of Sports, Robotics, and Rehabilitation. Cambridge, MA, USA: The MIT Press, 2021.

[9] T. Siebert and C. Rode. Computational modeling of muscle biomechanics. In: *Computational Modelling of Biomechanics and Biotribology in the Musculoskeletal System*. Ed. by Zhongmin Jin. Cambridge, United Kingdom: Woodhead Publishing, 2014, pp. 173–204.

[10] M. Millard et al. Flexing computational muscle: modeling and simulation of musculotendon dynamics. *Journal of Biomechanical Engineering*, 135(2), 021005, 2013.

[11] G. Giantesio, A. Musesti, and D. Riccobelli. A comparison between active strain and active stress in transversely isotropic hyperelastic materials. *Journal of Elasticity*, 137(1), 63–82, 2019.

[12] T. Klotz, C. Bleiler, and O. Röhrle. A physiology-guided classification of active-stress and active-strain approaches for continuum-mechanical modeling of skeletal muscle tissue. *Frontiers in Physiology*, 12, 685531, 2021.

[13] C. Rode et al. The effects of parallel and series elastic components on the active cat soleus force-length relationship. *Journal of Mechanics in Medicine and Biology*, 9(1), 105–122, 2009.

[14] T.L. Allinger, M. Epstein, and W. Herzog. Stability of muscle fibers on the descending limb of the force-length relation. A theoretical consideration. *Journal of Biomechanics*, 29(5), 627–633, 1996.

[15] G.I. Zahalak. Can muscle fibers be stable on the descending limbs of their sarcomere length-tension relations? *Journal of Biomechanics*, 30(11–12), 1179–1182, 1997.

[16] T.C.T. Ting. Positive definiteness of anisotropic elastic constants. *Mathematics and Mechanics of Solids*, 1(3), 301–314, 1996.

[17] T. Heidlauf et al. A continuum-mechanical skeletal muscle model including actin-titin interaction predicts stable contractions on the descending limb of the force-length relation. *PLoS Computational Biology*, 13(10), e1005773, 2017.

[18] V. Joumaa, T.R. Leonard, and W. Herzog. Residual force enhancement in myofibrils and sarcomeres. *Proceedings of the Royal Society B: Biological Sciences*, 275(1641), 1411–1419, 2008.

[19] T.R. Leonard, M. DuVall, and W. Herzog. Force enhancement following stretch in a single sarcomere. *American Journal of Physiology-Cell Physiology*, 299(6), C1398–C1401, 2010.

[20] I.A. Telley et al. Dynamic behaviour of half-sarcomeres during and after stretch in activated rabbit psoas myofibrils: sarcomere asymmetry but no 'sarcomere popping'. *Journal of Physiology*, 573(1), 173–185, 2006.

[21] V. Joumaa et al. Passive force enhancement in single myofibrils. *Pflügers Archiv-European Journal of Physiology*, 455(2), 367–371, 2007.

[22] E.-J. Lee and W. Herzog. Residual force enhancement exceeds the isometric force at optimal sarcomere length for optimized stretch conditions. *Journal of Applied Physiology*, 105(2), 457–462, 2008.

[23] A. Tomalka et al. The active force–length relationship is invisible during extensive eccentric contractions in skinned skeletal muscle fibres. *Proceedings of the Royal Society B: Biological Sciences*, 284(1854), 20162497, 2017.

[24] B.C. Abbott and X.M. Aubert. The force exerted by active striated muscle during and after change of length. *Journal of Physiology*, 117(1), 77–86, 1952.

[25] W. Herzog and T.R. Leonard. Force enhancement following stretching of skeletal muscle: a new mechanism. *Journal of Experimental Biology*, 205(9), 1275–1283, 2002.

[26] T.R. Leonard and W. Herzog. Regulation of muscle force in the absence of actin-myosin-based cross-bridge interaction. *American Journal of Physiology-Cell Physiology*, 299(1), C14–C20, 2010.

[27] T. Siebert et al. Three-dimensional muscle architecture and comprehensive dynamic properties of rabbit gastrocnemius, plantaris and soleus: input for simulation studies. *PLoS One*, 10(6), e0130985, 2015.

[28] D.R. Peterson, D.E. Rassier, and W. Herzog. Force enhancement in single skeletal muscle fibres on the ascending limb of the force–length relationship. *Journal of Experimental Biology*, 207(16), 2787–2791, 2004.

[29] T. Siebert et al. Does weightlifting increase residual force enhancement? *Journal of Biomechanics*, 49(10), 2047–2052, 2016.

[30] W. Herzog. Skeletal muscle mechanics: questions, problems and possible solutions. *Journal of Neuroengineering and Rehabilitation*, 14(1), 98, 2017.

[31] C. Stronczek et al. The N2A region of titin has a unique structural configuration. *The Journal of General Physiology*, 153(7), e202012766, 2021.

[32] H. Granzier and S. Labeit. Structure–function relations of the giant elastic protein titin in striated and smooth muscle cells. *Muscle Nerve*, 36(6), 740–755, 2007.

[33] D. Labeit et al. Calcium-dependent molecular spring elements in the giant protein titin. *Proceedings of the National Academy of Sciences*, 100(23), 13716–13721, 2003.

[34] W. Herzog et al. Residual force enhancement following eccentric contractions: a new mechanism involving titin. *Physiology*, 31(4), 300–312, 2016.

[35] W. Herzog et al. A new paradigm for muscle contraction. *Frontiers in Physiology*, 6, 174, 2015.

[36] C. Rode, T. Siebert, and R. Blickhan. Titin-induced force enhancement and force depression: a 'sticky-spring' mechanism in muscle contractions? *Journal of Theoretical Biology*, 259(2), 350–360, 2009.

[37] K.C. Nishikawa et al. Is titin a 'winding filament'? A new twist on muscle contraction. *Proceedings of the Royal Society B: Biological Sciences*, 279(1730), 981–990, 2012.

[38] G. Schappacher-Tilp et al. A novel three-filament model of force generation in eccentric contraction of skeletal muscles. *PLoS One*, 10(3), e0117634, 2015.

[39] M. Millard, D.W. Franklin, and W. Herzog. A three filament mechanistic model of musculotendon force and impedance. *eLife*, 12, RP88344, 2024.

[40] K. Nishikawa. Titin: a tunable spring in active muscle. *Physiology*, 35(3), 209–217, 2020.

[41] W. Herzog. Passive force enhancement in striated muscle. *Journal of Applied Physiology*, 126(6), 1782–1789, 2019.

[42] R. Lemos et al. Realistic skeletal muscle deformation using finite element analysis. In: *Proc. XIV Brazilian Symp. on Computer Graphics and Image Processing*. Florianopolis, Brazil, 2001, pp. 192–199.

[43] R. Seydewitz, T. Siebert, and M. Böl. On a three-dimensional constitutive model for history effects in skeletal muscles. *Biomechanics and Modeling in Mechanobiology*, 18(6), 1665–1681, 2019.

[44] T. Heidlauf et al. A multi-scale continuum model of skeletal muscle mechanics predicting force enhancement based on actin–titin interaction. *Biomechanics and Modeling in Mechanobiology*, 15(6), 1423–1437, 2016.

[45] G.A. Holzapfel. Nonlinear Solid Mechanics: A Continuum Approach for Engineering. Chichester, England: John Wiley & Sons, 2000.

[46] E.K. Moo, T.R. Leonard, and W. Herzog. In vivo sarcomere lengths become more non-uniform upon activation in intact whole muscle. *Frontiers in Physiology*, 8, 1015, 2017.

[47] S.A. Maas et al. FEBio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering*, 134(1), 011005, 2012.

[48] K. Trombitás et al. PEVK extension of human soleus muscle titin revealed by immunolabeling with the anti-titin antibody 9D10. *Journal of Structural Biology*, 122(1–2), 188–196, 1998.

[49] L. Tskhovrebova et al. Elasticity and unfolding of single molecules of the giant muscle protein titin. *Nature*, 387(6630), 308–312, 1997.

[50] M. Rief, J.M. Fernandez, and H.E. Gaub. Elastically coupled two-level systems as a model for biopolymer extensibility. *Physical Review Letters*, 81(21), 4764–4767, 1998.

[51] W.A. Linke et al. Nature of PEVK-titin elasticity in skeletal muscle. *Proceedings of the National Academy of Sciences*, 95(14), 8052–8057, 1998.

[52] J.A. Herzog et al. Are titin properties reflected in single myofibrils? *Journal of Biomechanics*, 45(11), 1893–1899, 2012.

[53] G. Schappacher-Tilp et al. Computing average passive forces in sarcomeres in length-ramp simulations. *PLoS Computational Biology*, 12(6), e1004904, 2016.

[54] H. Li et al. Reverse engineering of the giant muscle protein titin. *Nature*, 418(6901), 998–1002, 2002.

[55] L.G. Prado et al. Isoform diversity of giant proteins in relation to passive and active contractile properties of rabbit skeletal muscles. *The Journal of General Physiology*, 126(5), 461–480, 2005.

[56] V. Joumaa et al. The origin of passive force enhancement in skeletal muscle. *American Journal of Physiology-Cell Physiology*, 294(1), C74–C78, 2008.

[57] B.H. Dalton, V.S. Contento, and G.A. Power. Residual force enhancement during submaximal and maximal effort contractions of the plantar flexors across knee angle. *Journal of Biomechanics*, 78, 70–76, 2018.

[58] A.E. Oskouei and W. Herzog. Observations on force enhancement in submaximal voluntary contractions of human adductor pollicis muscle. *Journal of Applied Physiology*, 98(6), 2087–2095, 2005.

[59] F.E. Zajac. Muscle and tendon: properties, models, scaling, and application to biomechanics and motor control. *Critical Reviews in Biomedical Engineering*, 17(4), 359–411, 1989.

[60] A.E. Ehret, M. Böl, and M. Itskov. A continuum constitutive model for the active behaviour of skeletal muscle. *Journal of the Mechanics and Physics of Solids*, 59(3), 625–636, 2011.

[61] M.L. Sampaio de Oliveira and T.K. Uchida. Muscle constitutive model with a tangent modulus approximation: Ansys implementation and verification. *Journal of Biomechanical Engineering*, 145(7), 071002, 2023.

[62] J.L. Hicks et al. Is my model good enough? Best practices for verification and validation of musculoskeletal models and simulations of movement. *Journal of Biomechanical Engineering*, 137(2), 020905, 2015.

[63] M.M. Mehrabadi and S.C. Cowin. Eigentensors of linear anisotropic elastic materials. *The Quarterly Journal of Mechanics and Applied Mathematics*, 43(1), 15–41, 1990.

[64] M. Böl and S. Reese. Micromechanical modelling of skeletal muscles based on the finite element method. *Computer Methods in Biomechanics and Biomedical Engineering*, 11(5), 489–504, 2008.

[65] C.M. Eng et al. Scaling of muscle architecture and fiber types in the rat hindlimb. *Journal of Experimental Biology*, 211(14), 2336–2345, 2008.

[66] G. Chagnon, M. Rebouah, and D. Favier. Hyperelastic energy densities for soft biological tissues: a review. *Journal of Elasticity*, 120(2), 129–160, 2015.

[67] F. López Jiménez. Modeling of soft composites under three-dimensional loading. *Composites Part B: Engineering*, 59, 173–180, 2014.

[68] L.A. Spyrou, M. Agoras, and K. Danas. A homogenization model of the Voigt type for skeletal muscle. *Journal of Theoretical Biology*, 414, 50–61, 2017.

[69] C.P. McGowan, R.R. Neptune, and W. Herzog. A phenomenological model and validation of shortening-induced force depression during muscle contractions. *Journal of Biomechanics*, 43(3), 449–454, 2010.

5

Pressure Between Residual Limb and Transtibial Prosthesis Socket Considering Muscle Contraction: a Finite Element Study

The third main contribution of this thesis is the creation of a finite element model of a residual limb and transtibial prosthesis socket considering muscle contraction. We investigated the effect of muscle contraction on the residuum–socket interfacial pressure, which was the main research question of this thesis. We created a finite element model that included the gastrocnemius muscle and analyzed how the presence and activity of the muscle affects the residuum–socket interfacial pressure at key instants of gait.

The finite element model was created using the geometry of a limb following transtibial amputation, which was obtained from the OpenLimb repository [1]. We augmented this model with a gastrocnemius muscle using the muscle material model described in Chapter 4. Simulations were performed on eight cores of an Intel® Core i9 13900H, 2.60

GHz processor; each simulation required approximately 8 hours of wall-clock time. We faced several challenges during the development of the model and generation of the simulations. Notably, elements of the residuum anterior to the patella and inferior to the cut end of the fibula exhibited high distortion. We changed the order of the elements from quadratic to linear (lower-order elements tend to be stiffer, resulting in less deformation) and reduced the size of the elements to compensate for the reduction in the number of nodes per element. We performed a sensitivity analysis to explore the effect of changes in the muscle model parameters with the greatest uncertainty.

The main findings from this study revealed that muscle activation increases pressures in critical regions of the residuum. For example, the pressure in the popliteal fossa region was up to 42% greater when the muscle was active rather than passive. This study highlighted the importance of considering muscle contraction in prosthesis socket design. Most existing models ignore muscle contraction, which may affect the conclusions that are drawn with respect to the fit of the prosthesis. Our simulations also demonstrated that the position and orientation of the socket can change when muscle contraction is considered. This study emphasizes the need to incorporate muscle activity in future studies of residuum–socket fit.

We presented this work at the 3rd Annual Ottawa Musculoskeletal Research Symposium (Sampaio de Oliveira, M.L., Uchida, T.K. Finite element model with transtibial amputation considering the force enhancement behaviour of muscle, 3rd Annual Ottawa Musculoskeletal Research Symposium, October 11, 2024, Ottawa, Canada). The journal paper has been submitted and is currently under review; the content of this paper is provided in the remainder of this chapter.

5.1 Abstract

A lower-limb prosthesis is a mechanical device that replaces part of a biological limb to restore mobility. The tissues in the residual limb (the “residuum”) experience atypical stresses. Importantly, the distal end of the residuum must bear the load that would typ-

ically be borne by the foot. To avoid pain and tissue damage, the part of the prosthesis that interfaces with the residuum (the "socket") must be designed to fit properly not only at rest but also during walking and other daily activities. The shape of the residuum changes during movement as muscles contract, which can affect the fit of the socket. To study the residuum–socket interface, we created the first finite element model that considers the effect of muscle contraction on the residuum–socket interfacial pressure. During gait, the interfacial pressure at heel strike and toe off was greater in key regions of the residuum when the gastrocnemius muscle was active rather than passive. The greatest difference was observed in the popliteal fossa region during heel strike, where the pressure was 42% greater when the muscle was fully active than when passive. Our results provide insight into how muscle contraction affects socket fit throughout the gait cycle, potentially leading to improvements in the design and manufacture of prosthesis sockets.

Keywords: Finite element, Muscle contraction, Prosthesis, Residuum, Socket, Transtibial amputation.

5.2 Introduction

Approximately 40 million people live with amputation [2]. The most common causes of amputation are vascular disease, trauma, cancer, and congenital anomalies [3]. Lower-limb amputation profoundly affects an individual's mobility. The most common sites of lower-limb amputation are the toe (33.2%), below the knee (transtibial; 28.2%), above the knee (transfemoral; 26.1%), and the foot (10.6%) [4]. During a transtibial amputation surgery, a soft-tissue pad (muscle flap) is created by wrapping the existing muscle around the distal end of the bone at the amputation site. The lack of an adequate muscle flap can lead to skin breakdown, soft-tissue infection, and a need for revision surgery [5].

Over 80% of individuals with lower-limb amputation use a prosthesis [6]. Despite

having different shapes and designs, transtibial prostheses generally have the same five components: the socket, liner, suspension, structure (or pylon), and foot/ankle [7]. The socket connects the prosthesis to the residual limb (the "residuum") and is designed to provide a comfortable fit, proper transmission of loads, and stability. Prosthetic liners are inserted between the residuum and socket to improve comfort and provide cushioning for the loads that are transmitted between the residuum and the socket [8]. The suspension is responsible for keeping the socket attached to the residuum [9]. The pylon allows one to adjust the prosthesis to the height of the individual and to align the prosthesis components. Finally, the foot/ankle is designed to mimic the mechanical function of the anatomical foot and ankle [7].

The two main design approaches for transtibial sockets are specific surface bearing and total surface bearing. In the specific surface bearing design (or "hands-on casting"), the socket is manually modified so that it transfers loads to regions of the residuum that are load tolerant. One example of this type is the patellar tendon bearing design [10], which accounts for approximately 18% of sockets [11]. The total surface bearing design (or "hands-off casting") is a more recently developed approach that aims to achieve a more uniform load transmission over the entire surface of the residuum, thereby reducing the peak pressure [12].

The current process for socket manufacturing is time consuming and depends on the prosthetist's experience [13]. On average, prosthesis users visit the prosthetist nine times during the first year of use for adjustments to the prosthesis to improve fit [14]. An improper fit can result in pain, discomfort, and a reduction in the quality of life of the wearer [3]. Approximately 57% of prosthesis users report having experienced moderate to severe pain in the residuum [14]; however, even mild pain in the residuum can result in changes to posture and gait, which can then lead to back pain and reduced mobility [15]. The design of prosthetic sockets may be improved by complementing the experience of the prosthetist with analyses from numerical simulations.

Finite element analysis (FEA) is a numerical method in which a physical system is

divided into small parts called finite elements. The dynamics of each of these elements are governed by a constitutive model that describes the material's behaviour mathematically. FEA has been used to generate numerical simulations of the residuum–socket interaction for predicting internal tissue deformation and skin damage. For example, Portnoy et al. [16] developed a finite element model to calculate the strain in the tissue of the lower-limb muscles to characterize deep tissue injury. Cagle et al. [11] used a finite element model to compare pressure peaks on the surface of the residuum with skin breakdown developed by prosthesis users.

Finite element models can also facilitate the evaluation of candidate socket designs, new materials, and various loading conditions without creating physical prototypes or performing experiments. Frillici et al. [17] proposed the use of FEA to help prosthetists customize sockets according to the stresses predicted by simulations. Additionally, Frillici et al. proposed a shape-optimization strategy to minimize the weight of the socket and to reduce the peak pressure on the residuum to be below the pain threshold. Colombo et al. [18] proposed a strategy for designing patient-specific sockets in which FEA is used to guide technicians as they adjust the socket shape until the peak pressure is below the pain threshold. Moerman et al. [19] proposed a strategy to design a socket whose elastic properties vary over its surface according to a mapping function that correlates the pressure on the residuum with the mechanical properties of the socket. Regardless of the uses of finite element models—whether evaluating possible regions of deep tissue injury and skin breakdown or assessing proposed socket designs—they rely on the stress–strain behaviour predicted by the model. To design sockets that are well accepted by prosthesis users, finite element models must accurately characterize the stresses experienced by the residuum.

Changes in the volume and shape of the residuum can change the stresses on its surface. The volume of the residuum can change dramatically in the period between 12 and 18 months following an amputation surgery [20]. However, even after the post-operative period, the residuum undergoes volume and shape changes. Short-term volume

changes occur throughout the day due to the movement of interstitial fluids when the residuum experiences pressure and shear stress [21, 22]. The residuum also undergoes rapid changes in shape due to the activation of muscles within it [23, 24], which are typically still innervated [25]. If the volume of the residuum decreases, it can move within the socket and increase the stress at the distal end [3]. On the other hand, if the volume of the residuum increases, the stress on the skin can increase in regions where the cross-section increases [24]. Consequences of these increases in stress include discomfort, pain, and skin ulceration [26].

Muscle activity plays an important role in how the socket interfaces with the residuum. Lilja et al. [23] questioned whether prosthetists should consider the effect of muscle contraction when fitting a prosthesis. Lilja et al. also acknowledged that, as of the late 1990s, only a few experimental studies had explored how muscle contraction affects prosthesis fit. More recently, this statement was corroborated by Paternò et al. [13] and Solav et al. [24], who confirmed the need to consider muscle shape changes during socket design.

To the best of the authors' knowledge, muscle contraction is ignored in all existing finite element simulations that study the interface between the residuum and a transtibial prosthesis socket. Lin et al. [27] created a static model to study how the stiffness of the liner affects the pressure on the residuum during stance, considering all soft tissues as a single homogeneous linearly elastic material, and concluded that the pressure was moderately sensitive to the liner stiffness. Jia et al. [28] created a quasi-dynamic model with loads from walking to predict the pressure on the residuum, also considering all soft tissues as a single homogeneous linearly elastic material, and found that the greatest pressure was located on the popliteal fossa. Cagle et al. [11] developed a static model with loads from stance and heel strike to predict the pressure on the residuum, again using a single homogeneous linearly elastic material to represent the entire residuum, and found good agreement between the regions of high pressure predicted by the model and the locations of skin breakdown observed clinically. Steer et al. [29] created a quasi-static model to investigate how the choice of material models and parameters, contact

parameters, donning simulation method, and other modelling inputs affect the pressure on the residuum surface. Steer et al. concluded that modelling decisions should be justified based on the clinical frame of reference, hyperelastic material models produce more numerically stable simulations, and the explicit donning simulation method (where the socket slides onto the residuum) is suitable for studying shear stress whereas the overclosure donning simulation method (where the socket is fixed in a donned position and the residuum deforms within it) is not.

In this study, we present a finite element model to investigate how muscle contraction affects the pressure on the residuum surface. We use and modify the OpenLimb [1] model for our model geometry. We augment this model with a gastrocnemius muscle whose dynamic behaviour is governed by a constitutive muscle material model that exhibits force enhancement. Force enhancement has been observed at all structural levels of a muscle [30, 31] and increases the force produced by the muscle during active stretching when compared to the force produced isometrically (i.e., when the muscle is held at a constant length). Considering force enhancement guarantees a positive material stiffness during active stretching, which is important for maintaining numerical stability [32]. We use the muscle material model from Sampaio de Oliveira and Uchida [33], which is a transversely isotropic, hyperelastic material that exhibits force enhancement and is able to capture changes in muscle stiffness when the muscle is activated. We simulate two load profiles representing the conditions during the heel-strike and toe-off events of gait. We compare the pressure obtained from our finite element model when the muscle is active versus passive and perform a sensitivity analysis to assess the reliability of our simulation results. We have shared our geometric models at https://simtk.org/projects/fea_limb_socket so that others can reproduce and extend our results.

5.3 Methodology

Three finite element models were created to study how muscle contraction affects the pressure on the residuum surface during gait. One model was similar to those considered

elsewhere in the literature, where all soft tissue in the residuum was treated as a single body represented by a single material; we refer to this model as the Muscleless Model. The other two models we created include within the residuum a gastrocnemius muscle of two sizes, which we refer to as the Standard-Size-Muscle Model and the Small-Muscle Model. The pressure on the residuum surface was compared among these models. The models were developed in the finite element software Ansys 2023 R2 (Ansys, Canonsburg, PA, USA) and the model geometry was created in Ansys SpaceClaim 2023 R2 (SpaceClaim Corp., Concord, MA, USA).

5.3.1 Geometry

The geometry of the Standard-Size-Muscle Model and the Small-Muscle Model comprised four parts: the residuum, liner, socket, and muscle. The geometry of the residuum was created using the model of a residuum following a transtibial amputation surgery, obtained from the OpenLimb project [1]. OpenLimb is a repository containing an open-access statistical shape model of a residuum (right leg). For our simulations, we used the model with the mean values from the principal modes of variation, which are amputation height, overall size, soft-tissue bulk, and relative bone–soft tissue length. To reduce the computational burden, we replaced the bones in the model (the femur, patella, tibia, and fibula) with fixed boundary conditions on the adjoining soft tissues [11]; we refer to these empty spaces as bone hollows.

The liner and socket geometry were created from the residuum geometry. The liner thickness was based on the Össur Dermo Liner (Reykjavík, Iceland), which has a 3 mm thickness. This liner is commonly used and has properties similar to those of other popular liners [11]. The liner geometry was created by offsetting the residuum skin surface by 3 mm. Similarly, the geometry of the socket was created by offsetting the outer surface of the liner by 6 mm [19, 34]. A cut line was used to define the socket rim, with the cut located on the anterior side of the residuum below the patellar tendon region and on the posterior side below the popliteal fossa [19]. The geometry of the residuum,

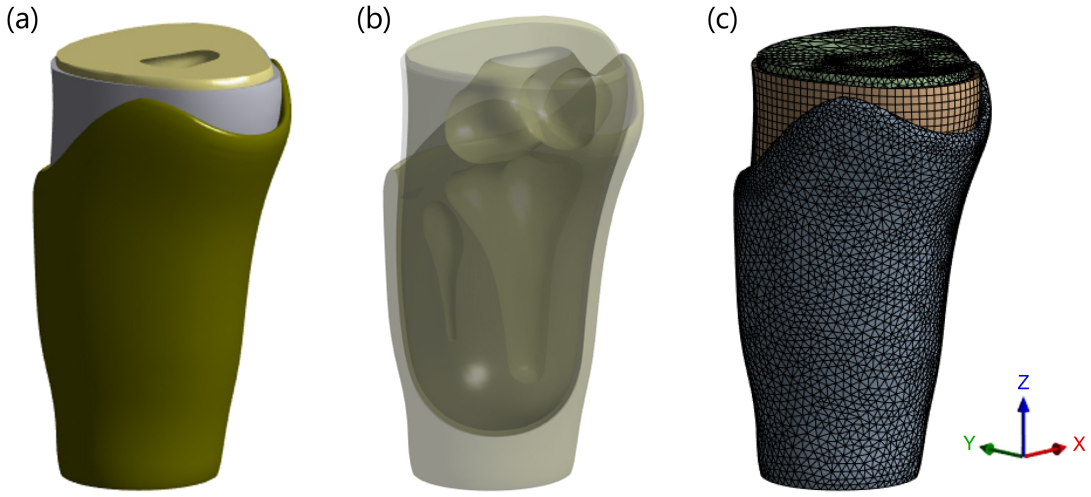

Figure 5.1: Geometry used in our finite element simulations. (a) The residuum and liner. (b) Bones within the residuum. (c) Converged mesh with average element size of 4.5 mm. The reference frame shown is the global coordinate system: X-axis, lateral–medial direction (medial is positive); Y-axis, anterior–posterior direction; Z-axis, inferior–superior direction.

liner, and bone hollows are shown in Figs. 5.1(a) and 5.1(b).

The geometry of the muscle in the Standard-Size-Muscle Model was based on the geometry of a healthy gastrocnemius (including medial and lateral heads) from Andreassen et al. [35]. During amputation surgery, a muscle flap is created by wrapping the gastrocnemius around the distal end of the residual tibia from the posterior side to the anterior side [25], which we approximated in our model. Depending on the surgical technique, the soleus muscle may be partially debulked or even completely removed during transtibial amputation [25, 36]; thus, we included in our model only the gastrocnemius muscle. First, the healthy muscle geometry was sliced vertically in 2 cm increments to create two-dimensional boundaries using closed splines. The splines were then lofted and smoothed to create the muscle volume. Finally, we used Ansys SpaceClaim to wrap the geometry around the distal end of the residual tibia to approximate a muscle flap. The resulting muscle geometry is shown in Fig. 5.2(a) for the Standard-Size-Muscle Model. (Note that the geometry of the muscle plus the surrounding soft tissue are equivalent to the single residuum geometry in the Muscleless Model.) Because the geometry of the

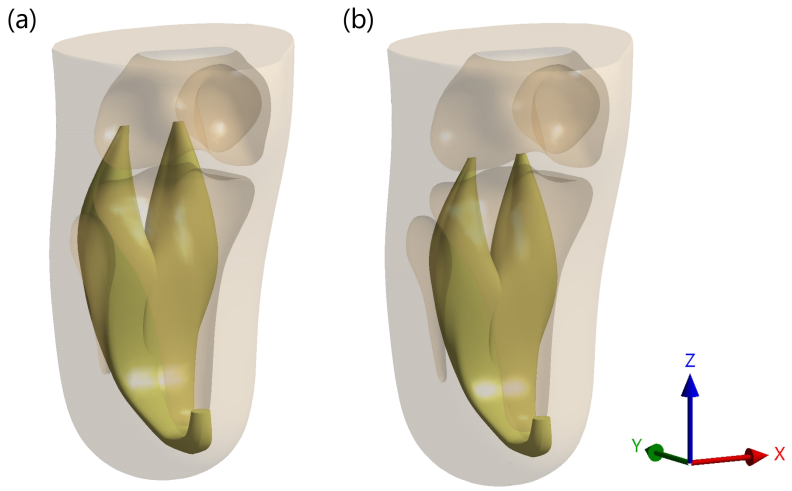

Figure 5.2: Muscle geometry created for our finite element models. (a) Standard-Size-Muscle Model. (b) Small-Muscle Model, in which the muscle from the Standard-Size-Muscle Model was scaled to 90% of its original size.

muscle was an approximation and varies among individuals, we scaled its size by 90% to create the Small-Muscle Model (Fig. 5.2(b)), which we used to explore the effect of muscle size on the pressure between the residuum and the socket.

5.3.2 Material models

The socket and liner were modelled as linear materials. The remaining residuum, excluding the approximated muscle, was modelled as a hyperelastic material due to its nonlinear stress–strain behaviour using a Neo-Hookean constitutive model. The parameter values used for the liner, socket, and residuum materials were obtained from Steer et al. [29].

The muscle was modelled as a nearly incompressible hyperelastic material with the ability to contract upon activation. We used the model of Sampaio de Oliveira and Uchida [33], which considers the force enhancement behaviour of muscle. The stress–stretch behaviour in the fiber direction depends on four parameters: maximum isometric stress ($\sigma_{\max}$); optimal fiber stretch (λ_{opt}), which is the ratio of the fiber length at which the maximum isometric stress occurs to the fiber length at the beginning of the simulation; fiber stretch (relative to the initial fiber length) at the onset of activation (λ_{act}); and

Table 5.1: Parameters for the socket, liner, and residuum material models [29] and muscle material model [33, 37].

Geometry	Parameter	Value
Socket	Young's modulus, E	1500 MPa
	Poisson's ratio, ν	0.39
Liner	Young's modulus, E	0.384 MPa
	Poisson's ratio, ν	0.4992
Residuum	Shear modulus, C	8.075 kPa
	Incompressibility parameter, D	1.234 MPa^{-1}
Muscle	Maximum isometric stress, $\sigma_{\max}$	73 kPa
	Along-fiber shear modulus, G_1	50 MPa
	Cross-fiber shear modulus, G_2	50 MPa
	Effective bulk modulus, K	50 MPa

muscle activation (α). Maximum isometric stress ($\sigma_{\max}$) was obtained from Spyrou et al. [37]. The muscle material model contains three additional parameters that are not related to the fiber stretch: the along- and cross-fiber shear moduli (G_1 and G_2), and the effective bulk modulus (K). Preliminary tests using the parameters from Sampaio de Oliveira and Uchida [33] did not converge due to excessive shear deformation of the muscle tissue. Therefore, we increased the shear moduli from $G_1 = G_2 = 50$ kPa by factors of 10 until convergence was achieved, which occurred at a value of 50 MPa. We confirmed that the near-incompressibility property was preserved by noting that the volume ratio of each element in the muscle of the Small-Muscle Model remained between 0.98 and 1.04 in the simulations of both heel strike and toe off. The muscle material model parameters are provided in Table 5.1.

The parameters λ_{opt}, λ_{act}, and α have high uncertainty. Huang and Ferris [38] used electromyography to measure the activity of the muscles in the residuum during gait and found high variability among individuals, with activations ranging from 0.12 to 0.95 at toe off. The parameters λ_{opt} and λ_{act} are also difficult to determine precisely in vivo (e.g., using an optical microendoscope [39]). Therefore, we varied these parameters to study their effect on the residuum–socket pressure. In our baseline models, we set λ_{opt},

λ_{act}, and α to 1.

The direction of each fiber at the beginning of the simulation with respect to the global coordinate system (the "initial fiber direction") is an input to the constitutive model. We used the Fiber Generator Tool in FEBio [40] to define the initial fiber directions. The tool first generated a scalar field that was defined by solving a Poisson-type boundary-value problem. The fiber direction vector for each element was then determined by computing tangents of this scalar field in the center of the element.

5.3.3 Mesh

Unique meshing settings were determined for each part of the model to accommodate the size and geometry of each part. The socket and liner were modelled with linear tetrahedral and hexahedral elements ("SOLID185" in Ansys). The residuum and muscle were modelled with quadratic tetrahedral elements ("SOLID187") in our simulations of heel strike; linear tetrahedral elements were used in our simulations of toe off as these simulations did not converge using quadratic elements (see Discussion). The software method to mesh the residuum, muscle, and socket was set to "Patch Conforming," which resulted in a mesh that preserved the details of their complex geometries; the "Sweep" method was used for the liner because of its simple geometry.

We used our Muscleless Model at heel strike to perform a convergence analysis for determining appropriate element sizes. The initial element size for all parts was set to 6 mm. This size was reduced in increments of 0.5 mm until a difference of less than 1% was observed in the average pressure on the residuum when compared to the average pressure predicted using the previous, coarser mesh. The converged element size setting was 4.5 mm. Smaller elements were necessary in regions requiring a finer mesh to capture the details of the geometry, such as the socket rim and the fibula bone hollow (Table 5.2). Additionally, during our simulations of toe off, smaller elements were used to address numerical convergence issues arising from the greater deformation of elements in the residuum. To balance accuracy with computational efficiency, the element size was

Table 5.2: Element size settings used to create the meshes in our Muscleless, Standard-Size-Muscle, and Small-Muscle Models. Smaller elements were necessary in regions with more detailed geometry and during toe off to achieve convergence.

	Element size setting (mm)			
Region	Without muscle (heel strike)	Without muscle (toe off)	With muscle (heel strike)	With muscle (toe off)
Bone hollow: femur	4.5	4.5	4.5	4.5
Bone hollow: patella	4.5	2.5	4.5	2.5
Bone hollow: tibia	4.0	4.0	4.0	4.0
Bone hollow: fibula	2.0	2.0	2.0	2.0
Muscle excluding endcaps	–	–	4.5	4.5
Muscle proximal endcap	–	–	1.5	1.0
Muscle distal endcap	–	–	2.0	2.0
Residuum excluding endcap	4.5	3.2	4.5	3.2
Residuum proximal endcap	4.5	4.0	4.5	4.0
Liner	4.5	4.0	4.2	4.0
Socket excluding rim	5.0	5.0	5.0	5.0
Socket rim	3.2	3.2	3.2	3.2

increased to 5.0 mm for the socket. In our simulations of heel strike, the converged mesh had 123,473 elements for the Muscleless Model, 143,739 elements for the Standard-Size-Muscle Model, and 135,376 elements for the Small-Muscle Model. In our simulations of toe off, the converged mesh had 188,194 elements for the Muscleless Model, 229,281 elements for the Standard-Size-Muscle Model, and 203,826 elements for the Small-Muscle Model.

5.3.4 Contact and boundary conditions

Three contacting surfaces were defined in our model: between the socket and liner, between the liner and residuum, and between the muscle and the surrounding soft tissue. The socket–liner and liner–residuum interfaces were set to "frictional" in Ansys with coefficients of friction $\mu = 0.5$ and $\mu = 2.0$, respectively [11, 29]. According to Cagle et al. [11], these values approximate "bonded contact" (i.e., no sliding or separation between faces or edges of the elements), which is appropriate to enhance mechanical coupling

between the residuum and the prosthesis [41]. The interface between the muscle and the surrounding soft tissue was set to "Shared Topology" in Ansys SpaceClaim, following the approach of Rayward and Little [42] for modelling contact between soft tissues. This setting assumes "bonded contact" between the two soft-tissue geometries and avoids penetration errors at their interface.

Unique boundary conditions were defined for each of the three load cases we explored: heel strike, midstance, and toe off. These events correspond to approximately 10%, 50%, and 80% of the stance phase during gait. We used the experimental loads reported by Koehler et al. [43] after scaling them by the approximate mass of our modelled participant. (The same experimental loads were used by Steer et al. [29].) The model from Plesec and Harih [34] was used to approximate the mass of our participant. They provided the average mass (77 kg) and dimensions (mid-patella circumference of 368 mm and tibia length of 150 mm) from a population of European males extracted from the AnyBody Managed Model Repository [44]. The mid-patella circumference of our model was 353.5 mm and the tibia length was 147 mm; thus, we assumed a body mass of approximately 72 kg. Forces and moments in only the sagittal plane were applied as they are the dominant loads at heel strike, midstance, and toe off. Loads were applied relative to a coordinate system that was created from the tibia's principal axes (Table 5.3). The scaled forces and moments (Table 5.4) were applied to the inferior tip of the socket. We applied the heel-strike, midstance, and toe-off load cases to the Muscleless Model to compare our results to those of Steer et al. [29]; however, we considered only the heel-strike and toe-off load cases to study how the interfacial pressure is affected by muscle contraction.

5.3.5 Simulation steps

We divided our simulation into three steps to ramp up the loads and muscle activation, and to resolve any contact penetration. Before the simulation began, the tip of the socket was positioned 20 mm inferior to the tip of the exterior surface of the liner; we explicitly

Table 5.3: Principal axes of the tibia. The vector directions are calculated with respect to the global coordinate system, as shown in Fig. 5.1.

Axis	Anatomical direction	Vector direction
X	Inferior (−) to superior (+)	$[0.0367, -0.1356, 0.9901]$
Y	Lateral (−) to medial (+)	$[0.9988, 0.0351, -0.0322]$
Z	Posterior (−) to anterior (+)	$[-0.0365, 0.9901, 0.1368]$

Table 5.4: Loads applied to the inferior tip of the socket with respect to the tibia's principal axes. Experimental loads from Koehler et al. [43] were scaled by the approximate mass of the participant (72 kg).

	Load case		
Load	Heel strike	Midstance	Toe off
F_X (N)	618	660	730
F_Z (N)	−76	−63	90
M_Y (N·m)	5.4	−0.09	−45

simulated the donning process, following the strategy used by Jamaludin et al. [45] and Steer et al. [29]. Throughout all simulation steps, the bone hollows were fixed in all directions. Each time step had a pseudo-duration of 1 s and advanced in increments of 0.02 s. The three simulation steps are summarized as follows:

1. Overclosure: Penetration between bodies is resolved. Bone hollows and the exterior surface of the socket are fixed.

2. Donning: The socket is displaced superiorly by 20 mm. Bone hollows are fixed.

3. Loading: External loads and muscle activation are ramped up. Bone hollows are fixed, loads are applied to the base of the socket, and the muscle is activated.

5.3.6 Sensitivity analysis

A sensitivity analysis was performed to assess the effect of modelling uncertainties (specifically, the muscle geometry and material model parameters) on the quantities of interest in this study. Three values for each of the parameters λ_{opt}, λ_{act}, and α were used in

the sensitivity analysis with both the Small-Muscle Model and the Standard-Size-Muscle Model. Optimal fiber stretch (λ_{opt}) was set to 0.91, 1.00, or 1.10, which reside on the descending limb, plateau, and ascending limb of the active force–length curve, respectively [46]. Fiber stretch at the onset of activation (λ_{act}) was set to 1.0, 1.1, or 1.2, which correspond to fiber lengths on the plateau and descending limb of the active force–length curve at the onset of activation. Muscle activation (α) was set to 0.1, 0.5, or 1.0, which spans the range of activity observed in the muscles of the residuum during gait (0.12 to 0.95) [38]. We varied the parameters individually while leaving the other parameters set to their baseline value of 1, and considered the heel-strike and toe-off loading cases for both the Small-Muscle Model and the Standard-Size-Muscle Model. In total, 36 simulations were performed for the sensitivity analysis (9 parameter values $\times$ 2 muscle sizes $\times$ 2 load cases).

5.3.7 Model analysis

The simulations were analyzed by collecting the pressure of each element in five regions of clinical significance and calculating the 95th percentile of the pressure distribution. We chose this metric because our Muscleless Model was compared against the results of Steer et al. [29], which were reported using the same metric. We considered five regions on the residual limb that are associated with socket fit and comfort [47]: popliteal fossa, fibular head, tibial tuberosity, patellar tendon, and residuum tip (Fig. 5.3).

5.4 Results

The results of our simulations are organized as follows. In Sec. 5.4.1, we compare the residuum–socket pressures predicted by our Muscleless Model with those reported in the literature for a similar model without a muscle. In Sec. 5.4.2, we present a sensitivity analysis that investigated how the pressure predicted by our Small-Muscle Model varied as three key parameters varied. Finally, in Sec. 5.4.3, we compare the pressures predicted by our Small-Muscle Model and Standard-Size-Muscle Model when the muscle

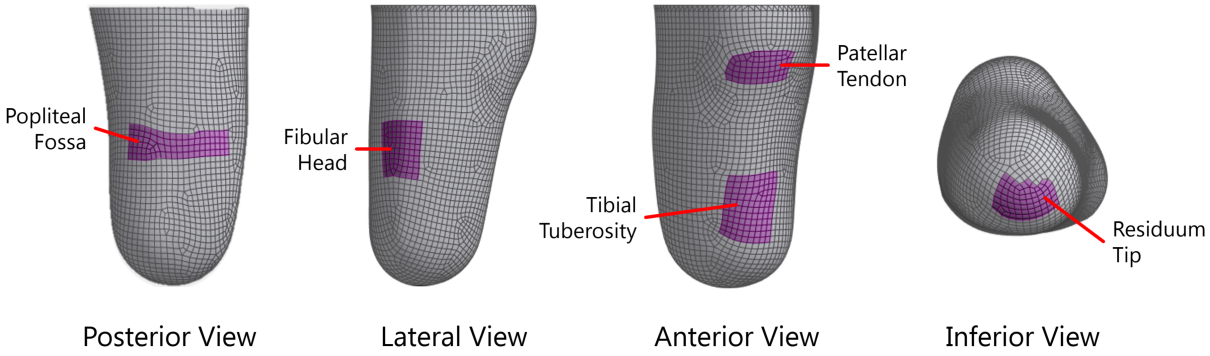

Figure 5.3: Regions in which residuum–socket pressures were considered. Left to right: posterior, lateral, anterior, and inferior views.

was passive ($\alpha = 0$) and fully active ($\alpha = 1$). Throughout, "pressure" refers to the 95th percentile of the pressure distribution among all the elements in the specified region of the residuum surface.

5.4.1 Benchmarking against an existing model

The residuum–socket pressures predicted by our Muscleless Model were compared to those reported by Steer et al. [29] from a finite element model of similar design, where the entire residuum was modelled as a single generic soft-tissue material (Table 5.5). At heel strike and midstance, our Muscleless Model predicted pressures that were within 10% of the corresponding values reported by Steer et al. in the residuum tip, fibular head, and tibial tuberosity regions. For the heel strike and midstance load cases, the greatest differences between the pressures predicted by our Muscleless Model and those reported by Steer et al. were in the patellar tendon region, with differences of 22% at heel strike and 36% at midstance. At toe off, our Muscleless Model predicted pressures that were between 22% and 33% of the corresponding values reported by Steer et al. in all four regions they considered; the greatest differences were observed in the fibular head and patellar tendon regions.

Table 5.5: Benchmark of our Muscleless Model. The 95th percentile of the pressure distributed on the residual limb surface was collected in the residuum tip, fibular head, tibial tuberosity, and patellar tendon regions at heel strike, midstance, and toe off.

		95th percentile pressure (kPa)		
Load case	Region	Steer et al. [29]	Muscleless Model	Difference (%)
Heel strike	Residuum tip	62.4	63.4	1.6
	Fibular head	70.7	70.5	−0.3
	Tibial tuberosity	66.3	60.2	−9.2
	Patellar tendon	22.8	27.9	22.4
Midstance	Residuum tip	69.1	73.7	6.7
	Fibular head	79.6	81.1	1.9
	Tibial tuberosity	71.0	67.3	−5.2
	Patellar tendon	33.1	44.9	35.6
Toe off	Residuum tip	66.7	81.5	22.2
	Fibular head	74.0	98.5	33.1
	Tibial tuberosity	40.0	51.1	27.8
	Patellar tendon	117.1	155.1	32.5

5.4.2 Sensitivity of predicted interfacial pressure to muscle parameters

The sensitivity of residuum–socket pressure predicted by the Small-Muscle Model was explored as λ_{opt} and λ_{act} varied, with all other parameters given in Table 5.1. When λ_{opt} was varied, the greatest increase in residuum–socket pressure was 2.8% in the patellar tendon region at heel strike (when $\lambda_{\text{opt}} = 1.1$) and the greatest decrease was 2.5% in both the tibial tuberosity and popliteal fossa regions (also when $\lambda_{\text{opt}} = 1.1$). When λ_{act} was varied, the greatest increase in residuum–socket pressure was 2.2% in the patellar tendon region at heel strike and the greatest decrease was 1.1% in the residuum tip region (both observed when $\lambda_{\text{act}} = 1.2$).

The sensitivity of residuum–socket pressures predicted by the Small-Muscle Model was also explored as muscle activation (α) varied. At heel strike, the pressure in all regions of interest varied modestly and monotonically as activation increased from 0 to 1 (Fig. 5.4(a)). At toe off, the pressure varied monotonically as activation increased from 0 to 1 in the residuum tip, tibial tuberosity, and popliteal fossa regions; however, in the

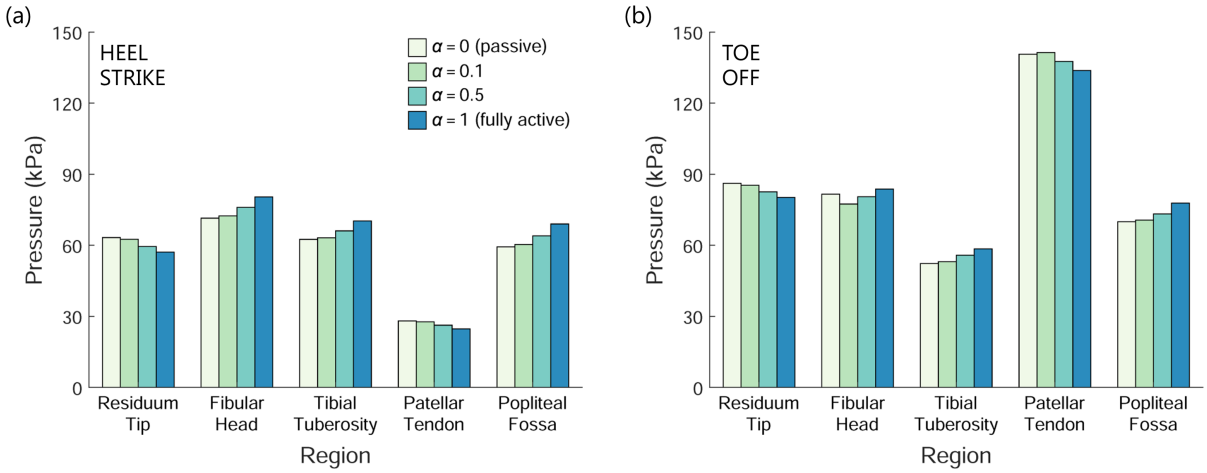

Figure 5.4: Residuum–socket pressures predicted by the Small-Muscle Model as activation (α) varied from 0 (passive) to 1 (fully active). The 95th percentile pressure was evaluated in the residuum tip, fibular head, tibial tuberosity, patellar tendon, and popliteal fossa regions at (a) heel strike and (b) toe off.

fibular head and patellar tendon regions, the variations in pressure were not monotonic (Fig. 5.4(b)). For example, the pressure in the fibular head region decreased by 5.1% when activation increased from 0 to 0.1, but the pressure then increased with further increases in activation.

5.4.3 Effect of muscle activation and size on residuum–socket pressure

Muscle activation affected the predicted residuum–socket pressure in all five regions of interest at both heel strike and toe off, regardless of the size of the gastrocnemius muscle. At heel strike, regardless of muscle size, the pressure was greatest in the residuum tip and patellar tendon regions when the muscle was passive rather than active; in the fibular head, tibial tuberosity, and popliteal fossa regions, the pressure was greatest when the muscle was fully active (Fig. 5.5(a)). The greatest effect of muscle activation was observed in the popliteal fossa region with the Standard-Size-Muscle Model, where the pressure was 42% greater when the muscle was fully active than when passive. Similar trends were observed at toe off (Fig. 5.5(b)): the greatest effect of muscle activation was again observed in the popliteal fossa region with the Standard-Size-Muscle Model, where the

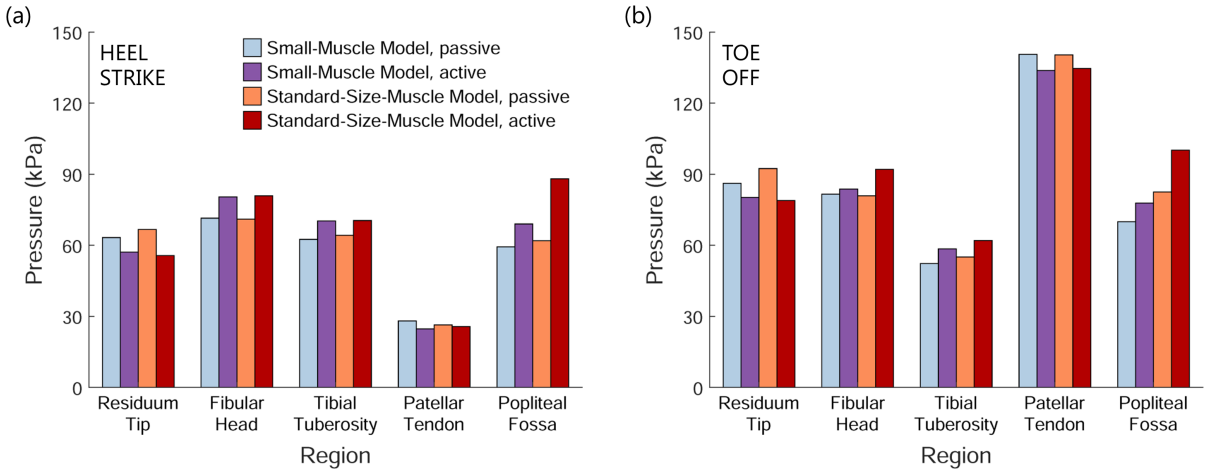

Figure 5.5: Residuum–socket pressures predicted by the Small-Muscle Model and Standard-Size-Muscle Model when the muscle was passive ($\alpha = 0$) and fully active ($\alpha = 1$). The 95th percentile pressure was evaluated in the residuum tip, fibular head, tibial tuberosity, patellar tendon, and popliteal fossa regions at (a) heel strike and (b) toe off.

pressure was 11% greater when the muscle was fully active than when passive. As an illustrative example, the pressure distribution across the residuum–socket interface is shown in Fig. 5.6 for the Small-Muscle Model at heel strike in the passive and fully active conditions.

The muscle size did not dramatically affect the predicted residuum–socket pressure at either heel strike or toe off in any region other than the popliteal fossa region. At heel strike, the pressure in this region was 28% greater with the Standard-Size-Muscle Model than with the Small-Muscle Model when the muscle was fully active; this difference was similar at toe off (29%). At toe off, there was also a noticeable pressure difference in the fibular head region: when the muscle was fully active, the pressure was 10% greater in the Standard-Size-Muscle Model than in the Small-Muscle Model.

5.5 Discussion

The aim of this study was to investigate how muscle contraction affects the pressure generated at the residuum–socket interface during gait using a transtibial prosthesis. We

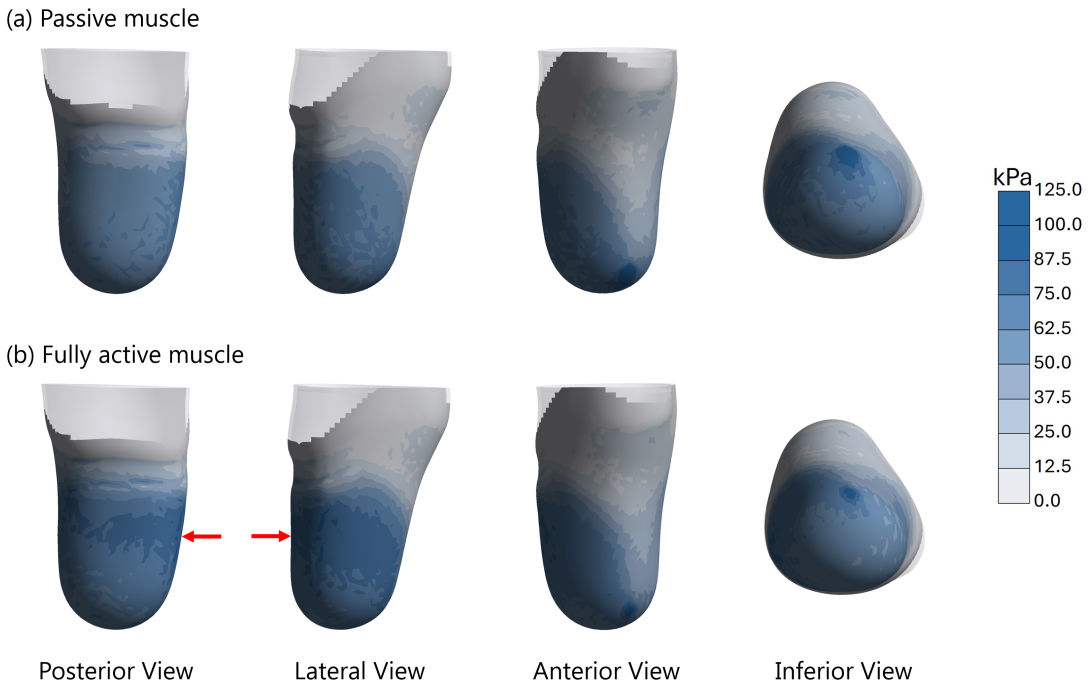

Figure 5.6: Colour map of the 95th percentile residuum–socket pressure at heel strike predicted by the Small-Muscle Model. (a) Passive muscle. (b) Fully active muscle. Left to right: posterior, lateral, anterior, and inferior views. In the posterior and lateral views, greater pressures (darker shades) were observed in the popliteal fossa and fibular head regions when the muscle was fully active (arrows).

generated the first finite element simulations of a residual limb following transtibial amputation that consider gastrocnemius muscle contraction. We explored the residuum–socket pressure in five critical regions using two muscle sizes in our model and with activation ranging from 0 (passive) to 1 (fully active), as the muscle geometry and activity are known to vary substantially among individuals during prosthesis-assisted gait [38]. At heel strike, using our Standard-Size-Muscle Model, the region that experienced the greatest pressure was the fibular head region when the muscle was passive and the popliteal fossa region when the muscle was fully active. Experimental studies of interfacial pressure from Goh et al. [48] and Dumbleton et al. [49] found that the region with greatest pressure during early stance (corresponding to our heel-strike condition) was the proximal posterior region (corresponding to our popliteal fossa region). Thus, our results suggest that considering the presence and activation of the muscle in finite element simulations is necessary to evaluate residuum–socket pressures in realistic conditions.

We compared the interfacial pressures predicted by our Muscleless Model to those reported by Steer et al. [29], whose finite element model considered the entire residuum as a single generic soft-tissue material. At heel strike and midstance, our Muscleless Model predicted residuum–socket pressures that were within 10% of the results of Steer et al. in the residuum tip, fibular head, and tibial tuberosity regions; in the patellar tendon region, our model predicted pressures that were 22% greater at heel strike and 36% greater at midstance. At toe off, the residuum–socket pressure was 22–33% greater in our Muscleless Model in all four regions that were investigated by Steer et al. There are several possible explanations for these differences. First, Steer et al. modelled the patellar and quadriceps tendons with different materials than the rest of the residual limb. Second, Steer et al. applied fixed constraints to only the proximal surfaces of the femur and quadriceps tendon, thereby permitting knee flexion during the simulation. In our model, the bones were represented by hollows with fixed constraints; thus, the knee angle was fixed in our simulations. Third, our Muscleless Model was defined using different geometry than that used by Steer et al., and geometry is known to affect the

residuum–socket pressure [11]. Despite the differences in the magnitudes of the pressures predicted by our Muscleless Model and the results of Steer et al. at toe off, the pressure predicted by our model was greater by approximately the same proportion in each region (between 22% and 33%; Table 5.5). Furthermore, the maximum pressure predicted by our Muscleless Model (155.1 kPa) was within the range of 100–250 kPa reported by existing finite element studies in the literature [11, 19, 28, 34, 50].

A sensitivity analysis was conducted to investigate the sensitivity of our predictions to changes in the muscle material model parameters. The residuum–socket pressures in all regions of interest were insensitive to variations in the parameters λ_{opt} (optimal fiber stretch) and λ_{act} (fiber stretch at the onset of activation), with pressure differences of less than 3% in all tested cases. This low sensitivity can be explained by noting the fiber stretch in the muscle belly region. In our baseline simulations, where $\lambda_{\text{opt}} = \lambda_{\text{act}} = 1$, the fiber stretch in the muscle belly region at heel strike was between 0.9 and 1.0. Therefore, the muscle fibers were operating in the plateau region of the active force–length curve [46], where active force does not vary substantially with small changes in fiber length. Consequently, variations in λ_{opt} and λ_{act} did not substantially affect either the stress generated by the muscle or the muscle's shape.

Muscle activation (α) affected the residuum–socket pressure in all five regions of interest using the Small-Muscle Model. At heel strike and toe off, the pressure in the fibular head, tibial tuberosity, and popliteal fossa regions was greater when the muscle was fully active, whereas the pressure in the residuum tip and patellar tendon regions was greater when the muscle was passive. The residuum–socket pressure in the residuum tip, tibial tuberosity, and patellar tendon regions was affected by the position and orientation of the socket. For example, Fig. 5.7 illustrates the location of the socket at heel strike when the muscle was passive and fully active. The socket in the passive-muscle simulation was positioned superiorly relative to the socket in the active-muscle simulation, which resulted in a greater pressure in the residuum tip region when the muscle was passive. Near the patellar tendon region, the socket in the passive-muscle simulation

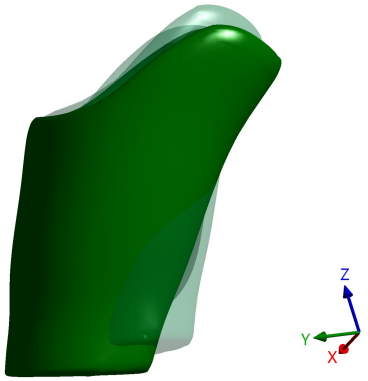

Figure 5.7: Lateral view of the socket at heel strike in the Small-Muscle Model with a passive muscle (transparent) and a fully active muscle (opaque). Deformation has been magnified 20 times for illustration purposes. Muscle activity altered the position of the socket which, in turn, affected the residuum–socket pressure in the residuum tip, tibial tuberosity, and patellar tendon regions.

was positioned posteriorly relative to the socket in the active-muscle simulation, which resulted in a greater pressure in the patellar tendon region when the muscle was passive. In contrast, the socket was positioned more posteriorly in the tibial tuberosity region when the muscle was active, which resulted in a greater pressure in this region with an active muscle.

Muscle size affected the residuum–socket pressure primarily in the popliteal fossa and fibular head regions. The greatest effect of muscle size was observed in the popliteal fossa region due to the close proximity of the muscle to the surface of the residuum in this region. In general, the presence of the muscle increased stiffness deep to the popliteal fossa region; consequently, the interfacial pressure was generally greater in this region with the Standard-Size-Muscle Model than with the Small-Muscle Model (Fig. 5.6). The greatest pressure in this region was observed with the larger muscle at full activation.

Several adjustments to modelling and simulation parameters were required to achieve convergence. Use of a muscle material model that exhibits force enhancement during active lengthening improves the numerical stability of finite element simulations because, unlike typical Hill-type muscle models, the active force–length curve has positive stiffness at all muscle lengths [33]. Notwithstanding the numerical stability of the muscle material

model, our initial residuum–socket model did not converge for the toe-off load case using either the Small-Muscle Model or the Standard-Size-Muscle Model. The elements of the residuum located anterior to the patella and inferior to the cut end of the fibula exhibited high distortion. Convergence was achieved by changing the order of the elements in the residuum from quadratic to linear (lower-order elements tend to be stiffer, resulting in less deformation) and then reducing the size of these elements to compensate for the reduction in the number of nodes per element (Table 5.2). Segmenting the soft tissue into muscle and non-muscle materials introduced more numerical instability, as was also observed by Cagle et al. [11]. Contact between the muscle and the remaining soft tissue was defined as "shared topology" (i.e., adjacent elements share nodes and edges), which implies bonded contact. Our simulations did not converge when other types of contact between these geometries were used.

Several simplifications were made to the model that may limit the applicability of our results. First, the bones were modelled as hollows rather than solid structures, neither tendons nor ligaments were modelled, and the structures within the residual limb were unable to slide over each other. One consequence of these simplifications is that the knee flexion angle was fixed during each simulation, which may affect the predicted residuum–socket pressure. Second, only quasi-static loading was simulated; analyses that use explicit simulation may capture the dynamic behaviour of each component more accurately [28]. Third, we compared the results from our Muscleless Model to those reported in the literature using a model that was similar but not identical [29]. Specifically, we did not obtain measurements of limb geometry, kinematics, muscle activation, ground reaction forces, or residuum–socket pressures from an individual and, instead, used existing data available in the literature. Experiments to collect these data from individuals would enable more rigorous calibration and validation of our model and simulations. Finally, we approximated the geometry of the gastrocnemius muscle following transtibial amputation. Using medical images obtained from an amputated limb would enable the creation of more realistic geometry and would enhance the fidelity of the model.

To be effective in clinical use, finite element models must provide sufficiently reliable and representative predictions. This study demonstrated that predictions of residuum–socket pressures can change dramatically when muscle geometry and contraction are considered in the model. Existing studies that model the residuum as a single homogeneous soft-tissue material, neglecting the presence and activity of the remaining muscle tissue, may predict different interfacial pressures and arrive at different conclusions with respect to the fit of the prosthesis. We recommend considering muscle geometry and contraction when designing personalized prosthesis sockets.

In summary, this study highlights the effect of considering muscle geometry and contraction on finite element simulations for predicting the pressure on the interface between the residuum and a transtibial prosthesis socket. To the best of our knowledge, this study presents the first finite element model of a limb with transtibial amputation that considers the geometry and contraction of the gastrocnemius muscle [51]. Our simulations suggest that interfacial pressures and socket fit are sensitive to muscle contraction, which is ignored by existing models. Finite element models have been used to propose new socket designs [17–19]; adoption of our model may help to improve these designs by providing more accurate predictions of interfacial pressures. Publicly available medical images, finite element models, and experimental measurements would accelerate the study of the biomechanics of prosthesis-assisted gait and the simulation-guided design of personalized prosthesis sockets. We have shared our models at https://simtk.org/projects/fea_limb_socket so that others can reproduce and extend our results.

Bibliography for Chapter 5

[1] F.E. Sunderland et al. OpenLimb: an open source transtibial residual limb model for simulation and design. *Prosthetics and Orthotics International*, 47, 165, 2023.

[2] M. Marino et al. Access to prosthetic devices in developing countries: pathways and challenges. In: *Proceedings 2015 IEEE Global Humanitarian Technology Conference*. Seattle, WA, USA, 2015, pp. 45–51.

[3] S. Chatterjee et al. Review: problems with use of trans-tibial prosthesis. *Journal of Medical Imaging and Health Informatics*, 6(2), 269–284, 2016.

[4] A. Esquenazi and S.K. Yoo. Lower limb amputations: epidemiology and assessment. *PMR Knowledge Now*, 3, 2016.

[5] W. Lovegreen et al. Lower limb amputation and gait. In: *Braddom's Physical Medicine and Rehabilitation*. Ed. by David X. Cifu. 6th. Philadelphia, PA, USA: Elsevier, 2021, pp. 174–208.

[6] K.A. Raichle et al. Prosthesis use in persons with lower- and upper-limb amputation. *Journal of Rehabilitation Research and Development*, 45(7), 961–972, 2008.

[7] A.C. Donaghy et al. Team approach to prosthetic prescription decision-making. *Current Physical Medicine and Rehabilitation Reports*, 8(4), 386–395, 2020.

[8] G.K. Klute, B.C. Glaister, and J.S. Berge. Prosthetic liners for lower limb amputees: a review of the literature. *Prosthetics and Orthotics International*, 34(2), 146–153, 2010.

[9] C.J.C. Lamoth et al. Variability and stability analysis of walking of transfemoral amputees. *Medical Engineering and Physics*, 32(9), 1009–1014, 2010.

[10] M.B. Silver-Thorn, J.W. Steege, and D.S. Childress. A review of prosthetic interface stress investigations. *Journal of Rehabilitation Research and Development*, 33(3), 253–266, 1996.

[11] J.C. Cagle et al. A finite element model to assess transtibial prosthetic sockets with elastomeric liners. *Medical and Biology Engineering Computing*, 56(7), 1227–1240, 2018.

[12] T. Arnstein and A. Buis. Methodology to investigate effect of prosthetic interface design on residual limb soft tissue deformation. *Canadian Prosthetics & Orthotics Journal*, 6(1), 7, 2023.

[13] L. Paternò et al. Sockets for limb prostheses: a review of existing technologies and open challenges. *IEEE Transactions on Biomedical Engineering*, 65(9), 1996–2010, 2018.

[14] L.E. Pezzin et al. Use and satisfaction with prosthetic limb devices and related services. *Archives of Physical Medicine and Rehabilitation*, 85(5), 723–729, 2004.

[15] R. Gailey et al. Review of secondary physical conditions associated with lower-limb amputation and long-term prosthesis use. *Journal of Rehabilitation Research and Development*, 45(1), 15–30, 2008.

[16] S. Portnoy et al. Internal mechanical conditions in the soft tissues of a residual limb of a trans-tibial amputee. *Journal of Biomechanics*, 41(9), 1897–1909, 2008.

[17] F.S. Frillici et al. The role of simulation tools to innovate the prosthesis socket design process. In: *Innovative Production Machines and Systems*. Ed. by D.T. Pham, E.E. Eldukhri, and A.J. Soroka. Cardiff University, UK: MEC, 2008, pp. 214–219.

[18] G. Colombo, G. Facoetti, and C. Rizzi. A digital patient for computer-aided prosthesis design. *Interface Focus*, 3(2), 20120082, 2013.

[19] K.M. Moerman et al. Automated and data-driven computational design of patient-specific biomechanical interfaces. *EngrXiv*, 2016.

[20] J.E. Sanders and S. Fatone. Residual limb volume change: systematic review of measurement and management. *Journal of Rehabilitation Research and Development*, 48(8), 949–986, 2011.

[21] S.G. Zachariah et al. Shape and volume change in the transtibial residuum over the short term: preliminary investigation of six subjects. *Journal of Rehabilitation Research and Development*, 41(5), 683–694, 2004.

[22] R.T. Youngblood et al. Effects of activity intensity, time, and intermittent doffing on daily limb fluid volume change in people with transtibial amputation. *Prosthetics and Orthotics International*, 43(1), 28–38, 2019.

[23] M. Lilja, S. Johansson, and T. Öberg. Relaxed versus activated stump muscles during casting for trans-tibial prostheses. *Prosthetics and Orthotics International*, 23(1), 13–20, 1999.

[24] D. Solav et al. A framework for measuring the time-varying shape and full-field deformation of residual limbs using 3-D digital image correlation. *IEEE Transactions on Biomedical Engineering*, 66(10), 2740–2752, 2019.

[25] Y. Yurttas et al. A novel technique for transtibial amputation in chronic occlusive arterial disease: modified Burgess procedure. *Prosthetics and Orthotics International*, 33(1), 25–32, 2009.

[26] C.C. Lyon et al. Skin disorders in amputees. *Journal of the American Academy of Dermatology*, 42(3), 501–507, 2000.

[27] C.C. Lin et al. Effects of liner stiffness for trans-tibial prosthesis: a finite element contact model. *Medical Engineering and Physics*, 26(1), 1–9, 2004.

[28] X. Jia et al. A quasi-dynamic nonlinear finite element model to investigate prosthetic interface stresses during walking for trans-tibial amputees. *Clinical Biomechanics*, 20(6), 630–635, 2005.

[29] J.W. Steer et al. Key considerations for finite element modelling of the residuum–prosthetic socket interface. *Prosthetics and Orthotics International*, 45(2), 138–146, 2021.

[30] V. Joumaa, T.R. Leonard, and W. Herzog. Residual force enhancement in myofibrils and sarcomeres. *Proceedings of the Royal Society B: Biological Sciences*, 275(1641), 1411–1419, 2008.

[31] W. Herzog et al. A new paradigm for muscle contraction. *Frontiers in Physiology*, 6, 174, 2015.

[32] S.H. Yeo et al. Numerical instability of Hill-type muscle models. *Journal of the Royal Society Interface*, 20(199), 20220430, 2023.

[33] M.L. Sampaio de Oliveira and T.K. Uchida. Phenomenological muscle constitutive model with actin–titin binding for simulating active stretching. *Journal of Biomedical Engineering*, 147(1), 011002, 2025.

[34] V. Plesec and G. Harih. Development of a generic numerical transtibial model for limb–prosthesis system evaluation. *Applied Sciences*, 13(4), 2339, 2023.

[35] T.E. Andreassen et al. Three dimensional lower extremity musculoskeletal geometry of the visible human female and male. *Scientific Data*, 10(1), 34, 2023.

[36] P.F. Stahel et al. Concepts of transtibial amputation: Burgess technique versus modified Brückner procedure. *Journal of Surgery*, 76(10), 942–946, 2006.

[37] L.A. Spyrou, M. Agoras, and K. Danas. A homogenization model of the Voigt type for skeletal muscle. *Journal of Theoretical Biology*, 414, 50–61, 2017.

[38] S. Huang and D.P. Ferris. Muscle activation patterns during walking from transtibial amputees recorded within the residual limb-prosthetic interface. *Journal of Neuroengineering and Rehabilitation*, 9(1), 55, 2012.

[39] M.E. Llewellyn et al. Minimally invasive high-speed imaging of sarcomere contractile dynamics in mice and humans. *Nature*, 454(7205), 784–788, 2008.

[40] S.A. Maas et al. FEBio: Finite Elements for Biomechanics. *Journal of Biomechanical Engineering*, 134(1), 011005, 2012.

[41] J.C. Cagle et al. Characterization of prosthetic liner products for people with transtibial amputation. *Journal of Prosthetics and Orthotics*, 30(4), 187–199, 2018.

[42] L. Rayward and J.P. Little. A subject-specific FEM to predict deep tissue mechanical stresses when supine: development of efficient contact interfaces using Shared Topology. *Journal of Biomechanics*, 137, 111085, 2022.

[43] S.R. Koehler, Y.Y. Dhaher, and A.H. Hansen. Cross-validation of a portable, six-degree-of-freedom load cell for use in lower-limb prosthetics research. *Journal of Biomechanics*, 47(6), 1542–1547, 2014.

[44] V. Carbone et al. TLEM 2.0 – a comprehensive musculoskeletal geometry dataset for subject-specific modeling of lower extremity. *Journal of Biomechanics*, 48(5), 734–741, 2015.

[45] M.S. Jamaludin et al. Analysis of pressure distribution in transfemoral prosthetic socket for prefabrication evaluation via the finite element method. *Bioengineering*, 6(4), 98, 2019.

[46] T.K. Uchida and S.L. Delp. Biomechanics of Movement: The Science of Sports, Robotics, and Rehabilitation. Cambridge, MA, USA: The MIT Press, 2021.

[47] M. Zhang and W.C.C. Lee. Quantifying the regional load-bearing ability of transtibial stumps. *Prosthetics and Orthotics International*, 30(1), 25–34, 2006.

[48] J.C.H. Goh, P.V.S. Lee, and S.Y. Chong. Stump/socket pressure profiles of the pressure cast prosthetic socket. *Clinical Biomechanics*, 18(3), 237–243, 2003.

[49] T. Dumbleton et al. Dynamic interface pressure distributions of two transtibial prosthetic socket concepts. *Journal of Rehabilitation Research and Development*, 46(3), 405–415, 2009.

[50] M.C. Faustini, R.R. Neptune, and R.H. Crawford. The quasi-static response of compliant prosthetic sockets for transtibial amputees using finite element methods. *Medical Engineering and Physics*, 28(2), 114–121, 2006.

[51] A.S. Dickinson, J.W. Steer, and P.R. Worsley. Finite element analysis of the amputated lower limb: a systematic review and recommendations. *Medical Engineering and Physics*, 43, 1–18, 2017.

6

Conclusions

Discomfort is a common challenge affecting users of transtibial prostheses and can lead to reductions in mobility and social engagement. Millions of people are currently living with amputation and there are approximately one million new amputation surgeries performed each year, or one every 30 seconds [1]. The continuing costs associated with care for individuals with lower-limb amputation are substantial, amounting to over $4 billion per year in the United States alone [2].

The residual limb undergoes rapid changes in shape due to the activation of the muscles within it [3, 4]. Changes in the shape of the residuum can alter the stresses on its surface, which can lead to discomfort and pain for prosthesis users. Therefore, the goal of this thesis was to understand how predictions of the pressure over the interface between the residual limb and prosthesis socket are affected by muscle contraction during gait. We developed a new muscle material model and generated finite element simulations of a residuum and prosthesis socket to address this research question.

6.1 Summary of literature gap

Several finite element models had been developed to predict the residuum–socket interfacial pressure [5–7] and to evaluate new socket designs [8–10]. This thesis presents a finite element model of a limb with transtibial amputation that considers the geometry and contraction of the gastrocnemius muscle. During the development of our finite element model, we identified and addressed the following three key limitations in the literature:

1. The difficulty of implementing even simple user-defined muscle material models has hindered the widespread adoption of state-of-the-art muscle models when using commercial software packages such as Ansys. The effort and expertise required to implement these models may exceed the resources that are available to a research team. Furthermore, when models are developed, they are rarely shared for others to use and extend.

2. In the literature, muscle material models have often been used in finite element simulations without first establishing that the material models are numerically stable [11]. Most muscle material models that are used in finite element simulations neglect force enhancement, which is a phenomenon that has been observed at all structural levels of muscle during active stretching. Furthermore, when force enhancement is ignored in simulations of active stretching, the force–length curve will have a region of negative stiffness [12], which can lead to numerically unstable simulations [11, 13].

3. Lilja et al. [3] emphasized the important question of whether prosthetists should consider the effect of muscle contraction when fitting a prosthesis socket. Lilja et al. also acknowledged that, as of the late 1990s, only a few experimental studies had explored how muscle contraction affects prosthesis fit. More recently, this statement was corroborated by Paternò et al. [14] and Solav et al. [4], who confirmed the need to consider muscle shape changes during socket design. To

the best of the authors' knowledge, all finite element simulations in the literature that study the interface between the residuum and transtibial prosthesis sockets ignore muscle contraction.

6.2 Summary of key findings

In our first study, "Muscle Constitutive Model with a Tangent Modulus Approximation: Ansys Implementation and Verification" (Chapter 3), we applied an approximation of the tangent modulus to simplify the derivation and implementation of an existing muscle material model [15]. This work was later used to facilitate the development of our own muscle constitutive model (Chapter 4). We validated the approximation using uniaxial simulations of three test models whose elements had different amounts of skewness. We calculated the stress and strain from the three test models and compared our results with the stress and strain obtained from analogous simulations in FEBio, which uses the same muscle model but with the exact tangent modulus.

Overall, good agreement was found between our Ansys and FEBio simulations, though some discrepancies were observed. The root-mean-square-percentage error (RMSPE) in the Von Mises stress along the centerline was 0.00%, 3.03%, and 6.75% for the three test models (from least to greatest element skewness); the RMSPE in the longitudinal stress was 0.00%, 2.52%, and 8.52%, and the RMSPE in the longitudinal strain was 0.0%, 1.71%, and 6.07%, respectively. The elements that presented the greatest error were those with greatest skewness, which caused sharp peaks in stress and nonsmooth longitudinal strains. However, as long as the mesh is carefully constructed so as to avoid skew elements, these errors in stress and strain can be mitigated. Based on the results above, and noting that the approximation method is independent of the specific material model that is used, we recommend applying this approach when developing new muscle material models to simplify the implementation and to help avoid programming errors.

In our second study, "Phenomenological Muscle Constitutive Model with Actin–Titin Binding for Simulating Active Stretching" (Chapter 4), we developed a muscle mate-

rial model that considers force enhancement, building on an existing model [15]. We demonstrated the numerical stability of our model using an eigenvalue analysis and by simulating a 3D filament model whose fibers were of different lengths. We then used a 3D muscle geometry to validate the effect of force enhancement on the development of stress and the distribution of fiber lengths.

During active stretching, our proposed muscle material model was more numerically stable than the model on which it was based [15]. Negative eigenvalues were observed in the spatial tangent modulus during an active stretch simulation when using the model of Blemker et al. Specifically, the eigenvalue associated with the fiber direction was negative over a large range of fiber stretch ($1.147 \leq \lambda \leq 1.574$), while analogous simulations using our muscle material model exhibited only positive eigenvalues. Simulations of a 3D filament model whose fibers had the same optimal length but slightly different lengths at the beginning of the simulation did not converge when using the model of Blemker et al. Using our proposed model, the simulation converged and all fibers stretched by approximately the same amount, as would be expected in a real muscle. In a series of simulations consisting of passive stretching followed by active stretching with the same total amount of stretch in each case, we observed greater stress following an active stretch compared to the stress during an isometric contraction (i.e., a purely passive stretch followed by activation of the muscle while held at a constant length). We also observed greater stress when a greater proportion of the total stretch was active rather than passive, which agrees with experimental observations of force enhancement [16]. Using our proposed model, we also observed sarcomere lengths that were in better agreement with experimental measurements than when using the model of Blemker et al., which does not consider force enhancement.

In our third study, "Pressure Between Residual Limb and Transtibial Prosthesis Socket Considering Muscle Contraction: A Finite Element Study" (Chapter 5), we developed finite element models to investigate how muscle contraction affects the pressure on the residuum surface. We used and modified the OpenLimb model [17] for our model

geometry. We augmented this model with a gastrocnemius muscle whose dynamic behaviour was governed by our new muscle constitutive model (Chapter 4). We simulated two load profiles representing the conditions during the heel-strike and toe-off events of gait. Three finite element models were created: a Muscleless Model, a Standard-Size-Muscle Model, and a Small-Muscle Model. We validated our modelling approach by comparing the residuum–socket pressure predicted by our Muscleless Model with that reported in the literature for a similarly muscleless model [7]. The Standard-Size-Muscle Model and Small-Muscle Model were then used to analyze the effect of muscle size and activation on the residuum–socket pressure.

The interfacial pressure predicted by our Muscleless Model compared well with the pressures reported by Steer et al. [7]. The differences we observed in interfacial pressure were believed to be due to different modelling choices and different geometry between the two models. Notwithstanding these differences, the maximum pressure predicted by our Muscleless Model was within the range of pressures reported by existing finite element studies in the literature [5, 6, 10, 18, 19]. Muscle activation affected the residuum–socket pressure in all five regions of interest (residuum tip, fibular head, tibial tuberosity, patellar tendon, and popliteal fossa). We observed that the position and orientation of the socket differed in the simulations with passive and active muscle. At heel strike and toe off, the pressure in the fibular head, tibial tuberosity, and popliteal fossa regions was greater when the muscle was fully active, whereas the pressure in the residuum tip and patellar tendon regions was greater when the muscle was passive. Muscle size affected the residuum–socket pressure primarily in the fibular head and popliteal fossa regions. From the results of our simulations, we concluded that existing studies that modelled the residuum as a single homogeneous soft-tissue material, neglecting the presence and activity of the muscle tissue within the residuum, may predict different interfacial pressures and may arrive at different conclusions with respect to the fit of the prosthesis. We recommend considering muscle geometry and contraction when designing personalized prosthesis sockets.

The studies presented in this thesis demonstrate advances in the field of biomechanics and computational modelling. By combining finite element methods with a novel muscle material model, this work provides insight into how muscle contraction affects the pressure across the residuum–socket interface. Incorporating the effects of muscle contraction into finite element simulations of transtibial prosthesis sockets could lead to improvements in prosthesis fit and comfort. We advanced constitutive muscle material modelling by proposing a muscle model that includes force enhancement and is appropriate for simulations of active stretching. The model also improved the numerical stability of the model on which it was based [15]. Furthermore, our work advanced the implementation of complex muscle material models in widely used commercial software like Ansys by simplifying the implementation. Collectively, these contributions have the potential to improve prosthesis design by enhancing the accuracy of numerical models.

6.3 Limitations of the studies

Our Ansys implementation of a muscle constitutive model using an approximation of the tangent modulus (Chapter 3) simplified the derivation and programming of the model, but three limitations should be addressed. First, the test cases we explored used relatively simple geometry. Using simple geometry enabled us to perform a thorough analysis and will allow others to easily reproduce our test cases. However, additional testing should be performed using more complex muscle geometries to evaluate the performance of the approximation over a broader range of use cases and to enable direct validation against experimental measurements. Second, we compared the stress and strain in our Ansys simulations to analogous simulations in FEBio. Some of the error in our test cases likely resulted from inherent differences between Ansys and FEBio rather than from the approximation of the tangent modulus itself—though we do not expect software-dependent differences to play a substantial role in all conditions since the relative error in stress and strain was 0.00% for the rectangular model. Third, we considered only hexahedral (trilinear) elements. Under the assumption of incompressibility, these elements can exhibit

volumetric locking if the ratio of the bulk modulus to the shear modulus is sufficiently small. In our test cases, the strain curve exhibited oscillatory behaviour, which may be due to volumetric locking. This issue can be mitigated by using reduced-integration elements (i.e., elements with fewer integration points) or higher-order elements.

Our constitutive muscle model incorporating actin–titin binding (Chapter 4) exhibits force enhancement during active stretching, a phenomenon that is ignored by most muscle models, but it has noteworthy limitations. Our study adopted assumptions about the actin–titin binding mechanisms, which remains a topic of debate in the field. The parameters that we used to construct the active and passive force–length curves were obtained using a single set of data from experiments on rabbit psoas muscles. More experimental data would provide greater confidence in the parameters used in the worm-like chain model for the PEVK segment of titin. The micromechanical behaviour of the fibers was directly applied at the macroscopic level, a simplification that is adopted by many muscle models but neglects the complex interactions and heterogeneities at the microscale, such as fiber–fiber interactions and the influence of extracellular matrix. Finally, we did not consider the force depression effect, which is an observed phenomenon where a muscle produces less force after an active shortening compared to the isometric force it can generate at the same length [20]. The worm-like chain model on which our model was based [21] is able to simulate this effect, but we did not confirm whether this capability extends to our model.

Our study on the pressure between the residual limb and transtibial prosthesis socket during muscle contraction (Chapter 5) advances the state-of-the-art in the finite element modelling of this interface, but this study also had limitations. First, our finite element model used a simplified version of the complex geometry of the gastrocnemius muscle. Second, several modelling decisions were made to simplify the model: the bones were modelled as hollows rather than solid structures, neither tendons nor ligaments were modelled, and the structures within the residual limb were unable to slide over each other. Third, we did not obtain measurements of limb geometry, kinematics, muscle activation,

ground reaction forces, or residuum–socket pressures from an individual; instead, we used existing data available in the literature. Experiments to collect these data from individuals would enable more rigorous calibration and validation of our model and simulations. Publicly available medical images, finite element models, and other related data would accelerate the study of the biomechanics of prosthesis-assisted gait. Lastly, our model does not account for long-term changes in tissue properties or socket wear over time.

6.4 Future work

Future work for the muscle constitutive model implemented in Ansys should aim at improving the accuracy of the tangent modulus approximation under highly nonlinear deformation scenarios. Expanding validation to include experimental data from various muscle geometries and activation conditions would enhance confidence in the approximation results. Additionally, testing the model under different boundary conditions and mesh element types would further explore its robustness. Furthermore, implementing the model in other commercial finite element platforms like Abaqus or open-source alternatives like FEBio would improve accessibility and facilitate broader adoption.

Future work for the actin–titin binding muscle constitutive model can focus on experimental validation to further evaluate the assumptions and parameters that were used, particularly using more data to validate the mechanical parameters of the PEVK segment. Due to the quasi-static nature of our simulations, time-dependent effects were ignored. The force–velocity relationship and other transient force response effects [22] could be considered in the future to enhance the model and broaden its applicability. Efforts to improve computational efficiency would enable the model's use in large-scale or real-time simulations without relying on polynomial interpolation of force–length curves generated a priori for different fiber lengths at the onset of activation. Additionally, performing homogenization between the extracellular matrix and fiber mediums would enable the model to better account for the interaction between the intramuscular tissues

[23]. Collaborative studies linking simulations of molecular dynamics with macroscopic finite element models may provide deeper insight into muscle behaviour. For example, the macroscopic effect of the variants (isoforms) of the titin segment could be observed from the model output [24]. Finally, the model could incorporate the force depression effect to better reflect muscle behaviour during simulations of active shortening [21].

Future work for the study on the pressure between the residual limb and transtibial prosthesis socket should focus on refining the finite element model to incorporate more individualized anatomical and material properties of the residuum. Use of patient-specific imaging data, such as MRI sequences or CT scans, could enhance the model's accuracy and clinical relevance. Further validation against experimental and clinical data, such as in vivo measurements of socket pressure during gait, would strengthen reliability in the model's predictions. Additionally, exploring a wider range of loading conditions and dynamic activities, such as climbing stairs, would provide a more comprehensive analysis of residuum–socket pressure during activities of daily living. Using patient-specific data, our finite element simulations could be used to generate customized sockets with new materials and designs—for example, sockets with different stiffnesses across different regions to minimize the peak interfacial pressure [10]. Prototypes of new sockets could then be produced and compared with the current models in terms of comfort and acceptance. Finally, during our simulations, we observed that the position and orientation of the socket changed when the muscle was activated. We hypothesized that prosthesis users may learn to modulate muscle activation throughout gait to adjust the residuum–socket fit. Further experimental work would be necessary to explore this question and its implications for socket design.

Bibliography for Chapter 6

[1] Access Prosthetics. 15 limb loss statistics that may surprise you. Accessed: 2021-05-17. Oct. 2017. URL: https://accessprosthetics.com/15-limb-loss-statistics-may-surprise.

[2] B. Imam et al. Incidence of lower limb amputation in Canada. *Canadian Journal of Public Health*, 108, 2017.

[3] M. Lilja, S. Johansson, and T. Öberg. Relaxed versus activated stump muscles during casting for trans-tibial prostheses. *Prosthetics and Orthotics International*, 23, 13–20, 1999.

[4] D. Solav et al. A framework for measuring the time-varying shape and full-field deformation of residual limbs using 3-D digital image correlation. *IEEE Transactions on Biomedical Engineering*, 66, 2740–2752, 2019.

[5] X. Jia et al. A quasi-dynamic nonlinear finite element model to investigate prosthetic interface stresses during walking for transtibial amputees. *Clinical Biomechanics*, 20, 630–635, 2005.

[6] J.C. Cagle et al. A finite element model to assess transtibial prosthetic sockets with elastomeric liners. *Medical and Biology Engineering Computing*, 56, 1227–1240, 2018.

[7] J.W. Steer et al. Key considerations for finite element modelling of the residuum–prosthetic socket interface. *Prosthetics and Orthotics International*, 45(2), 138–146, 2021.

[8] F.S. Frillici et al. The role of simulation tools to innovate the prosthesis socket design process. In: *Innovative Production Machines and Systems*. 2008, pp. 214–219.

[9] G. Colombo, G. Facoetti, and C. Rizzi. A digital patient for computer-aided prosthesis design. *Interface Focus*, 3, 2013.

[10] K.M. Moerman et al. Automated and data-driven computational design of patient-specific biomechanical interfaces. *EngrXiv*, 2016.

[11] S.H. Yeo et al. Numerical instability of Hill-type muscle models. *Journal of the Royal Society Interface*, 20(199), 20220430, 2023.

[12] T.L. Allinger, M. Epstein, and W. Herzog. Stability of muscle fibers on the descending limb of the force-length relation. A theoretical consideration. *Journal of Biomechanics*, 29, 627–633, 1996.

[13] M.L. Sampaio de Oliveira and T.K. Uchida. Phenomenological muscle constitutive model with actin–titin binding for simulating active stretching. *Journal of Biomechanical Engineering*, 147(1), 011002, 2025.

[14] L. Paternò et al. Sockets for limb prostheses: a review of existing technologies and open challenges. *IEEE Transactions on Biomedical Engineering*, 65, 1996–2010, 2018.

[15] S.S. Blemker, P.M. Pinsky, and S.L. Delp. A 3D model of muscle reveals the causes of nonuniform strains in the biceps brachii. *Journal of Biomechanics*, 38, 657–665, 2005.

[16] W. Herzog and T.R. Leonard. Force enhancement following stretching of skeletal muscle: a new mechanism. *Journal of Experimental Biology*, 205(9), 1275–1283, 2002.

[17] F.E. Sunderland et al. OpenLimb: An open-source transtibial residual limb model for simulation and design. In: *Proceedings of the International Society for Prosthetics and Orthotics World Congress*. Guadalajara, Mexico, 2023.

[18] M.C. Faustini et al. An experimental and theoretical framework for manufacturing prosthetic sockets for transtibial amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 14(3), 304–310, 2006.

[19] V. Plesec and G. Harih. Development of a generic numerical transtibial model for limb–prosthesis system evaluation. *Applied Sciences*, 13(4), 2339, 2023.

[20] N. Trecarten et al. Residual force depression in single sarcomeres is abolished by MgADP-induced activation. *Scientific Reports*, 5(1), 10555, 2015.

[21] G. Schappacher-Tilp et al. A novel three-filament model of force generation in eccentric contraction of skeletal muscles. *PLoS One*, 10, e0117634, 2015.

[22] V. Joumaa, T.R. Leonard, and W. Herzog. Residual force enhancement in myofibrils and sarcomeres. *Proceedings of the Royal Society B: Biological Sciences*, 275, 1411–1419, 2008.

[23] L.A. Spyrou, M. Agoras, and K. Danas. A homogenization model of the Voigt type for skeletal muscle. *Journal of Theoretical Biology*, 414, 50–61, 2017.

[24] L.G. Prado et al. Isoform diversity of giant proteins in relation to passive and active contractile properties of rabbit skeletal muscles. *The Journal of General Physiology*, 126, 461–480, 2005.

A

Software Implementations

A.1 Approximation of the elasticity tensor in Fortran

The Fortran code below implements the approximation of the elasticity tensor applied to the muscle material model proposed by Blemker et al. (2005). This code was developed by adapting and extending the USERMAT routine, a user-defined material subroutine found in the Ansys installation folder. The USERMAT routine allows for the integration of custom material models into Ansys simulations. The approximation simplifies the implementation of the material mode by avoiding the exact calculation of the fourth-order elasticity tensor. By leveraging the USERMAT framework, the code ensures compatibility with Ansys finite element analysis, facilitating simulations of muscle mechanics in biomechanical applications.

```
*deck,usermat      USERDISTRIB  parallel                                  gal
subroutine usermat(
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)
c************************************************************************
c     *** primary function ***
c
c           user defined material constitutive model
c
c      Attention:
c           User must define material constitutive law properly
c           according to the stress state such as 3D, plane strain
c           and axisymmetry, plane stress and 3D/1D beam.
c
c           a 3D material constitutive model can use for
c           plane strain and axisymmetry cases.
c
c           When using shell elements, a plane stress algorithm
c           must be use.
c
c                                            gal July, 1999
c
c       The following demonstrates a USERMAT subroutine for
c       a plasticity model, which is the same as TB, BISO,
c       for different stress states.
c       See "ANSYS␣user␣material␣subroutine␣USERMAT" for detailed
c       description of how to write a USERMAT routine.
c
c       This routine calls four routines,
c       usermat3d.F, usermatps.F usermatbm.F and usermat1d.F, w.r.t.
c       the corresponding stress states.
c       Each routine can be also a usermat routine for the specific
c       element.
c
c************************************************************************
c
c     input arguments
c     ===============
c      matId     (int,sc,i)           material #
c      elemId    (int,sc,i)           element #
c      kDomIntPt (int,sc,i)           "k"th domain integration point
c      kLayer    (int,sc,i)           "k"th layer
c      kSectPt   (int,sc,i)           "k"th Section point
c      ldstep    (int,sc,i)           load step number
c      isubst    (int,sc,i)           substep number
c      nDirect   (int,sc,in)          # of direct components
c      nShear    (int,sc,in)          # of shear components
c      ncomp     (int,sc,in)          nDirect + nShear
c      nstatev   (int,sc,l)           Number of state variables
c      nProp     (int,sc,l)           Number of material ocnstants
c
c      Temp      (dp,sc,in)           temperature at beginning of
c                                     time increment
c      dTemp     (dp,sc,in)           temperature increment
c      Time      (dp,sc,in)           time at beginning of increment (t)
c      dTime     (dp,sc,in)           current time increment (dt)
c
c      Strain   (dp,ar(ncomp),i)      Strain at beginning of time increment
c      dStrain  (dp,ar(ncomp),i)      Strain increment
c      prop     (dp,ar(nprop),i)      Material constants defined by TB,USER
```

```
c      coords   (dp,ar(3),i)          current coordinates
c      defGrad_t(dp,ar(3,3),i)       Deformation gradient at time t
c      defGrad  (dp,ar(3,3),i)       Deformation gradient at time t+dt
c
c     input output arguments
c     ======================
c      stress   (dp,ar(nTesn),io)      stress
c      ustatev   (dp,ar(nstatev),io)   user state variables
c      sedEl    (dp,sc,io)             elastic work
c      sedPl    (dp,sc,io)             plastic work
c      epseq    (dp,sc,io)             equivalent plastic strain
c      tsstif   (dp,ar(2),io)          transverse shear stiffness
c                                      tsstif(1) - Gxz
c                                      tsstif(2) - Gyz
c                                      tsstif(1) is also used to calculate
c                                      hourglass stiffness, this value must
c                                      must be defined when low order
c                                      element, such as 181, 182, 185
c                                      with uniform integration is used.
c      var?     (dp,sc,io)             not used, they are reserved
c                                      arguments for further development
c
c     output arguments
c     ================
c      keycut   (int,sc,io)              loading bisect/cut control
c                                        0 - no bisect/cut
c                                        1 - bisect/cut
c                                        (factor will be determined
c                                        by ANSYS solution control)
c      dsdePl   (dp,ar(ncomp,ncomp),io)  material jacobian matrix
c      epsZZ    (dp,sc,o)                strain epsZZ for plane stress,
c                                        define it when accounting
c                                        for thickness change
c                                        in shell and plane stress states
c
c*************************************************************************
c
c      ncomp   6   for 3D  (nshear=3)
c      ncomp   4   for plane strain or axisymmetric (nShear = 1)
c      ncomp   3   for plane stress (nShear = 1)
c      ncomp   3   for 3d beam      (nShear = 2)
c      ncomp   1   for 1D (nShear = 0)
c
c      stresss and strains, plastic strain vectors
c          11, 22, 33, 12, 23, 13    for 3D
c          11, 22, 33, 12            for plane strain or axisymmetry
c          11, 22, 12                for plane stress
c          11, 13, 12                for 3d beam
c          11                        for 1D
c
c      material jacobian matrix
c        3D
c           dsdePl    |  1111   1122   1133   1112   1123   1113 |
c           dsdePl    |  2211   2222   2233   2212   2223   2213 |
c           dsdePl    |  3311   3322   3333   3312   3323   3313 |
c           dsdePl    |  1211   1222   1233   1212   1223   1213 |
c           dsdePl    |  2311   2322   2333   2312   2323   2313 |
c           dsdePl    |  1311   1322   1333   1312   1323   1313 |
c        plane strain or axisymmetric (11, 22, 33, 12)
c           dsdePl    |  1111   1122   1133   1112 |
c           dsdePl    |  2211   2222   2233   2212 |
c           dsdePl    |  3311   3322   3333   3312 |
c           dsdePl    |  1211   1222   1233   1212 |
c        plane stress (11, 22, 12)
c           dsdePl    |  1111   1122   1112 |
c           dsdePl    |  2211   2222   2212 |
c           dsdePl    |  1211   1222   1212 |
c        3d beam (11, 13, 12)
```

```
c            dsdePl     |  1111   1113   1112 |
c            dsdePl     |  1311   1313   1312 |
c            dsdePl     |  1211   1213   1212 |
c         1d
c            dsdePl     |  1111 |
c
c***********************************************************************
#include "impcom.inc"
#include "ansysdef.inc"
c
INTEGER
&                  matId, elemId,
&                  kDomIntPt, kLayer, kSectPt,
&                  ldstep,isubst,keycut,
&                  nDirect,nShear,ncomp,nStatev,nProp
DOUBLE PRECISION
&                  Time,    dTime,   Temp,    dTemp,
&                  sedEl,   sedPl,   epseq,   epsZZ
DOUBLE PRECISION
&                  stress  (ncomp  ), ustatev (nStatev),
&                  dsdePl  (ncomp,ncomp),
&                  Strain  (ncomp  ), dStrain (ncomp  ),
&                  epsPl   (ncomp  ), prop    (nProp  ),
&                  coords  (3),
&                  defGrad (3,3),     defGrad_t(3,3),
&                  tsstif  (2)
c
EXTERNAL          usermat3d, usermatps, usermatbm, usermat1d
!EXTERNAL          myuserfunc
integer myvar

DOUBLE PRECISION var0, var1, var2, var3, var4, var5,
&                  var6, var7, var8

integer iott,wrinqr
external wrinqr
c
c***********************************************************************
c
iott = wrinqr(WR_OUTPUT)
!write(iott,*) ' ************************************************'
!write(iott,*) ' * #DEBUG# UserMatLib.dll USERMAT               *'
!write(iott,*) ' ************************************************'
IF(ncomp .GE. 4) THEN
c ***    3d, plane strain and axisymmetric example
call usermat3d (
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)

ELSE IF(nDirect.eq. 2 .and. ncomp .EQ. 3) THEN
c ***    plane stress example
call usermatps (
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
```

```
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)

ELSE IF(ncomp .EQ. 3) THEN
c ***    3d beam example
call usermatbm (
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)

ELSE IF(ncomp .EQ. 1) THEN
c ***    1d beam example
call usermat1d (
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)

END IF
return
end



*deck,usermat3d    USERDISTRIB  parallel                                  gal
subroutine usermat3d(
&                    matId, elemId,kDomIntPt, kLayer, kSectPt,
&                    ldstep,isubst,keycut,
&                    nDirect,nShear,ncomp,nStatev,nProp,
&                    Time,dTime,Temp,dTemp,
&                    stress,ustatev,dsdePl,sedEl,sedPl,epseq,
&                    Strain,dStrain, epsPl, prop, coords,
&                    var0, defGrad_t, defGrad,
&                    tsstif, epsZZ,
&                    var1, var2, var3, var4, var5,
&                    var6, var7, var8)
c**********************************************************************
c
c     input arguments
c     ===============
c      matId     (int,sc,i)           material #
c      elemId    (int,sc,i)           element #
c      kDomIntPt (int,sc,i)           "k"th domain integration point
c      kLayer    (int,sc,i)           "k"th layer
c      kSectPt   (int,sc,i)           "k"th Section point
c      ldstep    (int,sc,i)           load step number
c      isubst    (int,sc,i)           substep number
c      nDirect   (int,sc,in)          # of direct components
c      nShear    (int,sc,in)          # of shear components
c      ncomp     (int,sc,in)          nDirect + nShear
c      nstatev   (int,sc,1)           Number of state variables
c      nProp     (int,sc,1)           Number of material ocnstants
c
c      Temp      (dp,sc,in)           temperature at beginning of
c                                     time increment
```

```
c      dTemp      (dp,sc,in)           temperature increment
c      Time       (dp,sc,in)           time at beginning of increment (t)
c      dTime      (dp,sc,in)           current time increment (dt)
c
c      Strain   (dp,ar(ncomp),i)       Strain at beginning of time increment
c      dStrain  (dp,ar(ncomp),i)       Strain increment
c      prop     (dp,ar(nprop),i)       Material constants defined by TB,USER
c      coords   (dp,ar(3),i)           current coordinates
c      defGrad_t(dp,ar(3,3),i)         Deformation gradient at time t
c      defGrad  (dp,ar(3,3),i)         Deformation gradient at time t+dt
c
c     input output arguments
c     ======================
c      stress   (dp,ar(nTesn),io)        stress
c      ustatev   (dp,ar(nstatev),io)     user state variable
c          ustatev(1)                      - equivalent plastic strain
c          ustatev(2) - statev(1+ncomp)    - plastic strain vector
c          ustatev(nStatev)                - von-Mises stress
c      sedEl     (dp,sc,io)              elastic work
c      sedPl     (dp,sc,io)              plastic work
c      epseq     (dp,sc,io)              equivalent plastic strain
c      tsstif    (dp,ar(2),io)           transverse shear stiffness
c                                        tsstif(1) - Gxz
c                                        tsstif(2) - Gyz
c                                        tsstif(1) is also used to calculate
c                                        hourglass stiffness, this value must
c                                        be defined when low order element,
c                                        such as 181, 182, 185 with uniform
c                                        integration is used.
c      var?      (dp,sc,io)            not used, they are reserved
c                                        arguments for further development
c
c     output arguments
c     ================
c      keycut    (int,sc,io)             loading bisect/cut control
c                                        0 - no bisect/cut
c                                        1 - bisect/cut
c                                        (factor will be determined
c                                        by ANSYS solution control)
c      dsdePl    (dp,ar(ncomp,ncomp),io) material jacobian matrix
c      epsZZ     (dp,sc,o)               strain epsZZ for plane stress,
c                                        define it when accounting
c                                        for thickness change
c                                        in shell and plane stress states
c
c*******************************************************************
c
c      ncomp    6   for 3D  (nshear=3)
c      ncomp    4   for plane strain or axisymmetric (nShear = 1)
c      ncomp    3   for plane stress (nShear = 1)
c      ncomp    3   for 3d beam      (nShear = 2)
c      ncomp    1   for 1D (nShear = 0)
c
c      stresss and strains, plastic strain vectors
c          11, 22, 33, 12, 23, 13     for 3D
c          11, 22, 33, 12             for plane strain or axisymmetry
c          11, 22, 12                 for plane stress
c          11, 13, 12                 for 3d beam
c          11                         for 1D
c
c      material jacobian matrix
c        3D
c            dsdePl     |  1111   1122   1133   1112   1123   1113 |
c            dsdePl     |  2211   2222   2233   2212   2223   2213 |
c            dsdePl     |  3311   3322   3333   3312   3323   3313 |
c            dsdePl     |  1211   1222   1233   1212   1223   1213 |
c            dsdePl     |  2311   2322   2333   2312   2323   2313 |
c            dsdePl     |  1311   1322   1333   1312   1323   1313 |
```

```
c         plane strain or axisymmetric (11, 22, 33, 12)
c            dsdePl     |  1111   1122   1133   1112 |
c            dsdePl     |  2211   2222   2233   2212 |
c            dsdePl     |  3311   3322   3333   3312 |
c            dsdePl     |  1211   1222   1233   1212 |
c         plane stress (11, 22, 12)
c            dsdePl     |  1111   1122   1112 |
c            dsdePl     |  2211   2222   2212 |
c            dsdePl     |  1211   1222   1212 |
c         3d beam (11, 13, 12)
c            dsdePl     |  1111   1113   1112 |
c            dsdePl     |  1311   1313   1312 |
c            dsdePl     |  1211   1213   1212 |
c         1d
c            dsdePl     |  1111 |
c
c***********************************************************************
#include "impcom.inc"
#include "ansysdef.inc"
c
INTEGER
&                  matId, elemId,
&                  kDomIntPt, kLayer, kSectPt,
&                  ldstep,isubst,keycut,
&                  nDirect,nShear,ncomp,nStatev,nProp
DOUBLE PRECISION
&                  Time,    dTime,   Temp,    dTemp,
&                  sedEl,   sedPl,   epseq,   epsZZ
DOUBLE PRECISION
&                  stress  (ncomp  ), ustatev (nStatev),
&                  dsdePl  (ncomp,ncomp),
&                  Strain  (ncomp  ), dStrain (ncomp  ),
&                  epsPl   (ncomp  ), prop    (nProp  ),
&                  coords  (3),
&                  defGrad (3,3),     defGrad_t(3,3),
&                  tsstif  (2)
c
c***************** User defined part **********************************
c
c --- functions
c
DOUBLE PRECISION Doubledot_function,det,delta_function,Trace
c
c --- local variables
c
c Scalars
DOUBLE PRECISION JJ,Jsa,p,ptilda,TV2,fiber_x,fiber_y,fiber_z,
& SMALL,theta,k_ind,lambda_star,lambda_oft,sigma_max,ratio_lambda,
& lambda,alpha,sigma_active,sigma_passive,sigma_muscle_module,b2,
& energy_active, energy_passive,is_error

c 2nd order tensors
DOUBLE PRECISION Identity (3,3), FT (3,3), Fbar (3,3),
&                  FbarT(3,3), C(3,3), Cinv(3,3),Cbar(3,3),
&                  SigmaVol(3,3), SBar(3,3),T1(3,3),Siso(3,3),
&                  SigmaIsoVoight(3,3),SigmaIso(3,3),Fsa(3,3),
&                  FsaInv(3,3),F2(3,3),C2(3,3),F2t(3,3),C2inv(3,3),
&                  FsaBarInvT(3,3),C2Bar(3,3),SBar1(3,3),
&                  FsaBar(3,3),FsaBarInv(3,3), Inv(6,1),Siso1(3,3),
&                  SigmaIso1(3,3),SigmaIsoVoight1(3,3),SBar2(3,3),
&                  Siso2(3,3),SigmaIso2(3,3),SigmaIsoVoight2(3,3),
&                  F_A(3,1),TV1(1,3),g(2,1),R(3,3),U(3,3),
&                  Q(6,6), stress_corotated(6),stress_spatial(6),
&                  aux_matrix_jaumann(6,6),TTT1(6,6),Bbar(3,3),
&                  dsdePl_jaumann(6,6),dsdePl_test (6,6),
&                  dsdePl_spatial(6,6),T2(3,3),Invb(2,1),
&                  sigma_matrix(3,3),stress_matrix_co(3,3),
&                  stress_test(6),a(2,1),F_A_T(1,3),dF(3,3),
```

```
&                  Rt(3,3),Ut(3,3),p_ind(4),F_A_current(3,1),
&                  dsdePl_2(6,6),SigmaIso_cross(3,3),
&                       SigmaIso_along(3,3),eignVA(6),eignVC(6,6),
&                  dsdePl_destroyed(6,6)

c 4th order tensors
DOUBLE PRECISION Cref(3,3,3,3),Cspac(3,3,3,3),Cref1(3,3,3,3),
&                  Cspac1(3,3,3,3),Cref2(3,3,3,3),Cspac2(3,3,3,3),
&                  CC(3,3,3,3),CCC(3,3,3,3),Cvol(3,3,3,3),
&                  TT1(3,3,3,3),TT2(3,3,3,3),TT3(3,3,3,3),
&                  aux_tensor(3,3,3,3),CC_approx_co(3,3,3,3),
&                       dsdePl_jaumann_tensor(3,3,3,3),
&                  dsdePl_jaumann_tensor_co(3,3,3,3),
&                  CC_approx(3,3,3,3)
c
c
DOUBLE PRECISION   var0, var1, var2, var3, var4, var5,
&                                     var6, var7, var8,pass1,pass2
INTEGER                        i,j,k,l,m,n,IO,ic,jc,kc,lc
c
c**********************************************************************
c
integer iott,wrinqr
external wrinqr
c
c**********************************************************************
iott = wrinqr(WR_OUTPUT)
!write(iott,*) ' ************************************************'
!write(iott,*) ' * #DEBUG# UserMatLib.dll USERMAT3D             *'
!write(iott,*) ' ************************************************'
SMALL=1.0d-8

c Create identity matrix
do i=1,3
do j=1,3
Identity(i,j)=delta_function(i,j)
end do
end do

c Input material parameters
g(1,1)=prop(1)

g(2,1)=prop(2)

k_ind=prop(3)

fiber_x=prop(4)

fiber_y=prop(5)

fiber_z=prop(6)

lambda_star=prop(7)

lambda_oft=prop(8)

sigma_max=prop(9)

p_ind(1)=prop(10)

p_ind(2)=prop(11)

alpha=prop(12)


c Compute the other passive parameters
ratio_lambda=lambda_star/lambda_oft
p_ind(3)=p_ind(1)*p_ind(2)*exp(p_ind(2)*(ratio_lambda-1.0d0))
```

```
p_ind(4)=p_ind(1)*((1.0d0-p_ind(2)*ratio_lambda)*
&             exp(p_ind(2)*(ratio_lambda-1.0d0))-1.0d0)



c Create fiber vector
F_A(1,1)=(fiber_x)
F_A(2,1)=(fiber_y)
F_A(3,1)=(fiber_z)


c Calculate the matrix C,F and invariants
call c_matrix_and_inv(C,Cinv,FT,FBar,
& FbarT,Cbar,defGrad,F_A,JJ,BBar,Inv,lambda,F_A_current)


c Determine the Cauchy stress and elastic tensor
call GetStress(IO,Inv,JJ,C,Cinv,defGrad,FT,Cbar,k_ind,p,ptilda,
& SigmaIsoVoight,SigmaIso,SigmaVol,Siso,SBar,g,F_A,lambda,
& Identity,ncomp,stress_spatial,sigma_matrix,p_ind,sigma_max,
& lambda_star,lambda_oft,sigma_passive,sigma_active,
& sigma_muscle_module,alpha,F_A_current,BBar,SigmaIso_cross,
& SigmaIso_along)

ustatev(1)=lambda

c user state variables
b2=MAX((Inv(1,1)*Inv(4,1)-Inv(5,1))/(2.0d0*Inv(4,1)**0.5d0)
& ,1.0d0)

call ACTIVE_ENERGY(energy_active,lambda_oft,lambda_star,lambda,
& sigma_max,alpha)

call PASSIVE_ENERGY(p_ind,lambda_oft,lambda_star,lambda,
& sigma_max,energy_passive)

ustatev(1)=lambda
ustatev(2)=sigma_passive
ustatev(3)=SigmaIso(1,1)
ustatev(4)=SigmaVol(1,1)
ustatev(5)=SigmaIso_along(1,1)
ustatev(6)=SigmaIso_cross(1,1)
ustatev(7)=sigma_active
ustatev(8)=JJ


c Change tangent stiffness to Jaumann tangent stiffness
call extra_stiffness_tensor(aux_tensor,sigma_matrix)


c Change stress from spatial to corotated frame
call polarRU(defGrad,R,U)
call polarRU(defGrad_t,Rt,Ut)
call trans_matrx_6(Q,transpose(R))
call MULTENS_6x1_function(stress,Q,stress_spatial,1.0d0)




c Approximate Stifness routine
call GetStiffness_approx(CC_approx,defGrad,JJ,g, F_A,k_ind,
&  sigma_matrix,Identity,SMALL,R,dsdePl,p_ind,sigma_max,
&  lambda_star,lambda_oft,alpha)

c Update the Energy
theta=(Inv(1,1)*Inv(4,1)-Inv(5,1))/(2.0d0*Inv(4,1)**0.5d0)
sedEl =k_ind/2.0d0*log(JJ)**2.0d0+g(1,1)*(Inv(5,1)/Inv(4,1)**2.0d0
```

```
& -1.0d0)+g(2,1)*(log(theta))**2.0d0
sedPl=0.0d0

c transverse shear stiffness
tsstif(:) = dsdePl(6,6)



c That's all folks
return
end
!         FROM HERE ALL THE ROUTINES SUPLEMENT THE MAIN CALCULATION ABOVE

c**********************************************************************
SUBROUTINE MULTENS_function(TensRes,Tens1,Tens2,fact)

DOUBLE PRECISION TensRes(3,3),Tens1(3,3),Tens2(3,3),fact

INTEGER i,j,k

do i=1,3
do j=1,3
TensRes(i,j)=0.0d0
do k=1,3
TensRes(i,j)=TensRes(i,j)+Tens1(i,k)*Tens2(k,j)
end do
TensRes(i,j)=TensRes(i,j)*fact
end do
end do
END

c**********************************************************************
SUBROUTINE square_product(TensRes,Tens1,Tens2,fact)

DOUBLE PRECISION TensRes(3,3,3,3),Tens1(3,3),Tens2(3,3),fact

INTEGER i,j,k

do i=1,3
do j=1,3
do k=1,3
do l=1,3
TensRes(i,j,k,l)=Tens1(i,k)*Tens2(j,l)*fact
end do
end do
end do
end do
END

c**********************************************************************
SUBROUTINE MULTENS_6x1_function(TensRes,Tens1,Tens2,fact)

DOUBLE PRECISION TensRes(6),Tens1(6,6),Tens2(6),fact

INTEGER i,j,k

do i=1,6
TensRes(i)=0.0d0
do k=1,6
TensRes(i)=TensRes(i)+Tens1(i,k)*Tens2(k)
end do
TensRes(i)=TensRes(i)*fact
end do
END

c**********************************************************************
SUBROUTINE MULTENS_6x6_function(TensRes,Tens1,Tens2,fact)
```

```
DOUBLE␣PRECISION␣TensRes(6,6),Tens1(6,6),Tens2(6,6),fact

INTEGER␣i,j,k

do␣i=1,6
do␣j=1,6
TensRes(i,j)=0.0d0
do␣k=1,6
TensRes(i,j)=TensRes(i,j)+Tens1(i,k)*Tens2(k,j)
end␣do
TensRes(i,j)=TensRes(i,j)*fact
end␣do
end␣do
END

c**********************************************************************
SUBROUTINE␣T13dotT33(TensRes,Tvector,Tens,fact)

DOUBLE␣PRECISION␣TensRes(1,3),Tvector(1,3),Tens(3,3),fact

INTEGER␣i,j,k

do␣i=1,3
TensRes(1,i)=0.0d0
do␣j=1,3␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
TensRes(1,i)=TensRes(1,i)+Tvector(1,j)*Tens(j,i)
end␣do
TensRes(1,i)=TensRes(1,i)*fact
end␣do
END

c**********************************************************************
SUBROUTINE␣T33dotT31(TensRes,Tvector,Tens,fact)

DOUBLE␣PRECISION␣TensRes(3,1),Tvector(3,1),Tens(3,3),fact

INTEGER␣i,j,k

do␣i=1,3
TensRes(i,1)=0.0d0
do␣j=1,3␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
TensRes(i,1)=TensRes(i,1)+Tens(i,j)*Tvector(j,1)
end␣do
TensRes(i,1)=TensRes(i,1)*fact
end␣do
END

c**********************************************************************
SUBROUTINE␣T13dotT31(TensRes,Tens,Tvector,fact)

DOUBLE␣PRECISION␣TensRes,fact

DOUBLE␣PRECISION␣Tvector(3,1),Tens(1,3)


TensRes=(Tvector(1,1)*Tens(1,1)+Tvector(2,1)*Tens(1,2)
&+Tvector(3,1)*Tens(1,3))*fact
END

c**********************************************************************
SUBROUTINE␣TensorProduct_Vector(C,A,B)

DOUBLE␣PRECISION␣C(3,3),␣B(3,1),␣A(3,1)

do␣i=1,3
do␣j=1,3
C(i,j)=A(i,1)*B(j,1)
end␣do
```

```
end do

END

c*********************************************************************
SUBROUTINE WheelProduct(TensRes,Tens1,fact)

DOUBLE PRECISION TensRes(3,3,3,3),Tens1(3,3),fact

INTEGER i,j,k,l

do i=1,3
do j=1,3
do k=1,3
do l=1,3
TensRes(i,j,k,l)=fact/2.0d0*(Tens1(i,k)*Tens1(j,l)+
& Tens1(i,l)*Tens1(j,k))
end do
end do
end do
end do
END

c*********************************************************************
SUBROUTINE TensorProduct_function(TensRes,Tens1,Tens2,fact)

DOUBLE PRECISION TensRes(3,3,3,3),Tens1(3,3),Tens2(3,3),fact

INTEGER i,j,k,l

do i=1,3
do j=1,3
do k=1,3
do l=1,3
TensRes(i,j,k,l)=fact*Tens1(i,j)*Tens2(k,l)
end do
end do
end do
end do
END

c*********************************************************************
SUBROUTINE ScalTensor_function_debug(TensRes,Tensr,fact)

DOUBLE PRECISION TensRes(3,3),Tensr(3,3),fact

INTEGER i,j

call matrix_print(Tensr,3,3,'Tensr')

do i=1,3
do j=1,3
TensRes(i,j)=Tensr(i,j)*fact
end do
end do

print *,fact
call matrix_print(TensRes,3,3,'TensRes')
END

c*********************************************************************
SUBROUTINE ScalTensor_function(TensRes,Tensr,fact)

DOUBLE PRECISION TensRes(3,3),Tensr(3,3),fact

INTEGER i,j

do i=1,3
```

```
do␣j=1,3␣␣␣␣␣␣␣␣
TensRes(i,j)=Tensr(i,j)*fact
end␣do
end␣do

END

c**********************************************************************
SUBROUTINE␣ScalTens4_function(TensRes,Tensr,fact)

DOUBLE␣PRECISION␣TensRes(3,3,3,3),Tensr(3,3,3,3),fact

INTEGER␣i,j,k,l

do␣i=1,3
do␣j=1,3
do␣k=1,3
do␣l=1,3
TensRes(i,j,k,l)=Tensr(i,j,k,l)*fact
end␣do
end␣do
end␣do
end␣do
END

c**********************************************************************
Function␣DoubleDot_function(Tens1,Tens2)

DOUBLE␣PRECISION␣Tens1(3,3),Tens2(3,3),DoubleDot_function

INTEGER␣i,j

DoubleDot_function=0.0d0

do␣i=1,3
do␣j=1,3
DoubleDot_function=DoubleDot_function+Tens1(i,j)*Tens2(i,j)
end␣do
end␣do
return
END

c**********************************************************************
SUBROUTINE␣DoubleDot4_product(C,A,B,f)

DOUBLE␣PRECISION␣A(3,3,3,3),B(3,3,3,3),C(3,3,3,3)

DOUBLE␣PRECISION␣f

INTEGER␣i,j,k,l,m,n

do␣i=1,3
do␣j=1,3
do␣k=1,3
do␣l=1,3
C(i,j,k,l)=0.0d0
do␣m=1,3
do␣n=1,3
C(i,j,k,l)=C(i,j,k,l)+f*A(i,j,m,n)*B(m,n,k,l)
end␣do
end␣do
end␣do
end␣do
end␣do
end␣do

END
```

```
c**********************************************************************
SUBROUTINE␣T4ddT2_function(Tres,T4,T2,fact)

DOUBLE␣PRECISION␣T4(3,3,3,3),T2(3,3),Tres(3,3)

DOUBLE␣PRECISION␣fact

INTEGER␣i,j,k,l

do␣i=1,3
do␣j=1,3
Tres(i,j)=0.0d0
do␣k=1,3
do␣l=1,3
Tres(i,j)=Tres(i,j)+fact*T4(i,j,k,l)*T2(k,l)
end␣do
end␣do
end␣do
end␣do
END

c**********************************************************************
SUBROUTINE␣T2ddT4_function(Tres,T2,T4,fact)

DOUBLE␣PRECISION␣T4(3,3,3,3),T2(3,3),Tres(3,3)

DOUBLE␣PRECISION␣fact

INTEGER␣i,j,k,l
do␣i=1,3
do␣j=1,3
Tres(i,j)=0.0d0
do␣k=1,3
do␣l=1,3
Tres(i,j)=Tres(i,j)+fact*T4(k,l,i,j)*T2(k,l)
end␣do
end␣do
end␣do
end␣do
END

c**********************************************************************
Function␣Trace(Tensor)

DOUBLE␣PRECISION␣Tensor(3,3),Trace

INTEGER␣i

Trace=0.0d0
do␣i=1,3
Trace=Trace+Tensor(i,i)
end␣do
return
END

c**********************************************************************
SUBROUTINE␣Transpose_function(TensRes,Tensor)

DOUBLE␣PRECISION␣TensRes(3,3),Tensor(3,3)

INTEGER␣i,j
do␣i=1,3
do␣j=1,3
TensRes(i,j)=Tensor(j,i)
end␣do
end␣do
END
```

```
c***********************************************************************
SUBROUTINE␣Transpose_vector_function(VecRes,Vector)

DOUBLE␣PRECISION␣TensRes(3,1),Tensor(1,3)

INTEGER␣i
do␣i=1,3
TensRes(i,1)=Tensor(1,i)
end␣do
END

c***********************************************************************
SUBROUTINE␣Transpose4_function(TensRes,Tensor)

DOUBLE␣PRECISION␣TensRes(3,3,3,3),Tensor(3,3,3,3)

INTEGER␣i,j,k,l
do␣i=1,3
do␣j=1,3
do␣k=1,3
do␣l=1,3
TensRes(i,j,k,l)=Tensor(k,l,i,j)
end␣do
end␣do␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
end␣do
end␣do
END

c***********************************************************************
SUBROUTINE␣DEV_new_name(TensRes,Tensor)

DOUBLE␣PRECISION␣TensRes(3,3),Tensor(3,3),p

p=Tensor(1,1)+Tensor(2,2)+Tensor(3,3)
TensRes(1,1)=Tensor(1,1)-p/3.0d0
TensRes(3,3)=Tensor(3,3)-p/3.0d0
TensRes(2,2)=Tensor(2,2)-p/3.0d0
TensRes(1,2)=Tensor(1,2)
TensRes(2,1)=Tensor(2,1)
TensRes(3,1)=Tensor(3,1)
TensRes(1,3)=Tensor(1,3)
TensRes(2,3)=Tensor(2,3)
TensRes(3,2)=Tensor(3,2)
END

c***********************************************************************
SUBROUTINE␣EquateTens(Tens1,Tens2)

DOUBLE␣PRECISION␣Tens1(3,3),Tens2(3,3)

INTEGER␣i,j
do␣i=1,3
do␣j=1,3
Tens1(i,j)=Tens2(i,j)
end␣do
end␣do
END

c***********************************************************************
SUBROUTINE␣ADDTENSSC(TensRes,Tens1,Tens2,fact1,fact2)

DOUBLE␣PRECISION␣TensRes(3,3),Tens1(3,3),Tens2(3,3),fact1,fact2

INTEGER␣i,j
do␣i=1,3
do␣j=1,3
TensRes(i,j)=Tens1(i,j)*fact1+Tens2(i,j)*fact2
end␣do
```

```
end␣do
END

c**********************************************************************
SUBROUTINE␣ADDTENSSC6x6(TensRes,Tens1,Tens2,fact1,fact2)

DOUBLE␣PRECISION␣TensRes(6,6),Tens1(6,6),Tens2(6,6),fact1,fact2

INTEGER␣i,j
do␣i=1,6
do␣j=1,6
TensRes(i,j)=Tens1(i,j)*fact1+Tens2(i,j)*fact2
end␣do
end␣do
END

c**********************************************************************
SUBROUTINE␣ADDTENS4_function(TensRes,Tens1,Tens2)

DOUBLE␣PRECISION␣TensRes(3,3,3,3),Tens1(3,3,3,3),Tens2(3,3,3,3)

INTEGER␣i,j,k,l
do␣i=1,3
do␣j=1,3
do␣k=1,3
do␣l=1,3
TensRes(i,j,k,l)=Tens1(i,j,k,l)+Tens2(i,j,k,l)
end␣do
end␣do
end␣do
end␣do
END

c**********************************************************************
SUBROUTINE␣Unit_tensor4_function(TensRes)

DOUBLE␣PRECISION␣TensRes(3,3,3,3),Tens1(3,3,3,3),Tens2(3,3,3,3)

DOUBLE␣PRECISION␣delta_function

INTEGER␣i,j,k,l
do␣i=1,3
do␣j=1,3
do␣k=1,3
do␣l=1,3
TensRes(i,j,k,l)=1.0d0/2.0d0*(delta_function(i,l)*
&␣delta_function(j,k)+delta_function(i,k)*delta_function(j,l))
end␣do
end␣do
end␣do
end␣do
END

c**********************************************************************
FUNCTION␣delta_function(i,j)

DOUBLE␣PRECISION␣delta_function

INTEGER␣i,j
IF␣(i.eq.j)␣delta_function=1.0d0
IF␣(i.ne.j)␣delta_function=0.0d0
RETURN
END

c**********************************************************************
FUNCTION␣det(T)

DOUBLE␣PRECISION␣det,T(3,3)
```

```
det=T(1,1)*T(2,2)*T(3,3)+T(2,1)*T(3,2)*T(1,3)+T(1,2)*T(2,3)*T(3,1)
&␣-T(1,2)*T(2,1)*T(3,3)-T(2,3)*T(3,2)*T(1,1)-T(3,1)*T(2,2)*T(1,3)
RETURN
END

c␣Returns␣Inverse␣of␣the␣matrix
SUBROUTINE␣Inverse_function(TensRes,Tensr,fact)

DOUBLE␣PRECISION␣TensRes(3,3),Tensr(3,3),fact,d,det

INTEGER␣i,j
TensRes(1,1)=Tensr(2,2)*Tensr(3,3)-Tensr(2,3)*Tensr(3,2)
TensRes(1,2)=Tensr(1,3)*Tensr(3,2)-Tensr(3,3)*Tensr(1,2)
TensRes(1,3)=Tensr(1,2)*Tensr(2,3)-Tensr(2,2)*Tensr(1,3)
TensRes(2,1)=Tensr(2,3)*Tensr(3,1)-Tensr(3,3)*Tensr(2,1)
TensRes(2,2)=Tensr(1,1)*Tensr(3,3)-Tensr(1,3)*Tensr(3,1)
TensRes(2,3)=Tensr(1,3)*Tensr(2,1)-Tensr(2,3)*Tensr(1,1)
TensRes(3,1)=Tensr(2,1)*Tensr(3,2)-Tensr(2,2)*Tensr(3,1)
TensRes(3,2)=Tensr(1,2)*Tensr(3,1)-Tensr(1,1)*Tensr(3,2)
TensRes(3,3)=Tensr(2,2)*Tensr(1,1)-Tensr(2,1)*Tensr(1,2)
d=det(Tensr)
do␣j=1,3
do␣i=1,3
TensRes(i,j)=TensRes(i,j)*fact/d
end␣do
end␣do
return
end
c**********************************************************************
subroutine␣matrix_print(A,row,line,n)
integer␣i,j,row,line
DOUBLE␣PRECISION␣A(row,line)
CHARACTER(LEN=*)␣n
print␣*,␣n
do␣i=1,row
do␣j=1,line␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
print␣*,␣'|',␣A(i,j),␣'|-(',i,',',j,')'
end␣do
end␣do
end␣subroutine

c**********************************************************************
subroutine␣tensor4_print(A,row1,row2,row3,row4,n)
integer␣i,j,row1,row2,row3,row4,k,l
DOUBLE␣PRECISION␣A(row1,row2,row3,row4)
CHARACTER(LEN=*)␣n
print␣*,␣n
do␣i=1,row1
do␣j=1,row2␣␣␣␣␣
do␣k=1,row3
do␣l=1,row4
print␣*,␣'|',i,j,k,l,'|'
print␣*,␣'|',␣A(i,j,k,l),␣'|'
end␣do
end␣do
end␣do␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
```

```
Tss(1,5)=Ttttt(1,1,2,3)
Tss(1,6)=Ttttt(1,1,1,3)
Tss(2,1)=Ttttt(2,2,1,1)
Tss(2,2)=Ttttt(2,2,2,2)
Tss(2,3)=Ttttt(2,2,3,3)
Tss(2,4)=Ttttt(2,2,1,2)
Tss(2,5)=Ttttt(2,2,2,3)
Tss(2,6)=Ttttt(2,2,1,3)
Tss(3,1)=Ttttt(3,3,1,1)
Tss(3,2)=Ttttt(3,3,2,2)
Tss(3,3)=Ttttt(3,3,3,3)
Tss(3,4)=Ttttt(3,3,1,2)
Tss(3,5)=Ttttt(3,3,2,3)
Tss(3,6)=Ttttt(3,3,1,3)
Tss(4,1)=Ttttt(1,2,1,1)
Tss(4,2)=Ttttt(1,2,2,2)
Tss(4,3)=Ttttt(1,2,3,3)
Tss(4,4)=Ttttt(1,2,1,2)
Tss(4,5)=Ttttt(1,2,2,3)
Tss(4,6)=Ttttt(1,2,1,3)
Tss(5,1)=Ttttt(2,3,1,1)
Tss(5,2)=Ttttt(2,3,2,2)
Tss(5,3)=Ttttt(2,3,3,3)
Tss(5,4)=Ttttt(2,3,1,2)
Tss(5,5)=Ttttt(2,3,2,3)
Tss(5,6)=Ttttt(2,3,1,3)
Tss(6,1)=Ttttt(1,3,1,1)
Tss(6,2)=Ttttt(1,3,2,2)
Tss(6,3)=Ttttt(1,3,3,3)
Tss(6,4)=Ttttt(1,3,1,2)
Tss(6,5)=Ttttt(1,3,2,3)
Tss(6,6)=Ttttt(1,3,1,3)
end

c ***********************************************************************
subroutine polarRU(F,R,U)
!-----------------------------------------------------------------------
!-  Polar decomposition, F=RU
!     Reference:
!     JC Simo & TJR Hughes, Computational Inelasticity, 1998, Springer
!-----------------------------------------------------------------------
IMPLICIT NONE
INTEGER,PARAMETER :: RK=KIND(1.D0) ! real kind
! arguments
REAL(RK),INTENT(IN) :: F(3,3) ! matrix to decompose
REAL(RK),INTENT(OUT) :: R(3,3) ! rotation
REAL(RK),INTENT(OUT) :: U(3,3) ! stretch
! local
REAL(RK), PARAMETER :: pi    = 3.14159265358979324_RK
REAL(RK),PARAMETER :: I(3,3)=reshape((/1._RK,0._RK,0._RK,
&                                      0._RK,1._RK,0._RK,
&                                      0._RK,0._RK,1._RK/),(/3,3/))
REAL(RK) :: C(3,3),CC(3,3) ! right Cauchy-Green, squared
REAL(RK) :: Ui(3,3) ! U inverse
REAL(RK) :: IC, IIC, IIIC ! invariants of C
REAL(RK) :: IU, IIU, IIIU ! invariants of U
REAL(RK) :: l1,l2,l3 ! eigenvalues
REAL(RK) :: p,q,m,n,t,D ! constants
!-----------------------------------------------------------------------
c right Cauchy-Green
call MULTENS_function(C,transpose(F),F,1.0d0)
call MULTENS_function(CC,C,C,1.0d0)

c invariants
IC  = C(1,1) + C(2,2) + C(3,3)
IIC = 0.5_RK*(IC**2 - (CC(1,1) + CC(2,2) + CC(3,3)))
IIIC = C(1,1) * (C(2,2)*C(3,3) - C(2,3)*C(3,2))
&      + C(1,2) * (C(2,3)*C(3,1) - C(2,1)*C(3,3))
```

```
&     +C(1,3) *(C(2,1)*C(3,2) -C(2,2)*C(3,1))

c eigenvalues of sqrt(C)
p = IIC - (IC**2)/3._RK
q = -(2._RK/27._RK)*IC**3+IC*IIC/3._RK-IIIC
if(abs(p)<       epsilon(1._RK))then
l1 = sqrt( abs(-abs(q)**(1._RK/3._RK) + IC/3._RK) )
l2 = l1
l3 = l2
else
m = 2._RK*sqrt(abs(p)/3._RK)
n = 3._RK*q/(m*p)
if(abs(n)>1._RK) n = sign(1._RK, n)
t = atan2(sqrt(1._RK-n**2),n)/3._RK
l1 = sqrt( abs(m*cos(t) + IC/3._RK) )
l2 = sqrt( abs(m*cos(t+2._RK/3._RK*pi) + IC/3._RK) )
l3 = sqrt( abs(m*cos(t+4._RK/3._RK*pi) + IC/3._RK) )
endif

c stretch and inverse
!invariants
IU = l1 + l2 + l3
IIU = l1*l2 + l1*l3 + l2*l3
IIIU = l1*l2*l3
D = IU*IIU-IIIU

U(:,:)   =(-CC(:,:) + (IU**2-IIU)*C(:,:) + IU*IIIU*I(:,:))/D
Ui(:,:) = (C(:,:) - IU*U(:,:) + IIU*I(:,:))/IIIU

c Rotation
call MULTENS_function(R,F,Ui,1.0d0)

end subroutine

c*********************************************************************
SUBROUTINE trans_matrx_6(Q,R)
!-----------------------------------------------------------------------
! Construct the 6x6 Voigt rotation matrix
!-----------------------------------------------------------------------
IMPLICIT NONE
INTEGER,PARAMETER :: RK=KIND(1.D0) ! real kind
! arguments
REAL(RK), INTENT(OUT) :: Q(6,6)
REAL(RK), INTENT(IN) :: R(3,3)
!-----------------------------------------------------------------------
Q(1,1) = R(1,1)**2
Q(2,1) = R(2,1)**2
Q(3,1) = R(3,1)**2
Q(4,1) = R(1,1)*R(2,1)
Q(5,1) = R(2,1)*R(3,1)
Q(6,1) = R(3,1)*R(1,1)

Q(1,2) = R(1,2)**2
Q(2,2) = R(2,2)**2
Q(3,2) = R(3,2)**2
Q(4,2) = R(1,2)*R(2,2)
Q(5,2) = R(2,2)*R(3,2)
Q(6,2) = R(3,2)*R(1,2)

Q(1,3) = R(1,3)**2
Q(2,3) = R(2,3)**2
Q(3,3) = R(3,3)**2
Q(4,3) = R(1,3)*R(2,3)
Q(5,3) = R(2,3)*R(3,3)
Q(6,3) = R(3,3)*R(1,3)

Q(1,4) = R(1,1)*R(1,2)*2._RK
Q(2,4) = R(2,1)*R(2,2)*2._RK
```

```
Q(3,4) = R(3,1)*R(3,2)*2._RK
Q(4,4) = R(1,1)*R(2,2)+R(1,2)*R(2,1)
Q(5,4) = R(2,1)*R(3,2)+R(2,2)*R(3,1)
Q(6,4) = R(3,1)*R(1,2)+R(3,2)*R(1,1)

Q(1,5) = R(1,2)*R(1,3)*2._RK
Q(2,5) = R(2,2)*R(2,3)*2._RK
Q(3,5) = R(3,2)*R(3,3)*2._RK
Q(4,5) = R(1,2)*R(2,3)+R(1,3)*R(2,2)
Q(5,5) = R(2,2)*R(3,3)+R(2,3)*R(3,2)
Q(6,5) = R(3,2)*R(1,3)+R(3,3)*R(1,2)

Q(1,6) = R(1,1)*R(1,3)*2._RK
Q(2,6) = R(2,1)*R(2,3)*2._RK
Q(3,6) = R(3,1)*R(3,3)*2._RK
Q(4,6) = R(1,1)*R(2,3)+R(1,3)*R(2,1)
Q(5,6) = R(2,1)*R(3,3)+R(2,3)*R(3,1)
Q(6,6) = R(3,1)*R(1,3)+R(3,3)*R(1,1)
END SUBROUTINE

c**********************************************************************
subroutine extra_stiffness(aux_matrix,s)

Double Precision aux_matrix(6,6), s(6)
!----------------------------------------------------------------------
! Construct the extra term for the Jaumann tangent stiffness matrix
! page 617 from Ansys Technology demonstration guide
!----------------------------------------------------------------------
aux_matrix(1,1)=2.0d0*s(1)
aux_matrix(1,2)=0.0d0
aux_matrix(1,3)=0.0d0
aux_matrix(1,4)=s(4)
aux_matrix(1,5)=0.0d0
aux_matrix(1,6)=s(6)
aux_matrix(2,1)=0.0d0
aux_matrix(2,2)=2.0d0*s(2)
aux_matrix(2,3)=0.0d0
aux_matrix(2,4)=s(4)
aux_matrix(2,5)=s(5)
aux_matrix(2,6)=0.0d0
aux_matrix(3,1)=0.0d0
aux_matrix(3,2)=0.0d0
aux_matrix(3,3)=2.0d0*s(3)
aux_matrix(3,4)=0.0d0
aux_matrix(3,5)=s(5)
aux_matrix(3,6)=s(6)
aux_matrix(4,1)=s(4)
aux_matrix(4,2)=s(4)
aux_matrix(4,3)=0.0d0
aux_matrix(4,4)=(s(1)+s(2))/2.0d0
aux_matrix(4,5)=s(6)/2.0d0
aux_matrix(4,6)=s(5)/2.0d0
aux_matrix(5,1)=0.0d0
aux_matrix(5,2)=s(5)
aux_matrix(5,3)=s(5)
aux_matrix(5,4)=s(6)/2.0d0
aux_matrix(5,5)=(s(3)+s(2))/2.0d0
aux_matrix(5,6)=s(4)/2.0d0
aux_matrix(6,1)=s(6)
aux_matrix(6,2)=0.0d0
aux_matrix(6,3)=s(6)
aux_matrix(6,4)=s(5)/2.0d0
aux_matrix(6,5)=s(4)/2.0d0
aux_matrix(6,6)=(s(3)+s(1))/2.0d0

end subroutine

c ******************************************************************
```

```
subroutine extra_stiffness_tensor(aux_matrix,s)

Double Precision aux_matrix(3,3,3,3), s(3,3),delta_function
integer i,j,k,l
do i=1,3
do j=1,3
do k=1,3
do l=1,3
aux_matrix(i,j,k,l)=1.0d0/2.0d0*
&                   (delta_function(i,k)*s(j,l)+
&                   delta_function(j,l)*s(i,k)+
&                   delta_function(i,l)*s(j,k)+
&                   delta_function(j,k)*s(i,l))
end do
end do
end do
end do
end subroutine


c ********************************************************************
subroutine heaviside(x)

Double Precision x, heaviside3
heaviside3=0.0d0

IF (x .ge. 0.0d0) THEN
heaviside3=1.0d0
END IF

end subroutine

c ********************************************************************
subroutine fiber_passive(p_ind,lambda_oft,lambda_star,lambda,
& sigma_max,sigma_passive)

Double Precision  lambda_oft, lambda_star, lambda,
& force_passive,ratio_lambda,sigma_max,sigma_passive,p_ind(4)

ratio_lambda=lambda/lambda_oft

IF (lambda .le. lambda_oft) THEN
force_passive=0.0d0
ELSE IF (lambda .lt. lambda_star) THEN
force_passive=p_ind(1)*(exp(p_ind(2)*(ratio_lambda-1.0d0))
&                   -1.0d0)
ELSE
force_passive=p_ind(3)*ratio_lambda+p_ind(4)
END IF

sigma_passive=max(force_passive,0.0d0)*sigma_max*ratio_lambda


end subroutine

c ********************************************************************
subroutine fiber_active(lambda_oft,lambda_star,lambda,
& sigma_max,sigma_active,alpha)

Double Precision lambda_oft, lambda_star, lambda,
& force_active,ratio_lambda,sigma_max,sigma_active,alpha

ratio_lambda=lambda/lambda_oft

IF (lambda .lt. lambda_oft*0.4d0 .or.
&    lambda .ge. lambda_oft*1.6d0) THEN
force_active=0.0d0
ELSE IF(lambda .le. lambda_oft*0.6d0) THEN
```

```
force_active=9.0d0*(ratio_lambda-0.40d0)**2.0d0
ELSE IF (lambda .lt. lambda_oft*1.4d0) THEN
force_active=1.0d0-4.0d0*(1.0d0-ratio_lambda)**2.0d0
ELSE
force_active=9.0d0*(ratio_lambda-1.6d0)**2.0d0
END IF

sigma_active=max(force_active,0.0d0)*alpha*
&          sigma_max*lambda/lambda_oft

end subroutine

c ********************************************************************

SUBROUTINE GetStress(IO,Inv,JJ,C,Cinv,F,FT,Cbar,k_ind,p,ptilda,
& SigmaIsoVoight,SigmaIso,SigmaVol,Siso,SBar,g,F_A,lambda,
& Identity,ncomp,stress_spatial,sigma_matrix,p_ind,sigma_max,
& lambda_star,lambda_oft,sigma_passive,sigma_active,
& sigma_muscle_module,alpha,F_A_current,BBar,SigmaIso_cross,
& SigmaIso_along)

DOUBLE PRECISION JJ,aaa,delta_function,DoubleDot_function,
& SBar_1_ind,SBar_2_ind,SBar_4_ind,SBar_5_ind,p,ptilda,k_ind,
& b2,gmod2,y,b2_truncate,lambda,lambda_star,lambda_oft,
& sigma_muscle_module,alpha,sigma_passive,sigma_muscle_module2,
& sigma_active, JJ_mod,dev_1,dev_2,dev_4,dev_5,index_1,index_2,
& index_4,index_5,index_extra

DOUBLE PRECISION C(3,3),Cinv(3,3),SigmaIso(3,3),
& SigmaIsoVoight(6),Identity(3,3),Cbar(3,3),
& Siso(3,3),Sbar(3,3),T1(3,3),T2(3,3),F(3,3),FT(3,3),
& SBar_1(3,3),SBar_2(3,3),SBar_4(3,3),SBar_5,SBar_2_matrix(3,3),
& SBar_4_matrix(3,3),SBar_5_matrix(3,3),Inv(6,1),g(2,1),F_A(3,1),
& SigmaVol(3,3),stress_spatial(6),sigma_matrix(3,3),
& SBar_5_vec1(1,3),SBar_5_vec2(3,1),SBar_5_matrix1(3,3),
& SBar_5_matrix2(3,3),p_ind(4),SigmaMuscle(3,3),F_A_current(3,1),
& SBar_passive(3,3),BBar(3,3),BBar2(3,3),AXA(3,3),AbA(3,3),
& AbA1_vec(1,3),AbA2_vec(3,1),AbA1(3,3),AbA2(3,3),
& SigmaIso_test(3,3),SigmaIso_cross(3,3),SigmaIso_along(3,3),
& SigmaIso_fiber(3,3)


INTEGER ic,j,IO,ncomp


c Defining helping constants
b2=(Inv(1,1)*Inv(4,1)-Inv(5,1))/(2.0d0*sqrt(Inv(4,1)))

gmod2=0.0d0

IF (abs(b2-1.0d0) .LE. 1.0d-4) THEN
gmod2=g(2,1)
ELSE
gmod2=g(2,1)*log(b2+sqrt(b2**2.0d0-1.0d0))/
&   sqrt(b2**2.0d0-1.0d0)

END IF




c Calculate SigmaVol - OBS:p IS DEPENDENT OF THE STRAIN ENERGY
p=k_ind*log(JJ)/JJ

call ScalTensor_function(SigmaVol,Identity,p)


c Calculate the fiber part of the stress
```

```
call␣fiber_active(lambda_oft,lambda_star,lambda,
&␣sigma_max,sigma_active,alpha)

call␣fiber_passive(p_ind,lambda_oft,lambda_star,lambda,
&␣sigma_max,sigma_passive)␣␣␣␣␣␣
sigma_muscle_module=sigma_active+sigma_passive



c␣Calculate␣SBar␣-␣OBS:␣aaa␣dependent␣of␣the␣strain␣energy␣-␣cross␣fiber␣shear
dev_1=gmod2*sqrt(Inv(4,1))␣␣␣␣␣␣

dev_2=0.0d0␣␣␣␣␣

dev_4=0.5d0*gmod2*(Inv(1,1)*Inv(4,1)+Inv(5,1))
&␣/(Inv(4,1)**1.5d0)

dev_5=-gmod2/sqrt(Inv(4,1))


c␣Calculate␣SigmaIso␣-␣cross␣fiber␣shear
index_1=dev_1+Inv(1,1)*dev_2
index_2=-dev_2
index_4=Inv(4,1)*dev_4
index_5=Inv(4,1)*dev_5
index_extra=(dev_1*Inv(1,1)+2.d0*dev_2*Inv(2,1)+Inv(4,1)*dev_4+
&␣2.0d0*Inv(5,1)*dev_5)*(-1.0d0/3.0d0)␣␣

call␣MulTens_function(BBar2,BBar,BBar,1.0d0)
call␣TensorProduct_Vector(AXA,F_A_current,F_A_current)

!~~calculating␣a.BBar␣tensorproduct␣a~~!␣␣␣␣␣␣␣␣
call␣T13dotT33(AbA1_vec,transpose(F_A_current),BBar,1.0d0)␣␣␣␣␣␣
call␣TensorProduct_Vector(AbA1,transpose(AbA1_vec),F_A_current)

!~~calculating␣a␣tensorproduct␣BBar.a~~!
call␣T33dotT31(AbA2_vec,F_A_current,BBar,1.0d0)
call␣TensorProduct_Vector(AbA2,F_A_current,AbA2_vec)

!~~adding␣AbA␣1␣and␣2~~!
call␣AddTensSc(AbA,AbA2,AbA1,1.0d0,1.0d0)


!~~T␣=␣B*(W1␣+␣W2*I1)␣-␣B2*W2␣+␣AxA*(I4*W4)␣+␣ABA*(I4*W5);
call␣AddTensSc(SigmaIso_cross,BBar,BBar2,index_1,index_2)
call␣AddTensSc(SigmaIso_cross,SigmaIso_cross,AXA,1.0d0,index_4)
call␣AddTensSc(SigmaIso_cross,SigmaIso_cross,AbA,1.0d0,index_5)

call␣DEVIATOR(SigmaIso_cross,3.0d0,SigmaIso_cross)␣␣␣␣␣␣
call␣ScalTensor_function(SigmaIso_cross,SigmaIso_cross,2.0d0/JJ)


c␣Calculate␣SBar␣-␣OBS:␣aaa␣dependent␣of␣the␣strain␣energy␣-␣along␣fiber␣shear
dev_1=0.0d0␣␣␣␣␣

dev_2=0.0d0␣␣␣␣␣

dev_4=-2.0d0*g(1,1)*Inv(5,1)/Inv(4,1)**3.0d0


dev_5=g(1,1)/(Inv(4,1))**2.0d0


c␣Calculate␣SigmaIso␣-␣along␣fiber␣shear
index_1=dev_1+Inv(1,1)*dev_2
index_2=-dev_2
index_4=Inv(4,1)*dev_4
index_5=Inv(4,1)*dev_5
```

```
index_extra=(dev_1*Inv(1,1)+2.d0*dev_2*Inv(2,1)+Inv(4,1)*dev_4+
&␣2.0d0*Inv(5,1)*dev_5)*(-1.0d0/3.0d0)␣␣␣

call␣MulTens_function(BBar2,BBar,BBar,1.0d0)
call␣TensorProduct_Vector(AXA,F_A_current,F_A_current)

!~~calculating␣a.BBar␣tensorproduct␣a~~!␣␣␣␣␣␣␣␣
call␣T13dotT33(AbA1_vec,transpose(F_A_current),BBar,1.0d0)␣␣␣␣␣␣
call␣TensorProduct_Vector(AbA1,transpose(AbA1_vec),F_A_current)

!~~calculating␣a␣tensorproduct␣BBar.a~~!
call␣T33dotT31(AbA2_vec,F_A_current,BBar,1.0d0)
call␣TensorProduct_Vector(AbA2,F_A_current,AbA2_vec)

!~~adding␣AbA␣1␣and␣2~~!
call␣AddTensSc(AbA,AbA2,AbA1,1.0d0,1.0d0)


!~~T␣=␣B*(W1␣+␣W2*I1)␣-␣B2*W2␣+␣AxA*(I4*W4)␣+␣ABA*(I4*W5);
call␣AddTensSc(SigmaIso_along,BBar,BBar2,index_1,index_2)
call␣AddTensSc(SigmaIso_along,SigmaIso_along,AXA,1.0d0,index_4)
call␣AddTensSc(SigmaIso_along,SigmaIso_along,AbA,1.0d0,index_5)
call␣DEVIATOR(SigmaIso_along,3.0d0,SigmaIso_along)␣␣␣␣␣␣
call␣ScalTensor_function(SigmaIso_along,SigmaIso_along,2.0d0/JJ)


c␣Calculate␣SBar␣-␣OBS:␣aaa␣dependent␣of␣the␣strain␣energy␣-␣fiber␣strech
dev_1=0.0d0␣␣␣␣␣

dev_2=0.0d0␣␣␣␣␣

dev_4=sigma_muscle_module/(2.0d0*lambda**2.0d0)

dev_5=0.0d0


c␣Calculate␣SigmaIso␣-␣fiber␣strech
index_1=dev_1+Inv(1,1)*dev_2
index_2=-dev_2
index_4=Inv(4,1)*dev_4
index_5=Inv(4,1)*dev_5
index_extra=(dev_1*Inv(1,1)+2.d0*dev_2*Inv(2,1)+Inv(4,1)*dev_4+
&␣2.0d0*Inv(5,1)*dev_5)*(-1.0d0/3.0d0)␣␣␣

call␣MulTens_function(BBar2,BBar,BBar,1.0d0)
call␣TensorProduct_Vector(AXA,F_A_current,F_A_current)

!~~calculating␣a.BBar␣tensorproduct␣a~~!␣␣␣␣␣␣␣␣
call␣T13dotT33(AbA1_vec,transpose(F_A_current),BBar,1.0d0)␣␣␣␣␣␣
call␣TensorProduct_Vector(AbA1,transpose(AbA1_vec),F_A_current)

!~~calculating␣a␣tensorproduct␣BBar.a~~!
call␣T33dotT31(AbA2_vec,F_A_current,BBar,1.0d0)
call␣TensorProduct_Vector(AbA2,F_A_current,AbA2_vec)

!~~adding␣AbA␣1␣and␣2~~!
call␣AddTensSc(AbA,AbA2,AbA1,1.0d0,1.0d0)


!~~T␣=␣B*(W1␣+␣W2*I1)␣-␣B2*W2␣+␣AxA*(I4*W4)␣+␣ABA*(I4*W5);
call␣AddTensSc(SigmaIso_fiber,BBar,BBar2,index_1,index_2)
call␣AddTensSc(SigmaIso_fiber,SigmaIso_fiber,AXA,1.0d0,index_4)
call␣AddTensSc(SigmaIso_fiber,SigmaIso_fiber,AbA,1.0d0,index_5)
call␣DEVIATOR(SigmaIso_fiber,3.0d0,SigmaIso_fiber)␣␣␣␣␣␣
call␣ScalTensor_function(SigmaIso_fiber,SigmaIso_fiber,2.0d0/JJ)


c␣Adding␣the␣stress␣from␣the␣cross␣and␣along␣shear␣stress
```

```
call␣AddTensSc(SigmaIso,SigmaIso_cross,SigmaIso_along,
&␣1.0d0,1.0d0)
call␣AddTensSc(SigmaIso,SigmaIso,SigmaIso_fiber,1.0d0,1.0d0)


c␣Form␣output␣stress␣vector
call␣ADDTENSSC(sigma_matrix,SigmaVol,SigmaIso,1.0d0,1.0d0)␣␣␣␣␣␣

stress_spatial(1)=sigma_matrix(1,1)
stress_spatial(2)=sigma_matrix(2,2)
stress_spatial(3)=sigma_matrix(3,3)
stress_spatial(4)=sigma_matrix(1,2)
IF␣(ncomp.eq.6)␣THEN
stress_spatial(5)=sigma_matrix(2,3)
stress_spatial(6)=sigma_matrix(1,3)
END␣IF

c␣Creating␣output␣stress␣vector
SigmaIsoVoight(1)=SigmaIso(1,1)
SigmaIsoVoight(2)=SigmaIso(2,2)
SigmaIsoVoight(3)=SigmaIso(3,3)
SigmaIsoVoight(4)=SigmaIso(1,2)
SigmaIsoVoight(5)=SigmaIso(2,3)
SigmaIsoVoight(6)=SigmaIso(1,3)


END


c␣*****************************************************************

SUBROUTINE␣c_matrix_and_inv(C,Cinv,FT,FBar,FbarT,Cbar,F,F_A,JJ,
&␣BBar,Inv,lambda,F_A_current)␣␣


DOUBLE␣PRECISION␣JJ,det,aux,lambda,JJ_mod

DOUBLE␣PRECISION␣C(3,3),Cinv(3,3),FT(3,3),Fbar(3,3),FbarT(3,3),
&␣T2(3,3),F(3,3),Identity(3,3),Bbar(3,3),F_A_current(3,1),
&␣Inv(6,1),T1(3,3),Cbar(3,3),F_A(3,1),
&␣Cbar_squared(3,3)␣␣␣␣␣

INTEGER␣i,j,k,l


c␣Calculate␣determinant␣of␣defGrad
JJ=det(F)

if(jj<=0.0)then
keycut=1
return
else
endif
c␣Calculate␣right␣Cauchy-Green␣deformation␣tensor
call␣Transpose_function(FT,F)
call␣MulTens_function(C,FT,F,1.0d0)

c␣Calculate␣C-inverse
call␣Inverse_function(Cinv,C,1.0d0)

c␣Calculate␣the␣barred␣values␣Cbar
JJ_mod=JJ**(-2.0d0)
JJ_mod=JJ_mod**(1.0d0/3.0d0)
call␣ScalTensor_function(FBar,F,JJ**(-1.0d0/3.0d0))
call␣Transpose_function(FBarT,FBar)
call␣ScalTensor_function(CBar,C,JJ_mod)
call␣MulTens_function(Cbar_squared,CBar,CBar,1.0d0)
```

```
c␣Calculate␣the␣barred␣values␣Bbar
call␣MulTens_function(Bbar,FBar,FBarT,1.0d0)

call␣MulTens_function(T2,Bbar,Bbar,1.0d0)␣␣␣␣␣␣␣

c␣Calculate␣the␣invariants
aux=Cbar_squared(1,1)+Cbar_squared(2,2)+Cbar_squared(3,3)
Inv(1,1)=CBar(1,1)+CBar(2,2)+CBar(3,3)
Inv(2,1)=1.0d0/2.0d0*((Inv(1,1))**2.0d0-aux)
Inv(3,1)=det(CBar)

!␣calculating␣I4bar
Inv(4,1)=0.0d0
do␣i=1,3
do␣j=1,3
Inv(4,1)=Inv(4,1)+Cbar(i,j)*F_A(i,1)*F_A(j,1)
end␣do
end␣do

!␣calculating␣I5bar
Inv(5,1)=0.0d0
do␣i=1,3
do␣j=1,3
Inv(5,1)=Inv(5,1)+Cbar_squared(i,j)*F_A(i,1)*F_A(j,1)
end␣do
end␣do

c␣Calculating␣lambda␣-␣fiber␣stretch␣␣␣␣
lambda=sqrt(Inv(4,1))

c␣Create␣the␣current␣fiber␣direction␣vector
call␣T33dotT31(F_A_current,F_A,FBar,1.0d0/lambda)

END
c**********************************************************************
SUBROUTINE␣delta_F(dF,F,i,j,SMALL,R)

INTEGER␣i,j,a,b

DOUBLE␣PRECISION␣F(3,3),dF(3,3),aux(3,3),dF1(3,3),R(3,3)

DOUBLE␣PRECISION␣SMALL


c␣creating␣the␣ei␣x␣ej␣␣
do␣a=1,3
do␣b=1,3
aux(a,b)=R(a,i)*R(b,j)
end␣do
end␣do␣␣␣␣␣␣␣␣␣


c␣creating␣the␣delta␣F␣matrix
call␣MulTens_function(dF1,aux,F,SMALL/2.0d0)
call␣MulTens_function(dF,transpose(aux),F,SMALL/2.0d0)
call␣AddTensSc(dF,dF,dF1,1.0d0,1.0d0)

c␣creating␣the␣F␣pertubed
call␣AddTensSc(dF,dF,F,1.0d0,1.0d0)␣␣␣␣␣

END
c**********************************************************************
SUBROUTINE␣DEVIATOR(matrix_out,dimen,matrix_in)

INTEGER␣i,j
```

```
DOUBLE␣PRECISION␣dimen,␣matrix_out(dimen,dimen),t,
&␣matrix_in(dimen,dimen)␣␣␣␣␣␣␣␣

t=0.0d0
do␣i=1,dimen
t=t+matrix_in(i,i)/3.0d0␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
end␣do␣␣

do␣i=1,dimen
do␣j=1,dimen
if␣(i␣.eq.␣j)␣then␣␣␣␣␣␣␣␣␣␣␣␣
matrix_out(i,j)=matrix_in(j,j)-t
else
matrix_out(i,j)=matrix_in(i,j)
end␣if␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣
end␣do␣␣␣␣␣␣␣␣␣
end␣do␣␣


END␣␣␣␣␣
c**********************************************************************
SUBROUTINE␣PASSIVE_ENERGY(p_ind,lambda_oft,lambda_star,lambda,
&␣sigma_max,energy)

Double␣Precision␣␣lambda_oft,␣lambda_star,␣lambda,
&␣force_passive,sigma_max,energy,p_ind(4),c1,c2



IF␣(lambda␣.le.␣lambda_oft)␣THEN

energy=0.0d0

ELSE␣IF␣(lambda␣.lt.␣lambda_star)␣THEN
c2=lambda_oft*(p_ind(1)-p_ind(1)/p_ind(2))

energy=p_ind(1)*(exp(p_ind(2)*(lambda/lambda_oft-1.0d0))*
&␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣␣lambda_oft/p_ind(2)-lambda)+c2

ELSE
c2=lambda_oft*(p_ind(1)-p_ind(1)/p_ind(2))

c1=c2-(p_ind(1)+p_ind(4))*lambda_star
&␣␣␣␣␣␣␣␣-p_ind(3)*lambda_star**2.0d0/2.0d0/lambda_oft
&␣␣␣␣␣␣␣␣+p_ind(1)*(exp(p_ind(2)*(lambda_star/lambda_oft-1.0d0)))*
&␣␣␣␣␣␣␣␣lambda_oft/p_ind(2)␣␣␣␣

energy=p_ind(3)*lambda**2.0d0/(2.0d0*lambda_oft)+
&␣p_ind(4)*lambda+c1

END␣IF␣␣

energy=energy*sigma_max/lambda_oft

END␣␣␣␣␣
c**********************************************************************
SUBROUTINE␣ACTIVE_ENERGY(energy,lambda_oft,lambda_star,lambda,
&␣sigma_max,alpha)


DOUBLE␣PRECISION␣lambda_oft,lambda_star,lambda,
&␣sigma_max,energy,alpha

IF␣(lambda␣.lt.␣lambda_oft*0.4d0␣.or.
&␣␣␣␣lambda␣.ge.␣lambda_oft*1.6d0)␣THEN

energy=0.0d0
```

```
ELSE IF(lambda .le. lambda_oft*0.6d0) THEN

c1=-0.192d0*lambda_oft

energy=9.0d0*(lambda**3.0d0/(3.0d0*lambda_oft**2.0d0)-
&  0.40d0*lambda**2.0d0/lambda_oft+0.16d0*lambda)+c1

ELSE IF (lambda .lt. lambda_oft*1.4d0) THEN

c2=0.672d0*lambda_oft

energy=(-4.0d0*lambda**3.0d0/(3.0d0*lambda_oft**2.0d0)+
&  4.0d0*lambda**2.0d0/lambda_oft-3.0d0*lambda)+c2

ELSE

c3=-34.832d0/3.0d0*lambda_oft

energy=9.0d0*(lambda**3.0d0/(3.0d0*lambda_oft**2.0d0)-
&  1.60d0*lambda**2.0d0/lambda_oft+2.56d0*lambda)+c3

END IF

energy=energy*alpha*sigma_max/lambda_oft


END
c**********************************************************************
SUBROUTINE GetStiffness_approx(CC_approx,F,JJ,g, F_A,k_ind,
&   sigma_matrix,Identity,SMALL,R,dsdePl,p_ind,sigma_max,
&   lambda_star,lambda_oft,alpha)

INTEGER i,j,k,l,a_count,b_count,ncomp,ic,jc,kc,lc

DOUBLE PRECISION SMALL,JJ,dJJ,d,dp,dptilda,k_ind,sigma_max,
&  lambda_oft,lambda_star,alpha,dlambda

DOUBLE PRECISION F(3,3),sigma_matrix(3,3),F_A(3,1),g(2,1),
&  inc_matrix(2,6),Identity(3,3),test(3,3),R(3,3),dF_A_current(3,1)

DOUBLE PRECISION dF(3,3),dC(3,3),dCinv(3,3),dFT(3,3),dInv(6,1),
&  dBbar(3,3),dFBar(3,3),dFbarT(3,3),dSigmaVol(3,3),dCbar(3,3),
&  dSigmaIso(3,3), dSiso(3,3),dSBar(3,3),dstress_spatial(3,3),
&  dsigma_matrix(3,3),dSigmaIsoVoight(6),dsdePl(6,6),p_ind(4),
&  Q(6,6),dsdePl_0(6,6),dsdePl_2(6,6),dSigmaIso_along(3,3),
&  dSigmaIso_cross(3,3)


DOUBLE PRECISION CC_approx(3,3,3,3),dCVol(3,3,3,3),TT1(3,3,3,3),
&  TT2(3,3,3,3),TT3(3,3,3,3),CC_approx_co(3,3,3,3),Cspac(3,3,3,3)


inc_matrix(1,1)=1.0d0
inc_matrix(2,1)=1.0d0
inc_matrix(1,2)=2.0d0
inc_matrix(2,2)=2.0d0
inc_matrix(1,3)=3.0d0
inc_matrix(2,3)=3.0d0
inc_matrix(1,4)=1.0d0
inc_matrix(2,4)=2.0d0
inc_matrix(1,5)=1.0d0
inc_matrix(2,5)=3.0d0
inc_matrix(1,6)=2.0d0
inc_matrix(2,6)=3.0d0

ncomp=6

do k=1,6
```

```
i=inc_matrix(1,k)
j=inc_matrix(2,k)

call␣delta_F(dF,F,i,j,SMALL,R)␣␣␣␣␣␣␣␣␣␣

call␣c_matrix_and_inv(dC,dCinv,dFT,dFBar,dFbarT,dCbar,dF,F_A,dJJ,
&␣␣␣dBBar,dInv,dlambda,dF_A_current)


call␣GetStress(IO,dInv,dJJ,dC,dCinv,dF,dFT,dCbar,k_ind,dp,dptilda,
&␣dSigmaIsoVoight,dSigmaIso,dSigmaVol,dSiso,dSBar,g,F_A,dlambda,
&␣Identity,ncomp,dstress_spatial,dsigma_matrix,p_ind,sigma_max,
&␣lambda_star,lambda_oft,dsigma_passive,dsigma_active,
&␣dsigma_muscle_module,alpha,dF_A_current,dBBar,dSigmaIso_cross,
&␣dSigmaIso_along)

do␣a_count=1,3
do␣b_count=1,3
CC_approx(a_count,b_count,i,j)=1.0d0/(SMALL)*
&␣␣␣␣␣␣␣␣␣␣(dJJ/JJ*dsigma_matrix(a_count,b_count)-
&␣␣␣␣␣␣␣␣␣␣sigma_matrix(a_count,b_count))

CC_approx(a_count,b_count,j,i)=
&␣␣␣␣␣␣␣␣CC_approx(a_count,b_count,i,j)
end␣do
end␣do
end␣do


c␣Transforming␣into␣Voight␣notation
call␣Voight(dsdePl,CC_approx)


END
c**********************************************************************␣␣␣␣␣␣
!␣----------------------------------------------------------------------
SUBROUTINE␣eigsym66(A,␣Q,␣W)
!␣----------------------------------------------------------------------
!␣Calculates␣the␣eigenvalues␣and␣normalized␣eigenvectors␣of
!␣a␣symmetric␣3x3␣matrix␣A␣using␣the␣Jacobi␣algorithm.
!␣The␣upper␣triangular␣part␣of␣A␣is␣destroyed␣during␣the␣calculation,
!␣the␣diagonal␣elements␣are␣read␣but␣not␣destroyed,␣and␣the␣lower
!␣triangular␣elements␣are␣not␣referenced␣at␣all.
!␣----------------------------------------------------------------------
!␣Parameters:
!␣␣␣A:␣The␣symmetric␣input␣matrix
!␣␣␣Q:␣Storage␣buffer␣for␣eigenvectors
!␣␣␣W:␣Storage␣buffer␣for␣eigenvalues
!␣----------------------------------------------------------------------
!␣␣␣␣␣..␣Arguments␣..
DOUBLE␣PRECISION␣A(6,6)
DOUBLE␣PRECISION␣Q(6,6)
DOUBLE␣PRECISION␣W(6)

!␣␣␣␣␣..␣Parameters␣..
INTEGER␣␣␣␣␣␣␣␣␣␣N
PARAMETER␣␣␣␣␣␣␣␣(␣N␣=␣6␣)

!␣␣␣␣␣..␣Local␣Variables␣..
DOUBLE␣PRECISION␣SD,␣SO
DOUBLE␣PRECISION␣S,␣C,␣T
DOUBLE␣PRECISION␣G,␣H,␣Z,␣THETA
DOUBLE␣PRECISION␣THRESH
INTEGER␣␣␣␣␣␣␣␣␣␣I,␣X,␣Y,␣R

!␣␣␣␣␣Initialize␣Q␣to␣the␣identitity␣matrix
!␣␣␣␣␣---␣This␣loop␣can␣be␣omitted␣if␣only␣the␣eigenvalues␣are␣desired␣---
```

```
DO␣10␣X␣=␣1,␣N
Q(X,X)␣=␣1.0D0
DO␣11,␣Y␣=␣1,␣X-1
Q(X,␣Y)␣=␣0.0D0
Q(Y,␣X)␣=␣0.0D0
11␣␣␣CONTINUE
10␣CONTINUE

!␣␣␣␣␣Initialize␣W␣to␣diag(A)
DO␣20␣X␣=␣1,␣N
W(X)␣=␣A(X,␣X)
20␣CONTINUE

!␣␣␣␣␣Calculate␣SQR(tr(A))
SD␣=␣0.0D0
DO␣30␣X␣=␣1,␣N
SD␣=␣SD␣+␣ABS(W(X))
30␣CONTINUE
SD␣=␣SD**2

!␣␣␣␣␣Main␣iteration␣loop
DO␣40␣I␣=␣1,␣50
!␣␣␣␣␣␣␣Test␣for␣convergence
SO␣=␣0.0D0
DO␣50␣X␣=␣1,␣N
DO␣51␣Y␣=␣X+1,␣N
SO␣=␣SO␣+␣ABS(A(X,␣Y))
51␣␣␣␣␣CONTINUE
50␣␣␣CONTINUE
IF␣(SO␣.EQ.␣0.0D0)␣THEN
RETURN
END␣IF

IF␣(I␣.LT.␣4)␣THEN
THRESH␣=␣0.2D0␣*␣SO␣/␣N**2
ELSE
THRESH␣=␣0.0D0
END␣IF

!␣␣␣␣␣␣␣Do␣sweep
DO␣60␣X␣=␣1,␣N
DO␣61␣Y␣=␣X+1,␣N
G␣=␣100.0D0␣*␣(␣ABS(A(X,␣Y))␣)
IF␣(I␣.GT.␣4␣.AND.␣ABS(W(X))␣+␣G␣.EQ.␣ABS(W(X))
&␣␣␣␣␣␣␣␣␣␣.AND.␣ABS(W(Y))␣+␣G␣.EQ.␣ABS(W(Y)))␣THEN
A(X,␣Y)␣=␣0.0D0
ELSE␣IF␣(ABS(A(X,␣Y))␣.GT.␣THRESH)␣THEN
!␣␣␣␣␣␣␣␣␣␣␣␣␣Calculate␣Jacobi␣transformation
H␣=␣W(Y)␣-␣W(X)
IF␣(␣ABS(H)␣+␣G␣.EQ.␣ABS(H)␣)␣THEN
T␣=␣A(X,␣Y)␣/␣H
ELSE
THETA␣=␣0.5D0␣*␣H␣/␣A(X,␣Y)
IF␣(THETA␣.LT.␣0.0D0)␣THEN
T␣=␣-1.0D0␣/␣(SQRT(1.0D0␣+␣THETA**2)␣-␣THETA)
ELSE
T␣=␣1.0D0␣/␣(SQRT(1.0D0␣+␣THETA**2)␣+␣THETA)
END␣IF
END␣IF

C␣=␣1.0D0␣/␣SQRT(␣1.0D0␣+␣T**2␣)
S␣=␣T␣*␣C
Z␣=␣T␣*␣A(X,␣Y)

!␣␣␣␣␣␣␣␣␣␣␣␣␣␣Apply␣Jacobi␣transformation
A(X,␣Y)␣=␣0.0D0
W(X)␣␣␣␣=␣W(X)␣-␣Z
W(Y)␣␣␣␣=␣W(Y)␣+␣Z
```

```
DO␣70␣R␣=␣1,␣X-1
T␣␣␣␣␣␣␣=␣A(R,␣X)
A(R,␣X)␣=␣C␣*␣T␣-␣S␣*␣A(R,␣Y)
A(R,␣Y)␣=␣S␣*␣T␣+␣C␣*␣A(R,␣Y)
70␣␣␣␣␣␣␣␣␣CONTINUE
DO␣80,␣R␣=␣X+1,␣Y-1
T␣␣␣␣␣␣␣=␣A(X,␣R)
A(X,␣R)␣=␣C␣*␣T␣-␣S␣*␣A(R,␣Y)
A(R,␣Y)␣=␣S␣*␣T␣+␣C␣*␣A(R,␣Y)
80␣␣␣␣␣␣␣␣␣CONTINUE
DO␣90,␣R␣=␣Y+1,␣N
T␣␣␣␣␣␣␣=␣A(X,␣R)
A(X,␣R)␣=␣C␣*␣T␣-␣S␣*␣A(Y,␣R)
A(Y,␣R)␣=␣S␣*␣T␣+␣C␣*␣A(Y,␣R)
90␣␣␣␣␣␣␣␣␣CONTINUE

!␣␣␣␣␣␣␣␣␣␣␣␣␣␣Update␣eigenvectors
!---␣This␣loop␣can␣be␣omitted␣if␣only␣the␣eigenvalues␣are␣desired␣---
DO␣100,␣R␣=␣1,␣N
T␣␣␣␣␣␣␣=␣Q(R,␣X)
Q(R,␣X)␣=␣C␣*␣T␣-␣S␣*␣Q(R,␣Y)
Q(R,␣Y)␣=␣S␣*␣T␣+␣C␣*␣Q(R,␣Y)
100␣␣␣␣␣␣␣␣␣CONTINUE
END␣IF
61␣␣␣␣␣CONTINUE
60␣␣␣CONTINUE
40␣CONTINUE

PRINT␣*,␣"eigsym66:␣No␣convergence."

END␣SUBROUTINE
!␣End␣of␣subroutine␣eigsym33
!end␣code
```

A.2 Proposed muscle material model in C++

The C++ code below implements our muscle model that exhibits force enhancement behaviour. This model was designed to be used as a plugin for FEBio, an open-source finite element analysis software tailored for biomechanics and biomedical applications.

Listing 1: Header (.h) file

```
#pragma once
//#include "stdafx.h"
#include "FEBioMech/FEUncoupledMaterial.h"
#include <FECore/FEModelParam.h>
#include "FEBioMech/FEElasticMaterial.h"
#include <FECore/FEMaterialPoint.h>


//-----------------------------------------------------------------------------
//! Muscle Material Point Data
//! Material point data for visco-elastic materials
class FEForceVelPointData : public FEMaterialPointData
{

        public:
        FEForceVelPointData(FEMaterialPointData* mp = nullptr);
        FEMaterialPointData* Copy();
        void Init();
        void Update(const FETimeInfo& timeInfo);
        void Serialize(DumpStream& ar);

        public:
        double  m_lambdap; //fiber stretch at the previous time step
        double  m_lambda; //fiber stretch at the current time step
        double  m_lambdapp; //fiber stretch at the previous time step
        //-  to use in the elasticity tensor

        int  lambda_act; //test
};

//-----------------------------------------------------------------------------
//! This material uses the constitutive model developed by
//! Sampaio de Oliveira and Uchida (2025) to model muscles
//! which undergo active contraction

class Oliveira2024 : public FEUncoupledMaterial
{
        public:
        //! default constructor
        Oliveira2024(FEModel* pfem);

        public:
        // transverse constants
        double m_G1; //!< along-fiber shear modulus
        double m_G2; //!< cross-fiber shear modulus
        double m_G3; //!< new term

        // along fiber constants
        double  m_P1; //!< muscle fiber constant P1
        double  m_P2; //!< muscle fiber constant P2

        double  m_Lofl;  //!< optimal sarcomere length
        double  m_smax;  //!< maximum isometric stretch
        double  m_lam1;
        double  m_alpha;        //!< activation parameter
```

```
        FEVec3dValuator* m_fiber;

        public:
        //! Deviatoric Cauchy stress at material point
        virtual mat3ds DevStress(FEMaterialPoint& mp) override;

        //! Deviatoric spatial tangent stiffness at material point
        virtual tens4ds DevTangent(FEMaterialPoint& mp) override;

        // returns a pointer to a new material point object
        FEMaterialPointData* CreateMaterialPointData() override;

        private:
        // declare the parameter list
        DECLARE_FECORE_CLASS();


};
```

Listing 2: Main body (.cpp) file

```
#pragma once
#include "FECore/stdafx.h"
#include "Oliveira2024.h"
#include <iostream>
#include <fstream>
#include <typeinfo>
#include <algorithm>
#include "FEBioMech/FEUncoupledMaterial.h"
#include <FECore/DumpStream.h>
#include <FECore/FEModel.h>
#include <FECore/FECoreKernel.h>
#include <FEBioMech/FEElasticMaterial.h>
#include <FECore/FEMaterialPoint.h>
#include <FECore/FEElement.h>
#include <FECore/FESolidElement.h>
#include <FECore/FESolidDomain.h>
#include <random>

using namespace std;

#ifndef SQR
#define SQR(x) ((x)*(x))
#endif

//-----------------------------------------------------------------------------
FEForceVelPointData::FEForceVelPointData(FEMaterialPointData* mp)
: FEMaterialPointData(mp)
{
        m_lambdap = 1.0;
        m_lambda = 1.0;
        m_lambdapp = 1.0;
}

//-----------------------------------------------------------------------------
//! Create a shallow copy of the material point data
FEMaterialPointData* FEForceVelPointData::Copy()
{
        FEForceVelPointData* pt = new FEForceVelPointData(*this);
        if (m_pNext) pt->m_pNext = m_pNext->Copy();
        return pt;
}

//-----------------------------------------------------------------------------
//! Initializes material point data.
void FEForceVelPointData::Init()
```

```
{
        // intialize data to zero
        m_lambda =1.0;
        m_lambdap = 1.0;
        m_lambdapp = 1.0;
        lambda_act = 0;

        // don't forget to initialize the base class
        FEMaterialPointData::Init();
}

//-----------------------------------------------------------------------
//! Update material point data.
void FEForceVelPointData::Update(const FETimeInfo& timeInfo)
{


        // the elastic stress stored in pt is the Cauchy stress.
        // however, we need to store the 2nd PK stress
        m_lambdapp = m_lambdap;
        m_lambdap = m_lambda;


        // don't forget to call the base class
        FEMaterialPointData::Update(timeInfo);
}

//-----------------------------------------------------------------------
//! Serialize data to the archive
void FEForceVelPointData::Serialize(DumpStream& ar)
{
        FEMaterialPointData::Serialize(ar);
        ar& m_lambda& m_lambdap;

}



//////////////////////////////////////////////////////////////////////
//                              Muscle Material
//////////////////////////////////////////////////////////////////////



//-----------------------------------------------------------------------
//                              Variables
//-----------------------------------------------------------------------

//PolyCoef = Coefiecients of the polynomial used to interpolate
// the passive curve on active stretching.
// We used a 3rd-order polynomial. We have cases for
// length of {1.0,1.1,1.2,1.3,1.4} times the opt fiber length.

//PassiveCoef = Coefiecients of the polynomial used to interpolate
// the passive curve on purely passive stretching.
// We used a 4th-order polynomial.

//lambda_act_indices = Array with the possible fiber stretch at
// the onset of activation values that could be used on the model.
// This array is required to find the index number w.r.t.
// the row of PolyCoef matrix (associate the PolyCoef with m_lam1)

//lambda_act = global variable to store the index from
// lambda_act_indices associated with m_lam1

//dt = time step.

//num = number added to m_lam1 and m_Lofl to perturb the
```

```
// initial fiber length (used exclusively in the paper).

//lambdadot = time derivative of the fiber stretch. Fiber velocity.
// Calculated by the difference between the fiber stretch from the
// previous time step and  the fiber stretch in the current time step.
// The difference is divided by the time step.

//Fp = Normalized passive fiber force-length curve. Now this is
// calculated considering force enhancent. Therefore, for each
// sarcomere length at onset of activation a new passive curve needs
// to be generated. That means, new parameter for the PolyCoef and
// Passive Coef.


//FpActiveStretch = Normalized passive fiber force-length curve
// when the fibers are fully actived in active stretch experiment.
// This curve is the interpolated function using the coefficient
// PolyCoef.

//FpPassiveStretch = Normalized passive fiber force-length curve when
// the fibers are passive in fully passive stretch experiment.
// This curve is the interpolated function using the coefficient
// PassiveCoef.

//FpDlActiveStretch = derivative of FpActiveStretch w.r.t.
// the fiber stretch (lambda=lat).

//FpDlPassiveStretch = derivative of FpPassiveStretch w.r.t.
// the fiber stretch (lambda=lat).

//indexPassive = Flag number to decide in which of the four rows
// of the PolyCoef matrix will be used.

//Fv = Normalized passive fiber force-velocity curve

//FvDl = derivative of Fv w.r.t. the fiber stretch
// (lambda=lat).

//m_kc, m_ke, m_lamdmin, d = parameters to construct Fv.
// Please see Bol and Reese (2008) for details.

//m_Lofl = ratio of the optimal fiber stretch to the initial
//          fiber length

//m_lam1 = ratio of the initial fiber length to the fiber length at the
//          onset of activation

//m_G1 = coefficient related to the shear stress in the
//        longitudinal direction

//m_G2 = coefficient related to the shear stress in the
//        cross-fiber direction

//m_P1 = coefficient to build the passive curve
//        from Blemker et al. (2005)

//m_P2 = coefficient to build the passive curve
//        from Blemker et al. (2005)

//m_smax = Maximum isometric stress

//m_alpha = activation

//m_fiber = fiber direction
```

```
//-------------------------------------------------------------------------
//   Apply random initial fiber length to the simulation, see Figure 9
//-------------------------------------------------------------------------

// Import Table for Random Opt Fiber Stretch -
// Numerical Stability Test/Figure 9 in the paper.
// To do it so, add the FiberRandom matrix below in the line 248 and 496
// after PassiveCoef[1][5] vector.

//double FiberRandom[1][20] =
//{0.03590855282699751,-0.01995903244285782,-0.05785750966489714,
  //0.0993660098151602,-0.06872447406819257,0.008281253378930375,
        //-0.08359613598607188,0.0954571908075626,0.08078840800823997,
        //-0.08170980514007609,-0.03027788117077186,-0.08253536095624343,
        //0.05224446147704023,0.09885574815228054,0.02402762582519457,
        //0.015127585315798053,-0.06192283592409101,0.03747585895812555,
        //       -0.0966358746252487,-0.09732187745130423};

// Add the commands below to lines 297 and 510 to apply a pertubation
// the m_Lofl and m_lam1, creating the new variables m_Lofl2.
// Those values will be used as replacement for the optimal fiber stretch
// and fiber stretch at the onset of activation.

//double m_Lofl2 = m_Lofl + FiberRandom[0][ElementNumber - 1]/1000;

// Alternatively, one could use the commands below to generate the
// perturbation within the material subroutine. The seed of rand is
// based on the element number so that the simulation always yields
// the same results.

//srand(ElementNumber);
//double num = (rand() % 2001) / 10000.0 - 0.1;
//double num = (rand() % 1001) / 10000.0 * pow(-1.0, rand() % 10000);

//-------------------------------------------------------------------------



BEGIN_FECORE_CLASS(Oliveira2024, FEUncoupledMaterial)
ADD_PARAMETER(m_G1, "g1");
ADD_PARAMETER(m_G2, "g2");
ADD_PARAMETER(m_G3, "g3");
ADD_PARAMETER(m_P1, "p1");
ADD_PARAMETER(m_P2, "p2");
ADD_PARAMETER(m_Lofl, "Lofl");
ADD_PARAMETER(m_smax, "smax");
ADD_PARAMETER(m_lam1, "lam_max");
ADD_PARAMETER(m_alpha, "activation");
ADD_PROPERTY(m_fiber, "fiber");

END_FECORE_CLASS();

//constructor
//-----------------------------------------------------------------------------
Oliveira2024::Oliveira2024(FEModel* pfem) : FEUncoupledMaterial(pfem)
{
        m_G1 = 0;
        m_G2 = 0;
        m_G3 = 0;
        m_alpha = 0.0;
        //m_lamdmin = 1.0;

        //m_fiber = vec3d(1, 0, 0);
        m_fiber = nullptr;

}
```

```
//-----------------------------------------------------------------------
FEMaterialPointData* Oliveira2024::CreateMaterialPointData()
{
        FEForceVelPointData* mp =
        new FEForceVelPointData(new FEElasticMaterialPoint);
        return mp;
}

//-----------------------------------------------------------------------
//! Calculates the deviatoric stress at a material point.
//!
mat3ds Oliveira2024::DevStress(FEMaterialPoint& mp)
{
        // Import Tables to generate normalized passive curves during
        // active stretch. Each column represents the coefficient for a
        // 3rd order polynomial,each row represents the polynomial for
        // lambda_act of {1.0,1.01,1.02,1.03,1.04,...,1.49}. Remember from
        // lambda_act greater or equal to 1.5 there is no binding and we
        // have the passive curve only.
        // Figure 4 from the paper is created when {1.0,1.1,1.2,1.3,1.4}.

        double PolyCoef[50][4] = {
        {-0.259184038105527,0.703681396228948,2.001238192630714,
-2.500499762669549},
        {-0.299722944663740,0.924553996486710,1.613618779597475,
-2.297933855830922},
        {-0.287792044062813,0.869944309767671,1.695343613369996,
-2.354052002234926},
        {-0.301637703677545,0.941667050753426,1.577903887768561,
-2.309145829016744},
        {-0.288636359030189,0.880729792259995,1.671597344232738,
-2.372023473565357},
        {-0.301684908785071,0.948099023912879,1.561717316370244,
-2.330752209700193},
        {-0.287466856843851,0.879900143003484,1.669300031732138,
-2.401566177150210},
        {-0.272942122565528,0.809097425158186,1.783021947473220,
-2.476392375124135},
        {-0.283928184283043,0.865309849534609,1.692688409413146,
-2.445286898331594},
        {-0.268014388081024,0.786077645319158,1.822840193169679,
-2.529727277649410},
        {-0.218496512096289,0.557117163201095,2.007964229437606,
-2.536207533879683},
        {-0.211717919494802,0.533352910857095,2.013916750069883,
-2.533389444465056},
        {-0.192965370172349,0.440449685733639,2.149744740636416,
-2.609208395525165},
        {-0.194812706596567,0.451787031719839,2.114829869285584,
-2.594086415787725},
        {-0.175482972414697,0.354602167842464,2.260267707987545,
-2.675539897670201},
        {-0.156444690107198,0.257945893599023,2.406622068771278,
-2.758044375724219},
        {-0.155986245109803,0.256336055655708,2.395346544168545,
-2.755372467818098},
        {-0.136486136542756,0.155998427163504,2.550360730103629,
-2.843100671639595},
        {-0.134488492226124,0.145660633655833,2.555160859940637,
-2.849032032628492},
        {-0.114627972449354,0.042186148032827,2.717922006408235,
-2.941556612742558},
        {-0.108583309650224,0.018383150938835,2.730164191016034,
-2.940606842788512},
        {-0.090987402364384,-0.082856806111889,2.908268740736746,
-3.053027622894192},
        {-0.071475151723029,-0.186500868406846,3.075191291174807,
```

```
-3.148930900216778},
        {-0.065738562322237,-0.218151737606727,3.119684931515295,
-3.176722164967998},
        {-0.046282589058727,-0.322539532808972,3.289967743130732,
-3.275036516556993},
        {-0.039079993432998,-0.362460257061558,3.349934142363320,
-3.311527210270215},
        {-0.019893267799311,-0.466456408811898,3.521450093111789,
-3.411054912102409},
        {-0.001504145311328,-0.566691047002027,3.687586519966823,
-3.507944815865739},
        {0.007431247248210,-0.616726265013736,3.766798837857173,
-3.555466855240479},
        {0.025266999999659,-0.714740142807066,3.930605117095693,
-3.651460233974625},
        {0.033046903865464,-0.757573819204728,3.995424926774845,
-3.689946741588884},
        {0.052507394406177,-0.866381651700376,4.182224332630489,
-3.800375894764199},
        {0.068571373943386,-0.955822847908797,4.333003410034785,
-3.889512930619338},
        {0.079952107587404,-1.020041272573196,4.439428925264664,
-3.952976102177012},
        {0.095000977753228,-1.104407750661090,4.582176990553272,
-4.037718670888377},
        {0.107336283332130,-1.174135524480674,4.699156461506470,
-4.107478418917459},
        {0.121194170308346,-1.252335322505787,4.831668953367133,
-4.186441915717637},
        {0.133901416630662,-1.324349019275778,4.953371989526763,
-4.259111312432275},
        {0.146885370341677,-1.397993847813952,5.078297432418784,
-4.333753111897394},
        {0.158181214505704,-1.462373632780365,5.186617142050983,
-4.398542702721023},
        {0.168234149644738,-1.519816957167587,5.282294634129469,
-4.455701657952584},
        {0.181504943627837,-1.595296750338977,5.411226969867109,
-4.532874225085311},
        {0.189998042181454,-1.644038813529573,5.491076702540638,
-4.580412036983768},
        {0.202178958496073,-1.712721456762457,5.606310469031770,
-4.648830284597740},
        {0.209294574053946,-1.753769028889718,5.672119074163654,
-4.687713354024468},
        {0.215643523657076,-1.790523855853236,5.730068618265062,
-4.721691016248288},
        {0.228559644320757,-1.863924550258560,5.855267846785148,
-4.796429428471955},
        {0.231974852591925,-1.883222520736604,5.879138412663034,
-4.808723497617694},
        {0.244311993559551,-1.953002194254770,5.997014619113204,
-4.878638591733435},
        {0.245419303645167,-1.958543548075553,5.993921740518868,
-4.873517737982032}};

        // Vector that matches the lambda act with the
        //  coeffients of the polynomial above
        double lambda_act_indices[50] = {1.00,1.01,1.02,1.03,1.04,1.05,
                1.06,1.07,1.08,1.09,1.10,1.11,1.12,1.13,1.14,1.15,
                1.16,1.17,1.18,1.19,1.20,1.21,1.22,1.23,1.24,1.25,1.26,
                1.27,1.28,1.29,1.30,1.31,1.32,1.33,1.34,1.35,1.36,1.37,
                1.38,1.39,1.40,1.41,1.42,1.43,1.44,1.45,1.46,1.47,1.48,
                1.49 };

        // Each column represents the coefficient of a 4th-order
        // polynomial representing the fully passive normalized curve.
        // Normalized Fig3a plot from the paper.
```

```
double PassiveCoef[1][5] = {0.819290307034414,-5.53138441221523,
        13.2261642018205,-12.5063835706935,3.98678238655361};

// Lines of code to get the element ID
//(maybe for future application)
FEElement* pe = dynamic_cast<FEElement*>(mp.m_elem);
int ElementNumber = pe->GetID();

// Pointer to get the data from the material point
// (maybe for future application)
FEElasticMaterialPoint& pt =
*mp.ExtractData<FEElasticMaterialPoint>();

double dt = GetFEModel()->GetTime().timeIncrement;

FEForceVelPointData& vpt =
*mp.ExtractData<FEForceVelPointData>();

// Deformation gradient
mat3d& F = pt.m_F;
double J = pt.m_J;

// Get the local coordinate systems
mat3d Q = GetLocalCS(mp);

// Get the initial fiber direction
vec3d a0 = Q * m_fiber->unitVector(mp);

// Calculate the current material axis lam*a = F*a0;
vec3d a = F * a0;

// Normalize material axis and store fiber stretch
double la = a.unit();
double lat = la * pow(J, -1.0 / 3.0);
// i.e. lambda tilde = sqrt(I4)

// Current fiber stretch
vpt.m_lambda = lat;

// Velocity fiber stretch
// (currently not used -
//  maybe for future application)
double lambdadot = (lat - vpt.m_lambdap) / dt;

// Calculate deviatoric left Cauchy-Green tensor: B = Jm23*F*Ft
mat3ds B = pt.DevLeftCauchyGreen();

// Calculate square of B
mat3ds B2 = B.sqr();

// Calculate Ba = B*a
vec3d Ba = B * a;

// ----- Invariants of B -----
double I1 = B.tr();
double I2 = 0.5 * (I1 * I1 - B2.tr());
double I4 = lat * lat;
double I5 = I4 * (a * Ba);

// Applying the randomness of FiberRandom -
// not used in the normal application
double num = 0.0;
double m_Lofl2 = m_Lofl + num;
double m_lam12 = m_lam1;

// Finding the index used in the PolyCoef and PassiveCoef matrix
int i = 0;
if (vpt.lambda_act == 0) {
```

```
            while ((lambda_act_indices[i] - m_lam12 / m_Lofl2) < 0.0)
            {
                    i = i + 1;
            }
            vpt.lambda_act = i;
        }


        // ----- Calculate new invariants b1 and b2 of B ------

        // note that we need to be careful about roundoff errors
        // since these functions may create numerical problems

        double g = I5 / (I4 * I4) - 1;
        double b1 = (g > 0 ? sqrt(g) : 0);

        double b2 = acosh(0.5 * (I1 * I4 - I5) / lat);

        // Calculate omega (w)
        double I4r = sqrt(I4);
        double w = 0.5 * (I1 * I4 - I5) / sqrt(I4);

        // Set beta and ksi to their limit values
        double beta = 1.0;
        double ksi = -1.0 / 3.0;

        // if w does not equal unity, we can calculate beta and ksi
        if (w > 1.0001)
        {
            beta = b2 / sqrt(w * w - 1);
            ksi = (1.0 / (w * w - 1))
            * (1 - w * b2 / sqrt(w * w - 1));
        }

        // ----- Calculate strain derivatives -----

        // We assumed that W(I1, I4, I5, alpha) = F1(B1(I4, I5))
        // + F2(B2(I1,I4,I5)) + F3(lam(I4), alpha)
        double W1, W2, W4, W5;

        // Calculate derivatives for F1
        double F1D4 = -2 * m_G1 * (I5 / (I4 * I4 * I4));
        double F1D5 = m_G1 / (I4 * I4);

        // Calculate derivatives for F2
        double F2D1 = m_G2 * beta * lat;
        double F2D4 = m_G2 * beta * (I1 * I4 + I5) * 0.5 * pow(I4, -1.5);
        double F2D5 = -m_G2 * beta / lat;

        // Calculate derivatives for F3
        // these terms are proposed to fix the zero-stress problem
        double F3D4 = 9.0 * m_G3 * 0.125 * log(I4) / I4;

        // F4p - calculate passive fiber force
        double Fp; double xInt;
        double FpActiveStretch;
        double FpPassiveStretch;
        int indexPassive = 0;

//-----------------------------------------------------------------------
        // Calculate the Passive Stress from the preconstructed data
        // with force enhancment.
        // Data for initial stretch at the length of {1.0,1.1,1.2,1.3,1.4}
        // times the optimal fiber length

        // x-value for the interpolated polynomial in the passive curve
        xInt = lat / m_Lofl2; //that transforms the case in which m_Loft>1
        // to the case that the passive F-L curve
```

```
        //  starts at  the optimal fiber length.

        if (lat <= m_Lofl2)
        {
                Fp = 0;
        }
        else
        {
                // Case in which the activation starts
                // on the ascending limb of the active curve
                indexPassive = 0;
                if ((m_lam12 / m_Lofl2 - 1.0) <= 0.0) {
                }
                else{
                        indexPassive = vpt.lambda_act;
                }

                // Polynomial for the fully activated case
                // (Eq 17, f_passive^alpha=1)
                FpActiveStretch = PolyCoef[indexPassive][0] *
                xInt * xInt * xInt
                + PolyCoef[indexPassive][1] * xInt * xInt
                + PolyCoef[indexPassive][2] * xInt
                + PolyCoef[indexPassive][3];


                // Polynomial for the fully passive case
                //  (Eq 17, f_passive^alpha=0)
                FpPassiveStretch = PassiveCoef[0][0]
                * pow(lat / m_Lofl2, 4)
                + PassiveCoef[0][1] * pow(lat / m_Lofl2, 3)
                + PassiveCoef[0][2] * pow(lat / m_Lofl2, 2)
                + PassiveCoef[0][3] * pow(lat / m_Lofl2, 1)
                + PassiveCoef[0][4];

//-----------------------------------------------------------------------
                // Steps to make sure that the force functions are positive
                if (FpPassiveStretch < 0) { FpPassiveStretch = 0; };

                if (FpActiveStretch < 0) { FpActiveStretch = 0; };

                // Interpolation of the fully passive and active stretch
                // for submaximal contraction (Eq 17)
                Fp = FpPassiveStretch * (1 - m_alpha)
                + FpActiveStretch * (m_alpha);

                //If the fiber length at the onset of activation is greater
                // than 1.5 times the opt fiber length, there is
                // no binding and only the FpPassiveStretch occurs
                if (m_lam12 > 1.5 * m_Lofl2) { Fp = FpPassiveStretch; };

        }


        // F4a - calculate active fiber force
        double Fa = 0;

        // Translate the active force according to the initial stretch
        double m_lmin = 0.51 + (m_Lofl2 - 1.0);

        if ((lat <=m_lmin) )
        {
                // We have added this part to make sure that
                // Fa is zero for values below the min.
                // Eq (18), "otherwise" part
                Fa = 0;
        }
        else
```

```
        {
                // Eq (18), lambda > lambda_min part
                Fa = (lat - m_lmin)/(m_Lofl2 - m_lmin)
                *exp((2 * m_lmin - lat - m_Lofl2)
                *(lat - m_Lofl2)/(2*pow((m_lmin - m_Lofl2),2)));
        }

        //F4v - Force-velocity fiber force
        // (not implemented, possible future work)
        double Fv = 1.0;
        double m_kc = 5.0;
        double m_ke = 5.0;
        double m_lamdmin = -17.0;
        double d = 1.5;

        if (lambdadot <= 0.0)
        {
                Fv = (1-lambdadot/m_lamdmin)
                /(1 +m_kc * lambdadot / m_lamdmin);
        }
        else
        {
                Fv = d - (d-1) * (1 + lambdadot / m_lamdmin)
                / (1 - m_kc * m_ke * lambdadot / m_lamdmin);
        }


        // quasi-static analysis, not considering the F-V effect
        Fv = 1.0;

        // calculate total fiber force
        double FfD1 = m_smax * (Fp + m_alpha * Fa * Fv) / m_Lofl2;
        double FfD4 = 0.5 * FfD1 / lat;

        double FfD1a = m_smax * (m_alpha * Fa * Fv) / m_Lofl2;
        double FfD4a = 0.5 * FfD1a / lat;

        double FfD1p = m_smax * (Fp) / m_Lofl2;
        double FfD4p = 0.5 * FfD1p / lat;


        // Add all derivatives
        W1 = F2D1;
        W2 = 0;
        W4 = F1D4 + F2D4 + F3D4 + FfD4;
        W5 = F1D5 + F2D5;

        // Dyadic of a
        mat3ds AxA = dyad(a);

        // Symmetric dyad of a and Ba
        mat3ds ABA = dyads(a, Ba);

        // ----- calculate stress -----
        // Calculate T
        mat3ds T = B * (W1 + W2 * I1) - B2 * W2
        + AxA * (I4 * W4) + ABA * (I4 * W5);
        return T.dev() * (2.0 / J);


}


//-----------------------------------------------------------------------
//! Calculates the spatial deviatoric tangent at a material point
//!
tens4ds Oliveira2024::DevTangent(FEMaterialPoint& mp)
{
```

```
        FEElasticMaterialPoint& pt =
        *mp.ExtractData<FEElasticMaterialPoint>();

        // Import Tables to generate normalized passive curves
        // during active stretch. Each column represents the coefficient
        // for a 3rd-order polynomial, each row representsthe polynomial
        // for lambda_act of {1.0,1.01,1.02,1.03,1.04,...,1.49}.
        // Remember from lambda_act greater or equal to 1.5 there is
        // no binding and wehave the passive curve only. Figure 4
        // from the paper is created when {1.0,1.1,1.2,1.3,1.4}.

        double PolyCoef[50][4] = {
        {-0.259184038105527,0.703681396228948,2.001238192630714,
-2.500499762669549},
        {-0.299722944663740,0.924553996486710,1.613618779597475,
-2.297933855830922},
        {-0.287792044062813,0.869944309767671,1.695343613369996,
-2.354052002234926},
        {-0.301637703677545,0.941667050753426,1.577903887768561,
-2.309145829016744},
        {-0.288636359030189,0.880729792259995,1.671597344232738,
-2.372023473565357},
        {-0.301684908785071,0.948099023912879,1.561717316370244,
-2.330752209700193},
        {-0.287466856843851,0.879900143003484,1.669300031732138,
-2.401566177150210},
        {-0.272942122565528,0.809097425158186,1.783021947473220,
-2.476392375124135},
        {-0.283928184283043,0.865309849534609,1.692688409413146,
-2.445286898331594},
        {-0.268014388081024,0.786077645319158,1.822840193169679,
-2.529727277649410},
        {-0.218496512096289,0.557117163201095,2.007964229437606,
-2.536207533879683},
        {-0.211717919494802,0.533352910857095,2.013916750069883,
-2.533389444465056},
        {-0.192965370172349,0.440449685733639,2.149744740636416,
-2.609208395525165},
        {-0.194812706596567,0.451787031719839,2.114829869285584,
-2.594086415787725},
        {-0.175482972414697,0.354602167842464,2.260267707987545,
-2.675539897670201},
        {-0.156444690107198,0.257945893599023,2.406622068771278,
-2.758044375724219},
        {-0.155986245109803,0.256336055655708,2.395346544168545,
-2.755372467818098},
        {-0.136486136542756,0.155998427163504,2.550360730103629,
-2.843100671639595},
        {-0.134488492226124,0.145660633655833,2.555160859940637,
-2.849032032628492},
        {-0.114627972449354,0.042186148032827,2.717922006408235,
-2.941556612742558},
        {-0.108583309650224,0.018383150938835,2.730164191016034,
-2.940606842788512},
        {-0.090987402364384,-0.082856806111889,2.908268740736746,
-3.053027622894192},
        {-0.071475151723029,-0.186500868406846,3.075191291174807,
-3.148930900216778},
        {-0.065738562322237,-0.218151737606727,3.119684931515295,
-3.176722164967998},
        {-0.046282589058727,-0.322539532808972,3.289967743130732,
-3.275036516556993},
        {-0.039097993432998,-0.362460257061558,3.349934142363320,
-3.311527210270215},
        {-0.019893267799311,-0.466456408811898,3.521450093111789,
-3.411054912102409},
        {-0.001504145311328,-0.566691047002027,3.687586519966823,
-3.507944815865739},
```

```
        {0.007431247248210,-0.616726265013736,3.766798837857173,
-3.555466855240479},
        {0.025266999999659,-0.714740142807066,3.930605117095693,
-3.651460233974625},
        {0.033046903865464,-0.757573819204728,3.995424926774845,
-3.689946741588884},
        {0.052507394406177,-0.866381651700376,4.182224332630489,
-3.800375894764199},
        {0.068571373943386,-0.955822847908797,4.333003410034785,
-3.889512930619338},
        {0.079952107587404,-1.020041272573196,4.439428925264664,
-3.952976102177012},
        {0.095000977753228,-1.104407750661090,4.582176990553272,
-4.037718670888377},
        {0.107336283332130,-1.174135524480674,4.699156461506470,
-4.107478418917459},
        {0.121194170308346,-1.252335322505787,4.831668953367133,
-4.186441915717637},
        {0.133901416630662,-1.324349019275778,4.953371989526763,
-4.259111312432275},
        {0.146885370341677,-1.397993847813952,5.078297432418784,
-4.333753111897394},
        {0.158181214505704,-1.462373632780365,5.186617142050983,
-4.398542702721023},
        {0.168234149644738,-1.519816957167587,5.282294634129469,
-4.455701657952584},
        {0.181504943627837,-1.595296750338977,5.411226969867109,
-4.532874225085311},
        {0.189998042181454,-1.644038813529573,5.491076702540638,
-4.580412036983768},
        {0.202178958496073,-1.712721456762457,5.606310469031770,
-4.648830284597740},
        {0.209294574053946,-1.753769028889718,5.672119074163654,
-4.687713354024468},
        {0.215643523657076,-1.790523855853236,5.730068618265062,
-4.721691016248288},
        {0.228559644320757,-1.863924550258560,5.855267846785148,
-4.796429428471955},
        {0.231974852591925,-1.883222520736604,5.879138412663034,
-4.808723497617694},
        {0.244311993559551,-1.953002194254770,5.997014619113204,
-4.878638591733435},
        {0.245419303645167,-1.958543548075553,5.993921740518868,
-4.873517737982032}};

        // Vector that matches the lambda act with the coeffients of the
        // polynomial above
        double lambda_act_indices[50] =
        {1.00,1.01,1.02,1.03,1.04,1.05,1.06,
                1.07,1.08,1.09,1.10,1.11,1.12,1.13,1.14,1.15,
                1.16,1.17,1.18,1.19,1.20,1.21,1.22,1.23,1.24,
                1.25,1.26,1.27,1.28,1.29,1.30,1.31,1.32,1.33,
                1.34,1.35,1.36,1.37,1.38,1.39,1.40,1.41,1.42,
                1.43,1.44,1.45,1.46,1.47,1.48,1.49};

        // Each column represents the coefficient of a 4th-order polynomial
        // representing the fully passive normalized curve.
        // Normalized Fig3a plot from the paper.
        double PassiveCoef[1][5] = { 0.819290307034414,-5.53138441221523,
                13.2261642018205,-12.5063835706935,3.98678238655361};

        double dt = GetFEModel()->GetTime().timeIncrement;

        FEForceVelPointData& vpt = *mp.ExtractData<FEForceVelPointData>();

        // deformation gradient
        mat3d& F = pt.m_F;
        double J = pt.m_J;
```

```
// Lines of code to get the element ID
// (maybe for future application)
FEElement* pe = dynamic_cast<FEElement*>(mp.m_elem);
int ElementNumber = pe->GetID();

// Applying the randomness of FiberRandom -
// not used in the normal application
double num = 0.0;
double m_Lofl2 = m_Lofl + num;
double m_lam12 = m_lam1;

// deviatoric cauchy-stress, trs = trace[s]/3
mat3ds devs = pt.m_s.dev();

// get the local coordinate systems
mat3d Q = GetLocalCS(mp);

// get the initial fiber direction
vec3d a0 = Q * m_fiber->unitVector(mp);

// calculate the current material axis lam*a = F*a0
vec3d a = F * a0;

// normalize material axis and store fiber stretch
double la = a.unit();
double lat = la * pow(J, -1.0 / 3.0); // i.e. lambda tilde

// velocity fiber stretch
double lambdadot = (lat - vpt.m_lambdapp) / dt;



// ----- Calculate new invariants b1 and b2 of B ------
// get deviatoric left Cauchy-Green tensor
mat3ds B = pt.DevLeftCauchyGreen();

// calculate square of B
mat3ds B2 = B.sqr();

// calculate B*a
vec3d Ba = B * a;

// invariants of B
double I1, I2, I4, I5;

I1 = B.tr();

I2 = 0.5 * (I1 * I1 - B2.tr());

I4 = lat * lat;

I5 = I4 * (a * Ba);


// calculate new invariants
double g = I5 / (I4 * I4) - 1;
double b1 = (g > 0 ? sqrt(g) : 0);

double b2 = acosh(0.5 * (I1 * I4 - I5) / lat);

// calculate omega (w)
double w = 0.5 * (I1 * I4 - I5) / lat;

// set beta and ksi to their limit values
double beta = 1.0;
double ksi = -1.0 / 3.0;
```

```
        // if w does not equal unity, we can calculate beta and ksi
        if (w > 1.0001)
        {
                beta = b2 / sqrt(w * w - 1);
                ksi = (1.0 / (w * w - 1)) * (1 - w * b2 / sqrt(w * w - 1));
        }

        // --- strain energy derivatives ---
        // We assume that W(I1, I4, I5, alpha) =
        // F1(B1(I4, I5)) + F2(B2(I1,I4,I5)) + F3(lam(I4), alpha)
        double W1, W2, W4, W5;

        // -- A. matrix contribution --
        // calculate derivatives for F1
        double F1D4 = -2 * m_G1 * (I5 / (I4 * I4 * I4));

        double F1D5 = m_G1 / (I4 * I4);

        double F1D44 = 6 * m_G1 * (I5 / (I4 * I4 * I4 * I4));

        double F1D45 = -2 * m_G1 / (I4 * I4 * I4);


        // calculate derivatives for F2
        double F2D1 = m_G2 * beta * lat;

        double F2D4 = m_G2 * beta * (I1 * I4 + I5) * 0.5 * pow(I4, -1.5);

        double F2D5 = -m_G2 * beta / lat;

        double F2D11 = ksi * m_G2 * I4 * 0.5;

        double F2D44 = 2.0 * m_G2 * ksi * pow(0.25 * (I1 * I4 + I5)
        / pow(I4, 1.5), 2) -
        m_G2 * beta * (0.25 * (I1 * I4 + 3 * I5) / pow(I4, 2.5));

        double F2D55 = 0.5 * m_G2 * ksi / I4;

        double F2D14 = m_G2 * beta * 0.5 / lat + m_G2 * ksi
        * (I1 * I4 + I5) * 0.25 / I4;

        double F2D15 = -0.5 * m_G2 * ksi;

        double F2D45 = m_G2 * beta * 0.5 * pow(I4, -1.5) -
        m_G2 * ksi * 0.25 * (I1 * I4 + I5) / (I4 * I4);


        // calculate derivatives for F3
        // these terms are proposed to fix the zero-stress problem
        double F3D4 = 9.0 * m_G3 * 0.125 * log(I4) / I4;
        double F3D44 = 9.0 * m_G3 * 0.125 * (1 - log(I4)) / (I4 * I4);

//-----------------------------------------------------------------------
        // -- B. fiber contribution --

        // calculate passive fiber force
        double Fp, FpDl;
        double xInt;
        double FpActiveStretch, FpPassiveStretch;
        double FpDlActiveStretch, FpDlPassiveStretch;
        if (lat <= m_Lofl2)
        {
                Fp = 0;
                FpDl = 0;
        }
        else
        {
                // Calculate the passive stress from the preconstructed
```

```
// data with force enhancement.
// Data for initial stretch at the length of
// {1.0,1.1,1.2,1.3,1.4} times the optimal fiber length
int indexPassive=0;

//that transforms the case in which m_Loft>1 to
// the case that the passive F-L curve starts at
//  the optimal fiber length.
xInt = lat / m_Lofl2;

// Case in which the activation starts
// on the ascending limb of the active curve
if ((m_lam12 / m_Lofl2 - 1.0) <= 0.0) {
        indexPassive = 0;
}
else {
        indexPassive = vpt.lambda_act;
}

// Polynomial for the fully activated case (Eq 17)
FpActiveStretch = PolyCoef[indexPassive][0]
* xInt * xInt * xInt
+ PolyCoef[indexPassive][1] * xInt * xInt
+ PolyCoef[indexPassive][2] * xInt
+ PolyCoef[indexPassive][3];

// Derivative w.r.t. lambda for the fully activated case
FpDlActiveStretch = (3.0 * PolyCoef[indexPassive][0]
* xInt * xInt
+ 2.0 * PolyCoef[indexPassive][1] * xInt
+ 1.0 * PolyCoef[indexPassive][2]);

// Polynomial for the fully passive case (Eq 17)
FpPassiveStretch = PassiveCoef[0][0]
* pow(lat / m_Lofl2, 4)
+ PassiveCoef[0][1] * pow(lat / m_Lofl2, 3)
+ PassiveCoef[0][2] * pow(lat / m_Lofl2, 2)
+ PassiveCoef[0][3] * pow(lat / m_Lofl2, 1)
+ PassiveCoef[0][4];

// Derivative w.r.t. lambda for the fully passsive case
FpDlPassiveStretch = 4 * PassiveCoef[0][0]
* pow(lat / m_Lofl2, 3)
+ 3 * PassiveCoef[0][1] * pow(lat / m_Lofl2, 2)
+ 2 * PassiveCoef[0][2] * pow(lat / m_Lofl2, 1)
+ PassiveCoef[0][3];

// Steps to make sure that the force functions are positive
if (FpPassiveStretch < 0) { FpPassiveStretch = 0; };

if (FpActiveStretch < 0) { FpActiveStretch = 0; };

// Interpolation of the fully passive and active stretch
// for submaximal contraction (Eq 17)
Fp = FpPassiveStretch * (1 - m_alpha)
+ FpActiveStretch * (m_alpha);

// Interpolation of the derivatives of fully passive and
// active stretch for submaximal contraction
FpDl = FpDlPassiveStretch * (1 - m_alpha)
+ FpDlActiveStretch * (m_alpha);


// If the fiber length at the onset of activation is
// greater than 1.5 times the opt fiber length,
// there is no binding and only the FpPassiveStretch
// occurs
if (m_lam12 > 1.5 * m_Lofl2) {
```

```
                Fp = FpPassiveStretch;

                FpDl = FpDlPassiveStretch;
            };

        }
//-------------------------------------------------------------------------
        // calculate active fiber force
        double Fa = 0, FaDl = 0;
        // Translate the active force according to the initial stretch
        double m_lmin = 0.51+ (m_Lofl2-1.0);

        if ((lat <= 0.4 * m_Lofl2) || (lat >= 1.6 * m_Lofl2))
        {
            // We have added this part to make sure that
            // Fa is zero for values below the min.
            // Eq (18), "otherwise" part

            Fa = 0;
            FaDl = 0;
        }
        else
        {
            // Eq (18), lambda > lambda_min part
            Fa = (lat - m_lmin) / (m_Lofl2 - m_lmin)
            * exp((2 * m_lmin - lat - m_Lofl2)
            * (lat - m_Lofl2) / (2 * pow((m_lmin - m_Lofl2), 2)));

            FaDl = (exp (((2* m_lmin - m_Lofl2 - lat)*(-m_lmin + lat))
            /(2*pow((-m_lmin + m_Lofl2),2)))
            *(2* m_lmin - m_Lofl2 - lat)*
            (m_Lofl2 - lat)) / pow((m_lmin - m_Lofl2), 3);
        }

        //F4v - Force-velocity fiber force
        //(not implemented, possible future work)
        double Fv = 1.0;
        double FvDl = 0.0;
        double m_kc = 5.0;
        double m_ke = 5.0;
        double m_lamdmin = -17.0;
        double d = 1.5;

        if (lambdadot <= 0.0)
        {
            Fv = (1 - lambdadot / m_lamdmin)
            /(1 + m_kc * lambdadot / m_lamdmin);
            FvDl= -(1 + m_kc)
            /(SQR(1 + m_kc * lambdadot / m_lamdmin)*m_lamdmin*dt);
        }
        else
        {
            Fv = d - (d - 1) * (1 + lambdadot / m_lamdmin)
            / (1 - m_kc * m_ke * lambdadot / m_lamdmin);
            FvDl = -(d - 1) * (1 + m_ke*m_kc) /
            (SQR(1 - m_ke*m_kc * lambdadot /
            m_lamdmin) * m_lamdmin * dt);
        }

        double teste = abs(vpt.m_lambda - vpt.m_lambdapp);
        if (teste <= 1.0E-6)
        {
            Fv = 1.0;
            FvDl = 0.0;
        }

        //quasi-static analysis, not considering the F-V effect
        Fv = 1.0;
```

```
        FvDl = 0.0;

        // calculate total fiber force
        double FfDl = m_smax * (Fp + m_alpha * Fa * Fv) / m_Lofl2;
        double FfD4 = 0.5 * FfDl / lat;

        double FfDll = m_smax * (FpDl + m_alpha * FaDl
        * Fv+ m_alpha * Fa * FvDl) / m_Lofl2;
        double FfD44 = 0.25 * (FfDll - FfDl / lat) / I4;

        // add all derivatives
        W1 = F2D1;
        W2 = 0;
        W4 = F1D4 + F2D4 + F3D4 + FfD4;
        W5 = F1D5 + F2D5;

        // calculate second derivatives
        double W11, W12, W22, W14, W24, W15, W25, W44, W45, W55;

        W11 = F2D11;
        W12 = 0;
        W22 = 0;
        W14 = F2D14;
        W24 = 0;
        W15 = F2D15;
        W25 = 0;
        W44 = F1D44 + F2D44 + F3D44 + FfD44;
        W45 = F1D45 + F2D45;
        W55 = F2D55;

        // calculate dWdC:C
        double WCC = W1 * I1 + 2 * W2 * I2 + W4 * I4 + 2 * W5 * I5;

        // calculate C:d2WdCdC:C
        double CW2CCC = (W11 * I1 + W12 * I1 * I1 + W2 * I1 + 2 * W12 * I2
        + 2 * W22 * I1 * I2 + W14 * I4 + W24 * I1 * I4 + 2 * W15 * I5
        + 2 * W25 * I1 * I5) * I1
        - (W12 * I1 + 2 * W22 * I2 + W2 + W24 * I4 + 2 * W25 * I5)
        * (I1 * I1 - 2 * I2)
        + (W14 * I1 + 2 * W24 * I2 + W44 * I4 + 2 * W45 * I5) * I4
        + (W15 * I1 + 2 * W25 * I2 + W45 * I4 + 2 * W55 * I5) * 2 * I5
        + 2 * W5 * I5;

        // second-order identity tensor
        mat3dd ID(1);

        // 4th-order identity tensors
        tens4ds IxI = dyad1s(ID);
        tens4ds I = dyad4s(ID);

        // dyad of a
        mat3ds AxA = dyad(a);

        // --- calculate elasticity tensor ---

        // calculate push-forward of dI5/dC = I4*(a*Ba + Ba*a)
        mat3ds I5C = dyads(a, Ba) * I4;

        // calculate push-forward of d2I5/dCdC
        tens4ds I5CC = dyad4s(AxA, B) * I4;

        // calculate push-forward of I
        tens4ds Ib = dyad4s(B);
//-----------------------------------------------------------------------
        // calculate push-forward of dW/dCdC:C
        mat3ds WCCC =
        B * (W11 * I1 + W12 * I1 * I1 + W2 * I1 + 2 * W12 * I2
        + 2 * W22 * I1 * I2 + W14 * I4 + W24 * I1 * I4 + 2 * W15 * I5
```

```
        + 2 * W25 * I1 * I5)
        - B2 * (W12 * I1 + 2 * W22 * I2 + W2 + W24 * I4 + 2 * W25 * I5)
        + AxA * ((W14 * I1 + 2 * W24 * I2 + W44 * I4 + 2 * W45 * I5)
        * I4) + I5C * (W15 * I1 + 2 * W25 * I2 + W45 * I4 + 2 * W55
        * I5 + W5);
        // calculate push-forward of dW2/dCdC
        tens4ds W2CC =
        dyad1s(B) * (W11 + 2.0 * W12 * I1 + W2 + W22 * I1 * I1)
        + dyad1s(B, B2) * (-(W12 + W22 * I1)) + dyad1s(B2) * W22 - Ib
        * W2 + dyad1s(B, AxA) * ((W14 + W24 * I1) * I4)
        + dyad1s(B, I5C) * (W15 + W25 * I1)
        + dyad1s(B2, AxA) * (-W24 * I4)
        + dyad1s(AxA) * (W44 * I4 * I4)
        + dyad1s(AxA, I5C) * (W45 * I4)
        + dyad1s(I5C) * W55 + I5CC * W5;

        // put it all together
        // cw
        tens4ds cw = IxI * ((4.0 / (9.0 * J)) * (CW2CCC)) + W2CC * (4 / J)
        - dyad1s(WCCC, ID) * (4.0 / (3.0 * J));

        // elasticity tensor
        tens4ds c = dyad1s(devs, ID) * (-2.0 / 3.0) + (I - IxI / 3.0)
        * (4.0 * WCC / (3.0 * J)) + cw;

        return tens4ds(c);

}
```