The University of Cape Town

Masters Thesis

Material Parameter Identification for Modelling the Left Ventricle in the Healthy State

Author:
Mohammed Asaad Essack

Supervisor:
Dr. Sebastian Skatulla

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science in Engineering

in the

Department of Civil Engineering

May 2014

Declaration of Authorship

I, Mohammed Asaad Essack, declare that this thesis titled, 'Identification of Material Parameters to Model the Left Ventricle in the Healthy State' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed: _____

Date: _____

THE UNIVERSITY OF CAPE TOWN

Abstract

Engineering and the Built Environment
Department of Civil Engineering

Master of Science in Engineering

Material Parameter Identification for Modelling the Left Ventricle in the Healthy State

by Mohammed Asaad Essack

An idealized truncated ellipsoidal model, was used to simulate a healthy canine left ventricle. Passive behaviour of the myocardium was modelled using the constitutive model of Usyk. In addition, active behaviour of the myocardium was modelled by the active stress law of Guccione. Furthermore, the load faced by the left ventricle in ejecting blood into the arterial system, was modelled with the three element Windkessel model of Westerhof. The model was calibrated to pressure-volume data, which was adapted from the work of Kerckhoffs. The projected Levenberg-Marquardt algorithm was used to identify material parameters. Identification of the anisotropic constants in the model of Usyk proved to be difficult, with the calibration algorithm often converging to parameter values that produced numerical instability. The remaining parameters in the Usyk model were successfully identified and produced a good fit of the diastolic filling curve of Kerckhoffs. Calibration of active tension and Windkessel parameters, using pressure-time and volume-time data from the model of Kerckhoffs, proved to be unsuccessful, due to numerical rounding errors in the computation of the Jacobian matrices. However, an alternate approach to this calibration was developed, which compared simulated and reference pressure-volume relations, at three representative points, namely, end-isovolumetric contraction, peak systolic pressure and end-ejection. Using this approach active tension and Windkessel parameters were successfully identified and a good fit of the reference systolic pressure-volume curve was achieved. Thus, passive, active tension and Windkessel parameters were identified, that produced a closed loop pressure-volume curve, even upon further refinement of the particle distribution of the model. For the refined model, left ventricular behaviour in terms of deformation, was consistent with similar ellipsoid model studies. In addition, it was observed that the distribution of the average fibre strain between start and end-ejection throughout the ventricle, was more or less physiologically homogeneous.

Acknowledgements

Firstly, I would like to acknowledge my supervisor Dr. Sebastian Skatulla, for his his continuous support, funding and encouragement over the duration of this thesis. He has assisted me in understanding object orientated programming using C++, which is a language that I was completely unfamiliar with at the beginning of my MSc program. Dr. Skatulla, has also been extremely patient with me as a student, given my limited knowledge in the field of finite element modelling, cardiac mechanics and programming. He has also been extremely helpful in assisting me understand his program (SESKA) and without this input I would not have been able to implement what was required of me for my MSc. thesis.

In addition, I would like to acknowledge my parents, who have funded my MSc. studies. They have also provided me with continuous support and were always available to calm me down when i felt stressed and overwhelmed by my MSc. thesis.

I would also like to acknowledge Ritesh Rama, for his assistance in helping me understand the C++ language, in particular the idea of inheritance. He has also been a very good friend to me over the duration of my study, and was always available for advice and support, when I felt overwhelmed by this research.

Finally, I would like to acknowledge all the other parties, that have contributed to my ability to finish this thesis document, if I could write another document to acknowledge all of them, I would have to submit two large documents instead of just the one.

Contents

List of Figures

List of Tables

Dedicated to my Grandfather, Goolam Mosam, who taught me to never give up when faced with a challenge or seemingly insurmountable obstacle.

Chapter 1

Introduction

1.1 Context of the Research

In this work, the physical system being modelled is the left ventricle and the associated cardiac function, over a single beat of a cardiac cycle. Numerical simulations of cardiac function are conducted using the in-house modelling software SESKA. SESKA is a multi-platform numerical modelling software, based on the element free Galerkin method (EFGM) and uses the moving least squares method (MLS), for computing the shape functions [18]. The cardiac mechanics research that will be presented in this work, forms part of an ongoing effort by the Computational Continuum Mechanics Group (CCM), at the University of Cape Town, to understand and quantify the behaviour of the left ventricle.

1.2 Inverse Methods for Determining Material Parameters

In recent times, due to the exponential improvement in computing power, it has become increasingly possible to model large scale engineering problems, accurately, using numerical techniques such as finite element modelling (FEM) or element free Galerkin methods (EFGM). However, these engineering problems attempt to model real world physical phenomena, by employing certain constitutive models, that attempt to replicate the behaviour of the real physical system. Inherent to the chosen constitutive model are governing material parameters of the system. Previously, an understanding of these parameters was obtained by making several modelling assumptions that simplify the simulated physical system. Material parameters were then estimated using data from experimental testing of the real physical system. However, with the evolution and improvements in computing power, a new more interesting technique has come about for

the purposes of material parameter estimation. These methods, at their core, employ a branch of mathematics known as inverse problems.

Avril et al. [19] summarised various inverse methods for material parameter identification. However, this research will focus on a fairly new field, involving updating methods of inverse modelling [19]. Such methods involve the computation of simulated fields (such as strain or displacement) numerically (such as FEM), using initial estimates for material parameters, defined in the particular constitutive law chosen. The actual constitutive material parameters are identified by minimizing a cost function, which compares available data measurements from experimental testing, to the corresponding results simulated via a numerical analysis. These methods are advantageous as full-field measurements are useful, but not necessarily required. Rather, only a few measurements distributed across the volume of interest are necessary. Recently, Dubuis et al. [20] employed such methods to calibrate material parameters for soft tissues of the human calf, based on a Neo-Hookean constitutive model, using resultant displacements from MRI images, at various cross sectional heights. In terms of cardiac mechanics applications, Sun et al. [21] calibrated the biaxial mechanical response of bioprosthetic heart valve biomaterials, characterized by a Fung type constitutive model, using in-plane shear data from biaxial testing. Furthermore, work by Kural et al. [22] involved identification of passive parameters from a Fung type model, based on planar biaxial testing of diseased human coronary and carotid arteries. In the three mentioned works, a cost function was constructed to compare experimental and simulated data. Furthermore, the mentioned authors utilized a non-linear optimization algorithm in order to minimize constructed cost functions and obtain sought after material parameters.

Optimization algorithms can be split into two broad categories, namely, gradient based algorithms and non-gradient based algorithms [14, 23]. Gradient based algorithms, are usually associated with faster convergence towards a parameter set that minimizes a cost function. However, these methods do not guarantee convergence towards a global minima and are thus reliant on the initial starting parameter. Examples of gradient-based optimizers include Steepest descent, Gauss-Newton and Levenberg-Marquardt algorithms, among many others [23]. Next, non-gradient based algorithms tend to display slower convergence towards a parameter set that minimizes a cost function and this is usually attributed to the fact that a large initial population size, of different parameter sets over a search space is required [14]. However, these optimizers are reputed to be more adept at handling noisy experimental data, than gradient based optimizers and are also more efficient means of finding a global minima in a search space [14]. Examples of non-gradient based algorithms include Genetic Search and Particle Swarm algorithms, among many others. For the purposes of this research, a gradient based optimizer based

on the Levenberg-Marquardt algorithm is considered as a means of minimizing cost functions, in order to obtain desired material parameters. This choice is guided by the fact that the algorithm is devoted to non-linear least squares curve fitting applications [20].

Various methods for constrained optimization using the Levenberg-Marquardt algorithm have been presented in the literature [24, 25], from which there appears to be three different variations. Kanzow et al. [24], explored the first two variations for constrained optimization using the Leveneberg-Marquardt algorithm. The first approach solves a strictly convex minimization problem at each iteration, whereas the second method, which is referred to as a projected Levenberg-Marquardt algorithm, only solves one equation system at each iteration. Both methods have been shown to converge near quadratically under an error bound assumption and can be globalized in an easy manner [24]. The last method was employed by Guyon and Le Riche [25] and proposes that parameter constraints are handled by checking the Kuhn and Tucker condition of the constrained problem. For the purposes of this research, the simplicity and effectiveness of the projected Levenberg-Marquardt algorithm, proposed by Kanzow et al. [24], is considered as a means of minimizing cost functions for the inverse problem.

1.3 Cardiac Mechanics and Material Parameter Identification

Realistic left ventricle geometries can be derived from MRI images, taken at various cross-sections of the heart, in the undeformed and deformed states. However, the mechanical response of the left ventricle is anisotropic in nature, due to the embedded fibres in the myocardium [13]. To simulate the mechanical response of realistic left ventricle geometries, an appropriate mathematical description is required for the embedded fibre orientations of the myocardium, at all locations in the body. For this purpose, Niederer et al. [26] used data from animal studies to obtain a fibre field, which was subsequently utilized for mechanical simulations of a realistic left ventricle geometry. However, the use of realistic geometries and their associated embedded fibre orientation, is beyond the scope of this research. Instead, a truncated ellipsoidal model will be considered, whose geometry represents a close approximation to that of the real physical system. Similar approaches were employed by Niederer and Smith [27] and Kerckhoffs et al. [17], to model cardiac function for a rat and canine left ventricle, respectively.

To accurately model cardiac function, several sub-systems need to be considered. Firstly, the passive mechanics of the heart, which describe the response of the former to external

forces such as blood filling pressure, have to be considered. In this regard, the passive constitutive model of Usyk et al. [28] is considered, which is an example of a Fung type constitutive law, the latter being phenomenologically derived. In addition, the active mechanics, describing the contraction of myocardium, are modelled using the constitutive model proposed by Guccione et al. [29]. With respect to active mechanics, the active contraction in the ventricle is initiated at a microscopic level, and its propagation throughout the system is asynchronous with respect to time [17, 24]. To model this behaviour, Kerckhoffs et al. [17], coupled an active stress model to the Eikonal diffusion equation, the resulting electrical wave propagation provided a timing map for the active contraction initiation in the left ventricle. Niederer and Smith [27], chose to utilize a bidomain model to capture the asynchronous timing of active contraction initiation in the ventricle. In particular, when using the Eikonal diffusion equation, the timing of active contraction initiation can be calibrated such, that it appears physiologic, however, myofibre strain distribution in the ventricle, maybe unphysiologically inhomogeneous [30]. However, when the initiation of active contraction is assumed to be unphysiologically synchronous, then, the distribution of myofibre strain in the ventricle, can appear more homogeneous and physiologic [30]. Hence, as an initial starting point, this work assumes that the active contraction in the ventricle occurs in a synchronous manner. Next, during the ejection phase of the cardiac cycle, the load faced by the left ventricle in pumping blood into the systemic arterial tree needs to be captured. For this, the three element Windkessel model is coupled to the mechanical response of the left ventricle in ejection, following the works of Kerckhoffs et al. [17] and Bovendeerd et al. [12]. Furthermore, residual stresses are present in the myocardium at the start of diastolic filling, and exist primarily in the cross fibre direction [9]. However, these residual stresses are generally hard to quantify [9], and are thus ignored in this work.

Inherent to the active and passive mechanical models of the myocardium of the left ventricle, are material parameters that govern both the respective behaviours. In addition, material parameters that are representative of the arterial system are inherent to the three element Windkessel model. In order to properly simulate left ventricular mechanical response using these models, relevant material parameters must be identified that produce agreement between simulated and experimental data (e.g. stress, strain data and/or pressure-volume relations over the cardiac cycle). Recently, using an inverse approach, Göktepe et al. [31] calibrated passive material parameters of the ventricular myocardium of pig hearts, using stress-strain data from simple shear experiments on cubic samples of the tissue. In this case, the authors identified the parameters by minimizing the mean square error between simulated and experimentally observed tissue strains, at various stress states during the employed loading protocol. A similar approach was used by Dokos et al. [32] to identify passive material parameters of

pig hearts. However, using these methods of passive material parameter identification, the computationally observed left ventricular mechanical responses, in terms of stress, strain or pressure-volume relations, are not guaranteed to correspond to those observed experimentally. The reason for the discrepancy according to Einstein et al. [33] is that the continuum properties of the myocardium of the left ventricle differ greatly in-situ and ex-situ, due to residual stresses, water content, degree of cross linking, temperature and so on. Additionally, another contributing factor to the discrepancy is that within the left ventricle, a degree of heterogeneity of spatially varying material properties of the myocardium exists [33]. In light of these remarks, Wenk et al. [34] utilized a different approach to the identification of passive parameters. Firstly, the authors constructed a three dimensional geometry of the left ventricle, using MRI data from sheep hearts. In addition, in this model, the authors accounted for residual stresses in the myocardium, which exist predominantly in the cross fibre direction. Then, passive parameters were identified during diastolic filling, by minimizing the differences between regional cardiac strains[1] found from their computational model and those found experimentally. In terms of active tension material parameter identification, the authors used the same approach, however, in this case, the differences between regional fibre strains found from the computational model and those found experimentally, were minimized during systole (isovolumetric contraction and ejection). Einstein et al. [33] used a similar approach to calibrating active tension parameters. However, these authors chose to conduct the active tension parameter identifications, by minimizing the mean square error between regional strains found computationally and those found experimentally. Furthermore, common approaches to calibrating Windkessel parameters (e.g. those of the three element Windkessel model), include minimization of the mean squared error between experimentally observed and simulated pressures-time and flow rate-time relationships, in the proximal aorta [35, 36].

Hence, for the purposes of this research it would be preferable to identify passive and active tension material parameters (active and passive), using strain or displacement data derived from MRI images, at various cross-sections of the left ventricle, at representative points in the cardiac cycle. However, this method of parameter identification is not adopted by the author, because a mathematical framework to model the realistic fibre orientations of the myocardium in these realistic geometries, has yet to be implemented in SESKA. In addition, detailed full-field displacement or strain data, for idealized ellipsoidal models of the left ventricle are unavailable in the literature. Consequently, in this work active tension, passive and Windkessel material parameters are calibrated using left ventricular pressure and volume data. Such data is abundant in the literature for a variety of animals, including rat and canine. However, as an initial starting point, reference

[1] Strains at various positions in the left ventricle

pressure and volume data was adapted from the work of Kerckhoffs et al. [17], who also utilized a truncated canine ellipsoidal model. Consequently, for numerical simulations using SESKA, the same canine truncated ellipsoidal model geometry is considered by the author. Additionally, it is noted that pressure-volume data is particularly well suited for Windkessel parameter identification. To elaborate, it was earlier mentioned that these parameters are usually identified by minimizing the mean squared error between experimentally observed and simulated pressures-time and flow rate-time relationships, in the proximal aorta. Additionally, aortic pressure and flow rate can be directly linked to left ventricular pressure and volume during ejection. Firstly, the aortic pressure at the inlet to the aorta is approximately equal to left ventricular pressure during ejection [3]. Furthermore, at this inlet, the aortic flow rate into the aorta, is equal to the negative rate of change of left ventricular cavity volume, during ejection [3, 17]. Consequently, experimentally measured left ventricular pressure and volume during ejection, represent viable data for the calibration of Windkessel parameters. Furthermore, in order to conduct a mechanical analysis of left ventricular behaviour over a full cardiac cycle, a closed loop pressure-volume curve is required. Consequently, passive, active tension and Windessel parameters are calibrated using pressure-volume data, such that a closed loop cardiac cycle is achieved, for a single beat of a canine left ventricle.

1.4 Objectives of the Research

In this work, an inverse approach is employed to calibrate passive, active tension and Windkessel material parameters, to produce a closed loop pressure-volume curve for the first beat of a cardiac cycle. Furthermore, updating methods of inverse modelling are employed, which require an objective cost function that is to be minimized, to obtain material parameters. To conduct the inverse modelling, the projected Levenberg-Marquardt algorithm is selected as the non-linear optimization algorithm, that will be used to minimize the constructed cost functions. Furthermore, the passive and active constitutive models were previously implemented in the in-house modelling software SESKA. However, to model the full cardiac cycle, a three element Windkessel model, is coupled to the left ventricular mechanical response during ejection. Consequently, the objectives of this research are summarized in the following points:

- Implement the projected Levenberg-Marquardt algorithm, in the in-house modelling software SESKA.
- Implement the three element Windkessel model, in the SESKA software, and couple it to the mechanical response of the left ventricle in ejection.

- Formulate an objective cost function for passive material parameter identification.
- Formulate an objective cost function for active tension and Windkessel parameter identification.
- Minimize the objective cost functions, to obtain passive, active tension and Windkessel parameters, that produce a closed loop pressure-volume curve, for a single beat of a cardiac cycle.

1.5 Organization of the Dissertation

Chapter two will outline elements of continuum mechanics, which serve as the basis for the computation of the mechanical response of the left ventricle.

Next, in chapter three, the anatomy and function of the left ventricle will be explained. This information, is necessary for the selection of appropriate constitutive models, that govern the passive and active mechanical behaviour of the myocardium of the left ventricle. Furthermore, the anatomy and function of the circulatory system will be explained. Consequently, an appropriate model for the systemic or arterial circulatory system is selected.

Chapter four will explore the idealized truncated canine ellipsoid model, used for all simulations of cardiac function in this dissertation. In this chapter, the model geometry will be explained, as well as the manner in which the orientation and direction of embedded fibres, in the myocardium is modelled.

Chapter five will unpack the manner in which cardiac function is modelled within the in-house modelling software SESKA.

Chapter six, will summarize the concept of inverse modelling. Furthermore, methods for constructing objective cost functions will be illustrated. Next, the projected Levenberg-Marquardt algorithm will be presented, as a means of minimizing cost functions, to obtain material parameters.

Chapter seven will present the results of this work. Firstly, a benchmark example will be presented as a means of illustrating the behaviour of the implemented projected Levenberg-Marquardt algorithm. Next, the results of material parameter identification, in the context of cardiac mechanics modelling will be presented. Furthermore, a full mechanical analysis of the behaviour of the left ventricle will be provided.

Finally, in chapter eight, the conclusions derived from the obtained results, will be presented. In addition, further areas of research will be proposed, with regards to inverse and mechanical modelling.

Chapter 2

Continuum Mechanics

2.1 Introduction

The constituents of myocardium of the left ventricle, are small enough to facilitate a continuum mechanics based approach to modelling their behaviour, in terms of stress, strain and deformation. In continuum mechanics, the fundamental assumption made is that if one continuously magnifies a specific piece of a material, the underlying molecular structure is generalised by the overlying material properties. Thus, each point can now be subjected to any kind of stress and strain, based on certain physical and material laws or kinematic quantities, in the same way as any structure, irrespective of its size. From this, Continuum Mechanics can now provide the means to analyse a body, being subjected to different boundary conditions, by utilising any kind of discretisation strategies. In this chapter, fundamental equations related to non-linear continuum mechanics will be introduced and the relations of these equations to stress and strain will be highlighted, as they have been employed quite extensively afterwards. For more detailed descriptions and derivations, the reader may find the following book useful: Mase and Mase [37].

2.2 Kinematics

Consider a body, $\mathcal{B}$, in a three dimensional Euclidean vector space characterized as the reference configuration, which is the undeformed configuration. $\mathcal{B}$ is parametrized by the Cartesian coordinates X_1, X_2 and X_3, where the associated basis vectors are denoted by $\mathbf{e}_1$, $\mathbf{e}_2$ and $\mathbf{e}_3$, respectively. After being subjected to certain boundary conditions, the body at time t, in its deformed state (or current configuration), is referred to as $\mathcal{B}_t$. To describe the transformation from the reference to the current configuration, a non linear deformation mapping, $\phi_t : \mathcal{B} \rightarrow \mathcal{B}_t$, is introduced, enabling one to define the

relation of a material point in the reference configuration, $\mathbf{X} \in \mathcal{B}$, to that in the current configuration $\mathbf{x} \in \mathcal{B}_t$ as

$$\mathbf{x} = \phi_t(\mathbf{X}, t). \tag{2.1}$$

The deformation gradient, $\mathbf{F}$, can be obtained as the gradient of ϕ_t. Next, the Jacobian, defined as the determinant of $\mathbf{F}$, can be related to the deformation mapping, ϕ_t, using

$$J = \det \mathbf{F} = \det(\operatorname{Grad} \phi_t). \tag{2.2}$$

The operators with respect to the reference and current configuration are considered in the Tab. 2.1.

TABLE 2.1: Gradient and Divergence operation in the Reference and Current configuration

Operator	Reference Configuration	Current Configuration
Gradient	$\operatorname{Grad}(\bullet) = \boldsymbol{\nabla}(\bullet) = \frac{\partial}{\partial \mathbf{X}}(\bullet)$	$\operatorname{grad}(\bullet) = \boldsymbol{\nabla}_t(\bullet) = \frac{\partial}{\partial \mathbf{x}}(\bullet)$
Divergence	$\operatorname{Div}(\bullet) = \boldsymbol{\nabla} \cdot (\bullet) = \frac{\partial}{\partial \mathrm{X}_i}(\bullet) \cdot \mathbf{e}_i$	$\operatorname{div}(\bullet) = \boldsymbol{\nabla}_t \cdot (\bullet) = \frac{\partial}{\partial \mathrm{x}_i}(\bullet) \cdot \mathbf{e}_i$

The displacement field is given by, $\mathbf{u}(\mathbf{X}, t) = \mathbf{x}(\mathbf{X}, t) - \mathbf{X}$. Hence, the deformation gradient, $\mathbf{F}$, can be alternatively formulated in terms of $\mathbf{u}(\mathbf{X}, t)$ as

$$\mathbf{F} = \mathbf{1} + \boldsymbol{\nabla}(\mathbf{u}(\mathbf{X}, t)). \tag{2.3}$$

Next, any change in time is termed the local rate of change expressed as

$$\frac{\partial}{\partial t}(\bullet). \tag{2.4}$$

Therefore, the overall time rate of change is

$$\frac{D}{Dt}(\bullet) = \frac{\partial}{\partial t}(\bullet) + \frac{\partial}{\partial x_i}(\bullet)\frac{dx_i}{dt}. \tag{2.5}$$

Consequently, the Jacobian, which defines the change of volumes from different configuration states, can also vary with respect to time according to

$$\dot{J} = J \operatorname{div} \dot{\mathbf{x}}. \tag{2.6}$$

For the later usage, we also take into account that a surface element in the reference configuration, dA, with its unit normal, $\mathbf{n}$, is related to its counterpart in the current configuration, da, with its unit normal, $\boldsymbol{\nu}$, via

$$\boldsymbol{\nu} da = \det(\mathbf{F}) \mathbf{F}^{-\mathrm{T}} \mathbf{n} dA. \tag{2.7}$$

In addition, a volume element in the reference configuration, dV, is related to its counterpart in the current configuration, dv, by

$$dv = \det(\mathbf{F})dV = JdV. \tag{2.8}$$

Next, the change in length of a line after deformation would require the difference between the length in the reference configuration, $d\mathbf{x}^2$ and that in the current configuration, $d\mathbf{X}^2$ and is computed as

$$d\mathbf{x}^2 - d\mathbf{X}^2 = (\mathbf{F}^{\mathrm{T}}\mathbf{F} - \mathbf{1}) : (d\mathbf{X} \otimes d\mathbf{X}). \tag{2.9}$$

Based on the above, the $right\ Cauchy - Green$ tensor is defined as

$$\mathbf{C} = \mathbf{F}^T\mathbf{F}. \tag{2.10}$$

Thus, the *Green* strain tensor, which is symmetric in nature, can now be introduced as a function of $\mathbf{C}$ via

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{1}) = \frac{1}{2}(\mathbf{F}^T\mathbf{F} - \mathbf{1}). \tag{2.11}$$

The $right\ Cauchy - Green$ tensor can also be expressed in terms of the right stretch tensor, $\mathbf{U}$, as

$$\mathbf{C} = \mathbf{U}^2. \tag{2.12}$$

Finally, using the eigenvalues of $\mathbf{U}$, the scalar invariants of $\mathbf{C}$ are

$$I_1^{\mathrm{C}} = tr(\mathbf{C}) = \lambda_1^2 + \lambda_2^2 + \lambda_3^2, \tag{2.13}$$

$$I_2^{\mathrm{C}} = \frac{1}{2}\left[(tr\mathbf{C})^2 - tr(\mathbf{C}^2)\right] = \lambda_1^2\lambda_2^2 + \lambda_2^2\lambda_3^2 + \lambda_3^2\lambda_1^2, \tag{2.14}$$

$$I_3^{\mathrm{C}} = \det \mathbf{C} = \lambda_1^2\lambda_2^2\lambda_3^2. \tag{2.15}$$

2.3 Stress Measures

Bodies subjected to forces will undergo deformation, resulting in a redistribution of forces throughout the body. Consider a body in the current configuration, $\mathcal{B}_t$, subjected to a body force, $\mathbf{b}_t$, and a surface traction, $\mathbf{t}^{(\boldsymbol{\nu})}$, which is the traction vector acting on a surface with a unit normal, $\boldsymbol{\nu}$. Then the following relation holds;

$$\mathbf{t}^{(\boldsymbol{\nu})}(\mathbf{x}, t) = \boldsymbol{\sigma}^{\mathrm{T}}(\mathbf{x}, t)\boldsymbol{\nu}(\mathbf{x}, t), \tag{2.16}$$

where $\boldsymbol{\sigma}(\mathbf{x}, t)$ is defined as the *Cauchy* stress tensor in the spatial coordinate of $\mathbf{x}$. If the stress state of the same point in the reference configuration is to be found, then the element surface to be considered is dA, with unit normal of $\mathbf{n}$. Consequently, the traction vector is expressed as

$$\mathbf{t}^{(\mathbf{n})}(\mathbf{X}, t) = \mathbf{P}(\mathbf{X}, t)\mathbf{n}(\mathbf{X}, t), \tag{2.17}$$

where, $\mathbf{P}(\mathbf{X}, t)$ is the $first\ Piola - Kirchoff$ stress tensor. $\mathbf{P}(\mathbf{X}, t)$ relates the stresses with forces in the current configuration, as acting on areas in the reference configuration. Hence, the deformation gradient, $\mathbf{F}$, can be applied to map the forces to the reference configuration. In this case, the new stress tensor is defined as the $second\ Piola - Kirchoff$ stress tensor $\mathbf{S}$ using the relation

$$\mathbf{P} = \mathbf{FS}. \tag{2.18}$$

Using $\boldsymbol{\sigma}$ (the *Cauchy* stress tensor), $\mathbf{P}$ can be defined differently, by first assuming that a force, $\mathbf{f}$, acting on da is no different to the one applied on dA, yielding,

$$d\mathbf{f} = \mathbf{P}\mathbf{n}dA = \boldsymbol{\sigma}^{\mathrm{T}}\boldsymbol{\nu}da = \boldsymbol{\sigma}^{\mathrm{T}} \det(\mathbf{F})\mathbf{F}^{-\mathrm{T}}\mathbf{n}dA. \tag{2.19}$$

Consequently, the first Piola-Kirchoff stress, can be reformulated as

$$\mathbf{P} = \det(\mathbf{F})\boldsymbol{\sigma}^{\mathrm{T}}\mathbf{F}^{-\mathrm{T}}. \tag{2.20}$$

2.4 Balance Law of Continuum Mechanics

If a body is subjected to perturbations of any form (e.g. force, temperature or magnetic field), then the following fundamental physical laws can be used to describe the behaviour of the body [37]:

- **The law of Conservation of Mass** ensures that there is no mass loss or gain, when the body changes from the reference to the current configuration.
- **The law of Linear and Angular Momentum Conservation** is related to Newton's Second Law, which defines the rate of change of linear and angular momentum over time, resulting in force and momentum.
- **The Law of Energy Conservation** is related to the principle law of energy, which states that energy can neither be created nor destroyed, but can only change forms. However, in this project, we focus mainly on the conversion of energy into deformations.

2.4.1 Conservation of Mass

The mass m of a material continuum body, $\mathcal{B}_t$, at time t is given by

$$m = \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) dv, \tag{2.21}$$

where, ρ is the density in the current configuration. The *Law of Conservation of Mass*, states that the cumulative mass of a body, $\mathcal{B}_t$, is conserved with motion. Hence, the material derivative of Eq. (2.21) is zero, yielding,

$$\dot{m} = \frac{d}{dt} \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) dv = \int_{\mathcal{B}_t} \{\dot{\rho}(\mathbf{x}, t) + \rho(\mathbf{x}, t) J \mathrm{div}\dot{\mathbf{x}}\} dv = 0, \tag{2.22}$$

where the relations in Eq. (2.8) and Eq. (2.6) were used. Since $\mathcal{B}_t$ is arbitrary, the integrand in Eq. (2.22) must vanish, resulting in the so-called *continuity equation*, which in the Eulerian form is

$$\dot{\rho}(\mathbf{x}, t) + \rho(\mathbf{x}, t) J \mathrm{div}\dot{\mathbf{x}} = 0. \tag{2.23}$$

Furthermore, conservation of mass also states that the mass in the current and reference configurations are equal. Therefore,

$$m = \int_{\mathcal{B}_t} \rho(\mathbf{x}, t) dv = \int_{\mathcal{B}} \rho_0(\mathbf{X}) dV, \tag{2.24}$$

where ρ_0 denotes the density of the material in the reference configuration. With Eq. (2.8) and recognizing that $\mathbf{x} = (\mathbf{X}, t)$, Eq. (2.21) can be expressed in Lagrangian form by

$$m = \int_{\mathcal{B}} \rho(\mathbf{X}, t) J dV, \tag{2.25}$$

and substituting Eq. (2.25) into the left-hand side of Eq. (2.24), we arrive at

$$\int_{\mathcal{B}} \{\rho(\mathbf{X}, t) J - \rho_0(\mathbf{X})\} dV = 0. \tag{2.26}$$

Since $\mathcal{B}$ is arbitrary, the integrand in Eq. (2.26) must vanish, yielding the Lagrangian or material form of the continuity equation, which is given by

$$\rho(\mathbf{X}, t) J = \rho_0(\mathbf{X}). \tag{2.27}$$

With $\dot{\rho_0} = 0$ we further have that

$$\frac{d}{dt} (\rho(\mathbf{X}, t) J) = 0, \tag{2.28}$$

which is a relation that is particularly useful in evaluating the material derivative of an integral over some product ρA,

$$\frac{d}{dt}\int_{\mathcal{B}_t}\rho\left(\mathbf{x},t\right)A\left(\mathbf{x},t\right)dv=\frac{d}{dt}\int_{\mathcal{B}_t}\rho\left(\mathbf{x},t\right)\dot{A}\left(\mathbf{x},t\right)dv, \tag{2.29}$$

where $A\left(\mathbf{x},t\right)$ is a field of some property per unit mass.

2.4.2 Linear Momentum Conservation

If a body, $\mathcal{B}_t$, in the current configuration, is subjected to a body force, $\mathbf{b}_t$ and a surface traction, $\mathbf{t}^{\boldsymbol{\nu}}$, is applied to a boundary surface, $\partial\mathcal{B}_t$, then the principle of linear momentum conservation states that the rate of change of linear momentum, is equal to the total resultant force acting on the body, which yields,

$$\frac{d}{dt}\int_{\mathcal{B}_t}\rho(\mathbf{x},t)\dot{\mathbf{x}}dv=\int_{\partial\mathcal{B}_t}\mathbf{t}^{(\boldsymbol{\nu})}(\mathbf{x},t)da+\int_{\partial\mathcal{B}_t}\mathbf{b}_t(\mathbf{x},t)dv. \tag{2.30}$$

Introducing the *Cauchy* stress and making use of the Gauss divergence theorem, Eq. (2.30), is transformed into the global equation of motion, given by

$$\int_{\mathcal{B}_t}\{\rho(\mathbf{x},t)\ddot{\mathbf{x}}-\mathrm{div}\boldsymbol{\sigma}^{\mathrm{T}}-\mathbf{b}_t(\mathbf{x},t)\}dv=\mathbf{0}. \tag{2.31}$$

The local form of the equation of motion is obtained if one acknowledges that $\mathcal{B}_t$ is arbitrary, resulting in the relation

$$\mathrm{div}\boldsymbol{\sigma}^{\mathrm{T}}+\mathbf{b}_t(\mathbf{x},t)=\rho(\mathbf{x},t)\ddot{\mathbf{x}}. \tag{2.32}$$

For a constant velocity field, $\dot{\mathbf{x}}(\mathbf{x},t)$, the acceleration field, $\ddot{\mathbf{x}}$, is zero, reducing Eq. (2.32) to the equilibrium equation, given by

$$\mathrm{div}\boldsymbol{\sigma}^{\mathrm{T}}+\mathbf{b}_t(\mathbf{x},t)=\mathbf{0}. \tag{2.33}$$

The above steps can be repeated for quantities in the reference configuration, $\mathcal{B}$. The global equation of motion now becomes

$$\frac{d}{dt}\int_{\mathcal{B}}\rho_0\dot{\mathbf{x}}(\mathbf{X},t)dV=\int_{\partial\mathcal{B}}\mathbf{t}^{(\boldsymbol{\nu})}(\mathbf{X},t)dA+\int_{\partial\mathcal{B}}\mathbf{b}(\mathbf{X},t)dV, \tag{2.34}$$

where $\mathbf{b}$ is the body force in the reference configuration. With the elimination of the traction vector and applying the Gauss divergence theorem, Eq. (2.34) becomes

$$\int_{\mathcal{B}}\{\rho_0\ddot{\mathbf{x}}(\mathbf{X},t)dV-\mathrm{div}\mathbf{P}-\mathbf{b}(\mathbf{X},t)\}\,dV=\mathbf{0}. \tag{2.35}$$

Acknowledging that $\mathcal{B}$ is arbitrary, one can obtain the localized equation of motion, which shall now be called the *Lagrangian equation of motion*, as

$$\mathrm{div}\mathbf{P} + \mathbf{b}(\mathbf{X}, t) = \rho_0 \ddot{\mathbf{x}}(\mathbf{X}, t). \tag{2.36}$$

Finally, the Lagrangian equilibrium equation with constant velocity field is given by

$$\mathrm{div}\mathbf{P} + \mathbf{b}(\mathbf{X}, t) = \mathbf{0}. \tag{2.37}$$

2.4.3 Angular Momentum Conservation

Angular momentum is defined as a moment of the linear momentum, about a particular reference point, which is usually taken as the origin of the coordinate system [37]. Thus, in the spatial or current configuration, the principle of angular momentum conservation is formulated as

$$\frac{d}{dt}\int_{\mathcal{B}_t} \mathbf{x} \times \rho(\mathbf{x}, t)\dot{\mathbf{x}} dv = \int_{\partial\mathcal{B}_t} \mathbf{x} \times \mathbf{t}^{(\boldsymbol{\nu})}(\mathbf{x}, t) da + \int_{\partial\mathcal{B}_t} \mathbf{x} \times \mathbf{b}_t(\mathbf{x}, t) dv. \tag{2.38}$$

Normalising the integrands by taking the material derivative, using the *Cauchy* stress tensor, $\boldsymbol{\sigma}$, and applying the Gauss divergence theorem, we obtain that

$$\int_{\mathcal{B}_t} \left\{\mathbf{x} \times \rho(\mathbf{x}, t)\ddot{\mathbf{x}} - \mathrm{div}(\mathbf{x} \times \boldsymbol{\sigma}^{\mathrm{T}}) - \mathbf{x} \times \mathbf{b}_t(\mathbf{x}, t)\right\} dv = \mathbf{0}, \tag{2.39}$$

with

$$\mathrm{div}(\mathbf{x} \times \boldsymbol{\sigma}^{\mathrm{T}}) = \mathrm{Grad}\,\mathbf{x} \times \boldsymbol{\sigma}^{\mathrm{T}} + \mathbf{x} \times \mathrm{div}(\boldsymbol{\sigma}^{\mathrm{T}}). \tag{2.40}$$

Simplifying the above, Eq. (2.39) is transformed to

$$\int_{\mathcal{B}_t} \left\{\mathbf{x} \times \rho(\mathbf{x}, t)\ddot{\mathbf{x}} + \boldsymbol{\epsilon} : \boldsymbol{\sigma}^{\mathrm{T}} - \mathbf{x} \times \mathrm{div}\boldsymbol{\sigma}^{\mathrm{T}} - \mathbf{x} \times \mathbf{b}_t(\mathbf{x}, t)\right\} dv = \mathbf{0}, \tag{2.41}$$

where $\boldsymbol{\epsilon}$ denotes the permutation tensor. Since $\mathcal{B}_t$ in Eq. (2.41) is arbitrary we get that

$$\mathbf{x} \times \rho(\mathbf{x}, t)\ddot{\mathbf{x}} - \mathbf{x} \times \mathrm{div}\boldsymbol{\sigma}^{\mathrm{T}} - \mathbf{x} \times \mathbf{b}_t(\mathbf{x}, t) + \boldsymbol{\epsilon} : \boldsymbol{\sigma}^{\mathrm{T}} = \mathbf{0}. \tag{2.42}$$

Then, with Eq. (2.32) we have that

$$\boldsymbol{\epsilon} : \boldsymbol{\sigma}^{\mathrm{T}} = \mathbf{0}. \tag{2.43}$$

The above equation demonstrates the symmetric properties of the *Cauchy* stress tensor. Consequently, the Lagrangian form of the above equations, can now be formulated.

Firstly, the principle of angular momentum conservation can be reformulated as

$$\frac{d}{dt}\int_{\mathcal{B}} \mathbf{x}(\mathbf{X},t)\times\rho_0\dot{\mathbf{x}}(\mathbf{X},t)dV = \int_{\partial\mathcal{B}} \mathbf{x}(\mathbf{X},t)\times\mathbf{t}^{(\mathbf{n})}(\mathbf{X},t)dA + \int_{\partial\mathcal{B}} \mathbf{x}(\mathbf{X},t)\times\mathbf{b}(\mathbf{X},t)dV. \tag{2.44}$$

Taking the divergence of $(\mathbf{x}(\mathbf{X},t)\times\mathbf{P})$ we get that

$$\int_{\mathcal{B}} \{\mathbf{x}(\mathbf{X},t)\times\rho_0\ddot{\mathbf{x}}(\mathbf{X},t)dV - \mathrm{div}(\mathbf{x}(\mathbf{X},t)\times\mathbf{P}) - \mathbf{x}(\mathbf{X},t)\times\mathbf{b}(\mathbf{X},t)\}\, dV = \mathbf{0}. \tag{2.45}$$

Consequently, using Eq. (2.36)

$$\int_{\mathcal{B}} \mathbf{F}\times\mathbf{P}dV = \mathbf{0}, \tag{2.46}$$

and since the integrand is arbitrary,

$$\mathbf{F}\times\mathbf{P} = \boldsymbol{\epsilon} : \left(\mathbf{F}\mathbf{P}^{\mathrm{T}}\right) = \mathbf{0}. \tag{2.47}$$

Thus, the symmetric nature of $\mathbf{F}\mathbf{P}^{\mathrm{T}}$ is demonstrated, which means that

$$\mathbf{F}\mathbf{P}^{\mathrm{T}} = \mathbf{P}\mathbf{F}^{\mathrm{T}}. \tag{2.48}$$

Finally, since $\mathbf{S} = \mathbf{P}\mathbf{F}^{\mathrm{T}}$, it follows that the second Piola-Kirchoff stress is also symmetric.

2.5 Constitutive Laws

The stress in the body, $\mathcal{B}$, depends not only on its deformation, but also on its material properties. In most materials, when the deformation is small, the stress is a linear function of the strain. The relationship between stress and strain is called the constitutive law. Consequently, to describe the deformation of a material, we are required to define stress as a function of strain, $\mathbf{S}(\mathbf{E})$. For example, the St. Venant-Kirchoff model, is described by the constitutive law relating stress with strain as

$$\mathbf{S}(\mathbf{E}) = \lambda\mathrm{tr}(\mathbf{E})\mathbf{I} + 2\mu\mathbf{E}, \tag{2.49}$$

where λ and μ are the so-called *Lame* constants. These material constants can be chosen to model a range of materials. Eq. (2.49) represents a constitutive law, where the relationship between stress and strain is linear. However, when the deformations in the material are large, the relationship between stress and strain becomes non-linear. Examples of such materials are rubber or soft biological tissues, which are both examples of hyperelastic materials. For the case of homogeneous, hyperelastic materials, stress and

strain can be related via an elastic potential, ψ, which represents the stored strain energy per unit volume or *strain energy function* . If the strain energy function is formulated in terms of the Green strain tensor, i.e., $\psi(\mathbf{E})$, then the second Piola-Kirchoff stress tensor is given by

$$\mathbf{S}(\mathbf{E}) = \frac{\partial \psi(\mathbf{E})}{\partial \mathbf{E}}. \tag{2.50}$$

2.5.1 Hyperelastic Materials

For hyperelastic materials, it is convenient to define the strain energy function, in terms of the principle invariants of of the Green strain tensor, $\mathbf{E}$. The principle invariants of the tensor $\mathbf{E}$ are

$$I_1 = \mathrm{tr}\,(\mathbf{E}), \tag{2.51}$$

$$I_2 = \frac{1}{2}\left[(\mathrm{tr}\,\mathbf{E})^2 - \mathrm{tr}\,\left(\mathbf{E}^2\right)\right], \tag{2.52}$$

$$I_3 = \det \mathbf{E}. \tag{2.53}$$

Furthermore, an isotropic material is one where the material properties are independent of the direction in which they are measured. This requires that the strain energy function is objective, which means that it is invariant under any orthogonal coordinate transformation. Hence, the second Piola-Kirchoff stress tensor can finally be written in terms of the scalar invariants of $\mathbf{E}$ as

$$\mathbf{S} = \frac{\partial \psi(I_1, I_2, I_3)}{\partial \mathbf{E}} = \sum_{\alpha=1}^{\alpha=3} \frac{\partial \psi}{\partial I_\alpha} \frac{\partial I_\alpha}{\partial \mathbf{E}}. \tag{2.54}$$

2.5.2 Transverse Isotropy

Materials that exhibit different material properties depending on the direction in which they are measured are called anisotropic [37]. A special case of anisotropic material behaviour, is where the material deformation differs in two orthogonal directions. This form of anisotropic material behaviour is referred to as transverse isotropy. Transverse isotropy in a material occurs due to the microscopic nature of the material.

To mathematically describe this behaviour, the preferred material direction in the reference or Lagrangian description is introduced as the unit vector, $\mathbf{V}_1$. The structural tensor of a transversely isotropic material in the Lagrangian description is thus given by

$$\mathbf{M}_1 = \mathbf{V}_1 \otimes \mathbf{V}_1. \tag{2.55}$$

In addition, the preferred material direction, $\mathbf{V}_1$, results in two additional independent invariants becoming necessary. The additional invariants are I_4 and I_5, which are given respectively by

$$I_4 = \mathrm{tr}\left(\mathbf{M}_1\mathbf{E}\right), \tag{2.56}$$

$$I_5 = \mathrm{tr}\left(\mathbf{M}_1\mathbf{E}^2\right), \tag{2.57}$$

where the square of the tensor, $\mathbf{E}$, is defined as $\mathbf{E}^2 = \mathbf{E}\mathbf{E}$. Finally, the second Piola-Kirchoff stress tensor, can be expressed in terms of the scalar invariants as

$$\mathbf{S} = \frac{\partial\psi(I_1, I_2, I_3, I_4, I_5)}{\partial\mathbf{E}} = \sum_{\alpha=1}^{\alpha=5} \frac{\partial\psi}{\partial I_\alpha}\frac{\partial I_\alpha}{\partial\mathbf{E}}. \tag{2.58}$$

2.5.3 Orthotropy

For an orthotropic material, three independent material directions are present, $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$, which define an orthonormal basis system. In this case, two additional structural tensors need to be defined as

$$\mathbf{M}_2 = \mathbf{V}_2 \otimes \mathbf{V}_2, \tag{2.59}$$

$$\mathbf{M}_3 = \mathbf{V}_3 \otimes \mathbf{V}_3. \tag{2.60}$$

Also, due to the two additional material directions, $\mathbf{V}_2$ and $\mathbf{V}_3$, four additional independent scalar invariants are introduced as

$$I_6 = \mathrm{tr}\left(\mathbf{M}_2\mathbf{E}\right), \tag{2.61}$$

$$I_7 = \mathrm{tr}\left(\mathbf{M}_2\mathbf{E}^2\right), \tag{2.62}$$

$$I_8 = \mathrm{tr}\left(\mathbf{M}_3\mathbf{E}\right), \tag{2.63}$$

$$I_9 = \mathrm{tr}\left(\mathbf{M}_3\mathbf{E}^2\right). \tag{2.64}$$

Consequently, the Green strain tensor may be expressed in a local coordinate system, with basis vectors which are the material directions

$$\begin{aligned}\mathbf{E} =& \mathrm{E}_{11}\mathbf{V}_1 \otimes \mathbf{V}_1 + \mathrm{E}_{22}\mathbf{V}_2 \otimes \mathbf{V}_2 + \mathrm{E}_{33}\mathbf{V}_3 \otimes \mathbf{V}_3 \\ &+ \mathrm{E}_{12}\left(\mathbf{V}_1 \otimes \mathbf{V}_2 + \mathbf{V}_2 \otimes \mathbf{V}_1\right) + \mathrm{E}_{13}\left(\mathbf{V}_1 \otimes \mathbf{V}_3 + \mathbf{V}_3 \otimes \mathbf{V}_1\right) \\ &+ \mathrm{E}_{23}\left(\mathbf{V}_2 \otimes \mathbf{V}_3 + \mathbf{V}_3 \otimes \mathbf{V}_2\right),\end{aligned} \tag{2.65}$$

where E_{ij} represent the components of the local Green strain tensor. Finally, the second Piola-Kirchoff stress tensor can be expressed in terms of the nine scalar invariants as

$$\mathbf{S} = \frac{\partial \psi(I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9)}{\partial \mathbf{E}} = \sum_{\alpha=1}^{\alpha=9} \frac{\partial \psi}{\partial I_\alpha} \frac{\partial I_\alpha}{\partial \mathbf{E}}. \tag{2.66}$$

Chapter 3

Cardiac Mechanics

3.1 Introduction

In the chapter two, the essentials of continuum mechanics necessary for the computation of stresses and deformations in bodies such as the left ventricle were outlined. Therefore, it is possible to elaborate on some essentials of cardiac mechanics in this chapter, which describe the deformation and forces in the heart.

Before the cardiac mechanics of the left ventricle are explained, the structure and gross morphology of the heart, at the macroscopic level will be described, followed by details of the microscopic structure of the myocardium and its constituents. Next, left ventricular cardiac function during a complete cardiac cycle will be considered, an understanding of which is a key component in modelling the left ventricle, within the framework of a computational study. Thus, it will be possible to outline the manner in which the behaviour of the myocardium of the left ventricle is modelled, by exploring the mathematical formulation of various constitutive laws that govern this behaviour. Finally, in order to model the ejection phase of the cardiac cycle, the left ventricular mechanical response needs to be coupled to a model of the global hemodynamics of the blood circulatory system. Therefore, the final portions of this chapter will include an explanation on the anatomy and workings of the circulatory system, followed by an exploration of the models that can be employed to simulate it. For further details on some of the above aspects, the reader is referred to the following textbooks:

- Hurst's the heart [38].
- Computational Cardiology [5].
- Physiology of the Heart [3].

3.2 Heart Structure

In the following sections, the structure of the heart at the macroscopic and microscopic levels will be explored.

3.2.1 Macrostructure

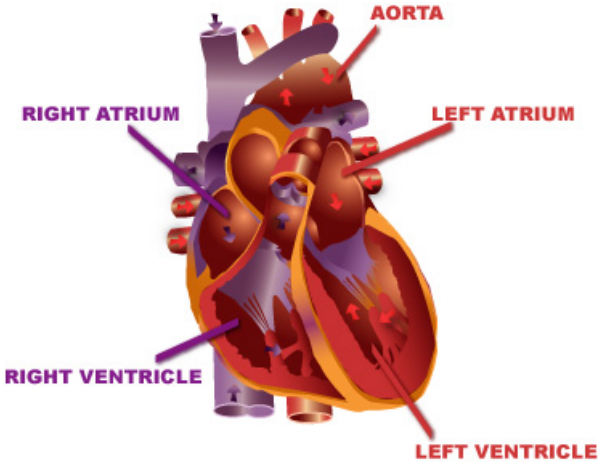

FIGURE 3.1: A labelled heart cross-section [1].

Within the heart, there are four main chambers through which blood passes, namely, the left atrium, right atrium, left ventricle and right ventricle (see Fig. 3.1). The ventricles are the largest chambers as they are responsible for pumping blood out of the heart to the body and lungs, whereas the atria pump blood into the ventricles via valves. The left ventricle is the largest chamber and is also subjected to higher blood pressures than the right ventricle or atria [3]. Consequently, in the case of cardiovascular disease, the left ventricle structure and function is most severely impaired [3]. Hence, the left ventricle is the primary focus of this study.

With regards to the left ventricle, its walls are formed from muscular myocardium. The outer layer of the left ventricle is called the epicardium and the inner layer the endocardium. The muscle between these two layers is referred to as the mid-wall. The bottom tip of the left ventricle is called the apex, while the top of the ventricle is called the base, the later representing the connection of the left ventricle to the left and right atrium.

The valves and arteries play an important role in circulation of blood throughout the body. Firstly, the mitral valve is responsible for controlling blood flow from the left atrium into the left ventricle. Also a large artery, called the aorta, serves as an outlet, leaving the left ventricle, and is responsible for distributing blood to the rest of the body.

Separating the the left ventricle and aorta is the aortic valve, which opens once blood pressure in the left ventricle exceeds the level of pressure in the aorta, allowing blood to flow from the left ventricle onto the rest of the body. The macroscopic structure of the heart is illustrated in Fig. 3.2.

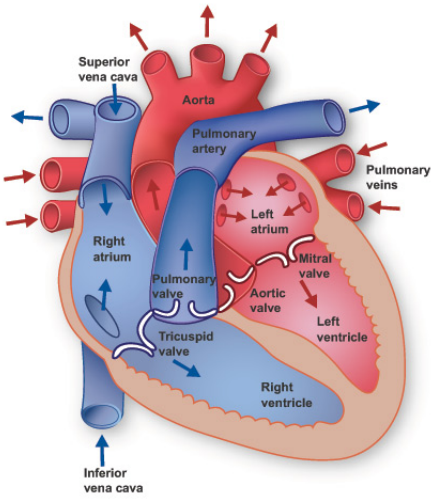

FIGURE 3.2: Macrostructure of the heart [2].

3.2.2 Microstructure

The microscopic nature of the myocardium, is primarily responsible for the manner in which the left ventricle deforms. Hence, in the following sections, the microscopic make-up of the myocardium is explored.

3.2.2.1 Myocytes

The myocardium, contains both myocytes and connective tissue. The cardiac myocytes account for the majority of myocardial mass, however, about 70 percent of the cells are smaller non-myocytes, which include vascular smooth muscle, endothelial cells and fibroblasts [3]. The fibroblasts, secrete and maintain the connective tissue fibres, the latter contributing significantly to the hearts overall stiffness and tensile strength [3]. Furthermore, the connective tissue is organized into the endomysium, the perimysium and the epimysium [3]. The endomysium surrounds individual cardiac myocytes, whereas the perimysium supports groups of cardiac myocytes and the epimysium encases the entire muscle.

In terms of cardiac myocytes, several types are found in the heart [3]. Firstly, working myocytes are found in the atria and ventricles and are specialized for contraction.

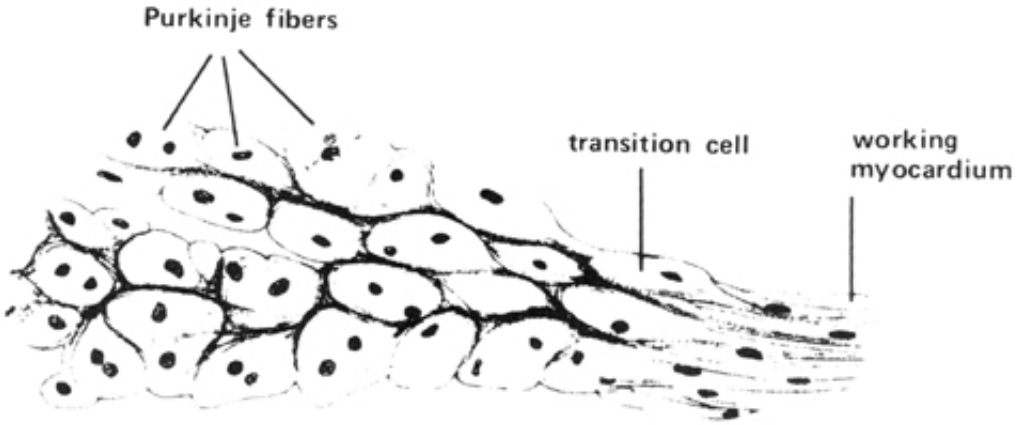

FIGURE 3.3: The branched network formed by the various types of cardiac myocytes [3].

Next, Purkinje fibres are found in the atrioventricular (AV) bundle, bundle branches and ventricular myocardium. These fibres are large, glycogen-rich cells, that are specialized for rapid electrical conduction. Also, transition cells, that are intermediate in appearance between the Purkinje fibres and working myocytes, are located between the former and the latter. These transition cells, like the Purkinje fibres, are glycogen-rich. However, the transition cells are smaller in size than the Purkinje fibres and conduct slowly partly due to this reason. The many different types of cardiac myocytes form a branched network, which is illustrated in Fig 3.3.The working cardiac myocytes are enclosed by a cell membrane, known as sarcolemma, which separates the extra-cellular and intra-cellular spaces. The latter contains the nucleus, sarcoplasmic reticulum, mitochondria, myofibrils, the cytoskeleton and is filled with aqueous cytoplasm. Fig. 3.4 is a schematic illustration of the make-up of a working cardiac myocyte. The working cardiac myocytes are grouped together, forming sheets of parallel myocytes. Components of the intra-cellular space of working myocytes, which are of particular interest are the myofibrils. Working cardiac myocytes themselves are composed of many myofibril chains, which in turn are composed of repeated components called sarcomeres (see Fig. 3.4). These sarcomeres are the contractile elements of the myocardium and are composed of the following two principal fibrous proteins:

- Myosin, which is a thick filament.
- Actin, which is a thin filament.

Each myosin filament comprises of approximately 200 myosin molecules, each having a myosin head. Fig. 3.5 is a schematic illustration of the myofibril micro-structure.

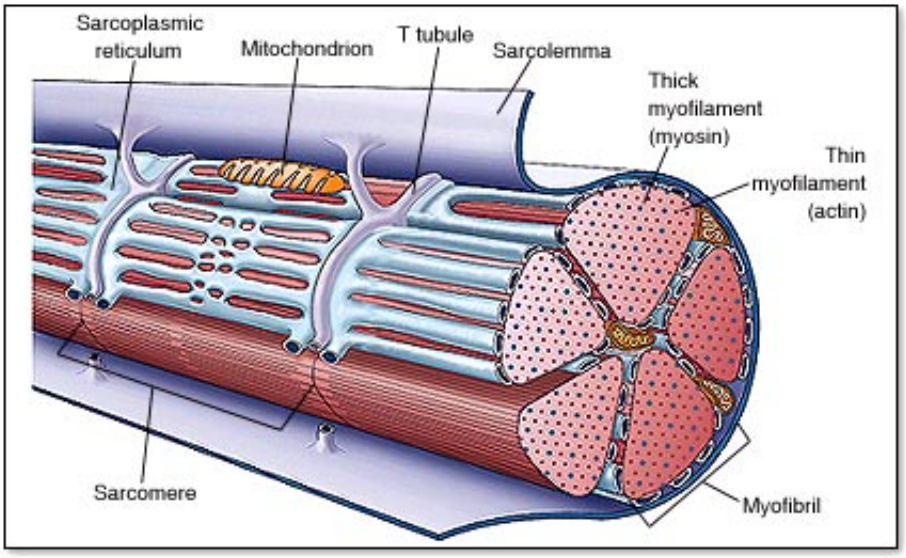

FIGURE 3.4: Schematic illustration of the make-up of a working cardiac myocyte [4].

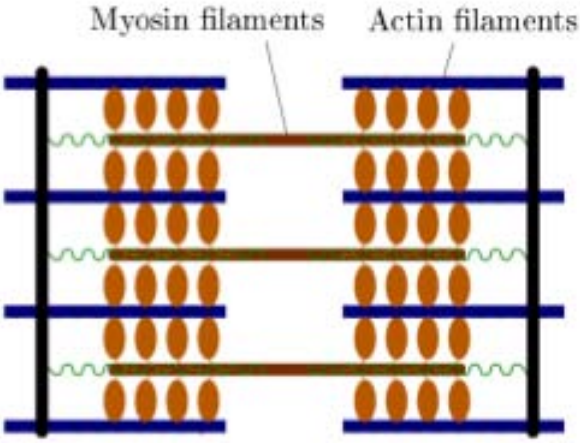

FIGURE 3.5: Schematic of the myofibril micro-structure [5].

3.2.2.2 Electro-Mechanics Initiated on the Microscopic Level

The electro-mechanics of the of the heart can be considered, which are initiated at the microscopic level. Firstly, near the atrial chambers, a region known as the sinoatrial (SA) node exists (see Fig. 3.6). This node generates an electrical wave that propagates to the atrial chambers, where it is held for approximately 0.1 seconds, after which it is released through the HIS bundle. The latter is located at the endocardium and then branches into the myocardium (see Fig. 3.6). The HIS bundle are myocardium cells responsible for conducting electrical signals to the Purkinje fibres. The Purkinje fibres are similar to myocardium fibres, with the exception that they are much thicker, which is a property that facilitates a more efficient and faster channel for the electrical wave to move through. The electrical wave is then transferred to the myocardium fibres, which

have a lower conductivity. At the myocyte level, the propagating electrical wave triggers

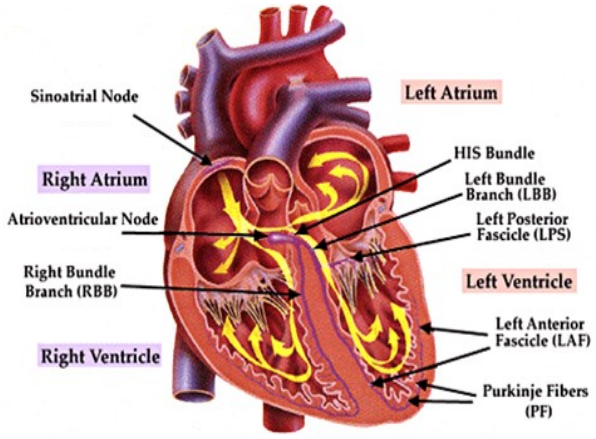

FIGURE 3.6: Illustration of the location of the sinoatrial node and HIS bundle in the heart [6].

an ion current, resulting in a large potential difference between the inside and outside of the negatively charged cell membrane of the myocytes (sarcolemma). This potential difference prompts the opening of the cell membrane gates, allowing certain ions to enter, which restores potential equilibrium. These ions, consist partly of calcium ions. The increased concentration of calcium ions triggers a reaction between the ions themselves and the myosin head. Consequently, the behaviour of the globular head of the myosins changes, as they now attach and pull on the actin proteins, creating a micro-tensile force (see Fig. 3.7). This micro-tensile force is the source of the active contraction of the heart during systole and isovolumetric relaxation.

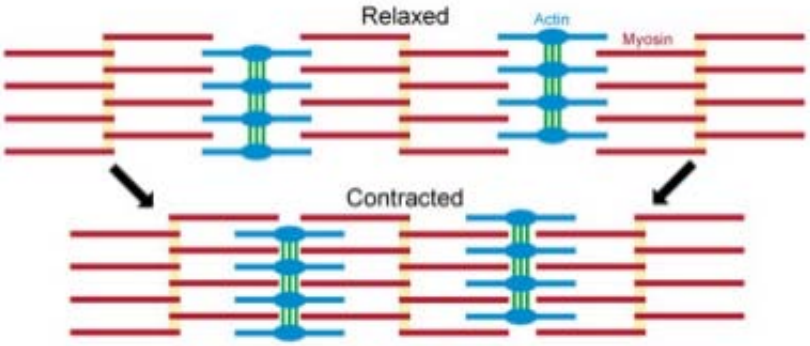

FIGURE 3.7: Sarcomere layout at relaxation and during contraction [7].

3.3 Cardiac Function

During a cardiac cycle the volume of blood within the left ventricle cavity and the pressure on its walls varies with the opening and closing of the valves and the contraction of the heart. Furthermore, the cardiac cycle is split into two phases, namely, systole and diastole. In addition, diastole and systole can be further divided into four phases; diastolic filling (final phase of diastole), isovolumetric contraction (first phase of systole), ejection (second phase of systole) and isovolumetric relaxation (first phase of diastole). Each of these phases are represented on Fig. 3.8, which is a schematic representation of an idealized pressure-volume loop for a single beat of the cardiac cycle.

Referring to Fig. 3.8, during diastolic filling, the mitral valves are open, allowing blood to flow from the left atrium into the left ventricle. Hence the cavity volume and pressure increases, as more blood causes the ventricle to inflate.

During isovolumetric contraction, the aortic and mitral valves are closed, simultaneously, an active contraction, which varies with time is initiated in the ventricle. Since the valves are closed, the cavity volume is constrained, thus, the cavity pressure increases in order to maintain equilibrium.

When the level of pressure in the ventricle exceeds the pressure in the aorta, this signals the beginning of ejection, where the aortic valve opens, and blood is ejected from the contracting ventricle into the aorta. During this phase the cavity volume decreases. In addition, the pressure on the ventricle walls increases initially due to the compliance of the aorta, and then decreases as the rate of ejection of blood from the ventricle begins to decrease.

Once the cavity ceases to contract further, the aortic valve closes and ejection terminates, signalling the beginning of isovolumetric relaxation. During this phase the cavity volume remains constant and the cavity pressure decreases as the ventricle relaxes. The volume at the end of diastolic filling is referred to as the end diastolic volume (EDV), and that at the end of systole is referred to as the end-systolic volume (ESV). Additionally, the stroke volume is defined as the volume of blood ejected during systole, which is the difference between the end-diastolic and end-systolic volumes (see Fig. 3.8). Also, the ejection fraction is the ratio of the stroke volume to the end-diastolic volume.

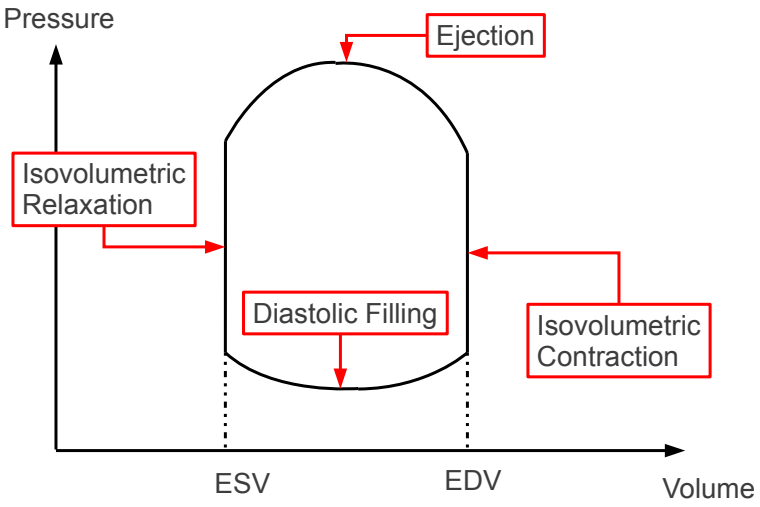

FIGURE 3.8: Idealized pressure volume relationship of the left ventricle over a complete beat of the cardiac cycle.

3.4 Passive Mechanical Behaviour of Myocardium

During diastolic filling, the myocardium of the left ventricle behaves in a purely passive manner. The following aspects of passive myocardium mechanical behaviour shall be elaborated on in the following sections:

- Incompressibility.
- Residual stresses.
- Visco-elastic response.
- Anisotropic mechanical behaviour.

Furthermore, it shall be possible to outline the constitutive model, that is utilized in the framework of this computation study, to simulate the passive mechanical behaviour of the myocardium.

3.4.1 Incompressibility

Work by Vossoughi et al. [39], showed that myocardium can be regarded as nearly incompressible materials. This conclusion was reached by subjecting tissue samples to varying levels of hydrostatic stress, and recording the associated volumetric strains.

Therefore, it is important that near incompressibility is enforced within the chosen constitutive model, used to simulate the behaviour of passive myocardium.

3.4.2 Residual Stresses

Omens and Fung [40], showed that unloaded healthy myocardium are residually stressed. These residual stresses exist normal to the fibre direction, and to a lesser extent along the muscle fibre direction [9]. In addition, these residual stresses are thought to arise during growth and remodelling, and ultimately affect the stress pattern in the typical physiological state [8, 41].

Residual stresses can be quantified by making a cut in the myocardium, transverse to the fibres. The result is that the tissue opens up, with the new configuration having lower residual stresses than the original one (see Fig. 3.9). However, these residual stresses are very hard to quantify, and will be ignored for the purpose of this study (Holzapfel and Ogden [9]).

Despite the fact that residual stresses are ignored, residual strains are accounted for in the active stress model, which will be explored later in this chapter. The effect of residual strains on the myocardium is taken into account as the sarcomere rest length, in the unloaded configuration, shows a transmural variation with smaller values at the endocardium and higher values at the epicardium [8].

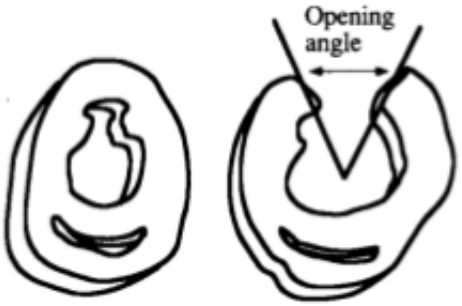

FIGURE 3.9: Illustration of the effect of residual stresses in the rat ventricle [8].

3.4.3 Visco-elastic response

Holzapfel and Ogden [9] argued that healthy myocardium tissue does possess a visco-elastic response. However, the authors mentioned that this aspect of tissue behaviour is not important from the point of view of mechanical modelling on the time scale of the

cardiac cycle, which appears to be much shorter than the relaxation time required for visco-elastic response. Visco-elastic response is therefore neglected in terms of passive mechanical modelling of the myocardium.

3.4.4 Anisotropic Mechanical Behaviour

Dokos et al. [32] described healthy myocardial tissue as being an orthotropic material, comprising of three mutually orthogonal planes, each with distinct material responses. This conclusion was drawn from observations on simple shear tests, conducted on passive myocardium of pig hearts. This orthotropic behaviour is a product of the microscopic make up of ventricular myocardium. To elaborate, a block of myocardium tissue can be isolated from the ventricle wall (see figure 3.10), the former is composed of layers of parallel myocytes, which are the predominant fibre types [9]. The remaining volume consists of various interstitial components [9]. By observing the microscopic make up of ventricular myocardium, it is clear that three mutually orthogonal directions are present, namely, a sheet axis (**s**), fibre axis (**f**) and sheet-normal axis (**n**). Each of these three orthogonal material directions posses a distinct material response, i.e., orthotropic material behaviour. The right handed orthogonal coordinate system illustrated in Fig. 3.10, will be used as the local coordinate system throughout this chapter.

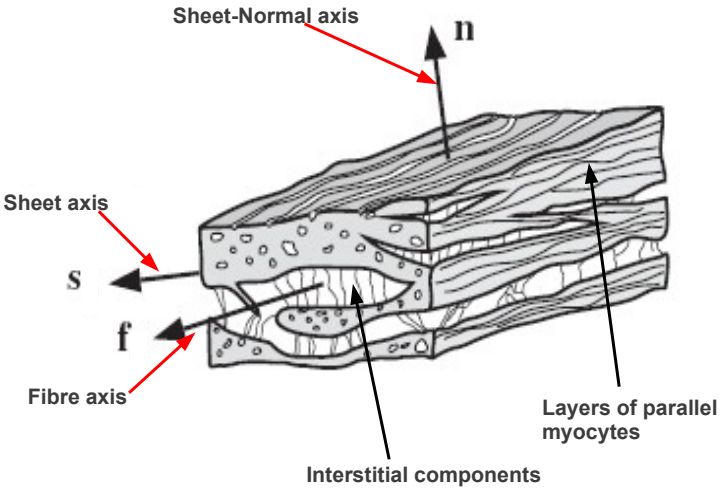

FIGURE 3.10: Layered organization of myocytes and collagen fibres between the sheets referred to by a right handed orthonormal coordinate system, with fibre axis (**f**), sheet axis (**s**) and sheet-normal axis (**n**) [9].

3.4.5 Constitutive Model For Passive Myocardium

From the previous sections, it is clear that passive ventricular myocardium is nearly incompressible and behaves in an orthotropic manner. In order to capture this passive mechanical behaviour of ventricular myocardium, the mechanical model proposed by Usyk et al. [28] is considered. Their model describes the myocardium as orthotropic and nearly incompressible, by using the strain energy function given by

$$\begin{aligned}\psi =&\frac{A}{2}\left(\exp^{Q}-1\right)\\ &+A_{\text{comp}}\left(\det\left[\mathbf{U}\right]\ln\left(\det\left[\mathbf{U}\right]\right)-\det\left[\mathbf{U}\right]+1\right),\end{aligned}\tag{3.1}$$

where A is a stiffness factor and A_{comp} is a constant controlling the compressibility of myocardium. In addition, $\mathbf{U}$ refers to the right stretch tensor and Q is defined as

$$\begin{aligned}Q =&b_{ff}\mathrm{E}_{ff}^2+b_{ss}\mathrm{E}_{ss}^2+b_{nn}\mathrm{E}_{nn}^2+b_{fs}(\mathrm{E}_{fs}^2+\mathrm{E}_{sf}^2)\\ &+b_{fn}(\mathrm{E}_{fn}^2+\mathrm{E}_{nf}^2)+b_{sn}(\mathrm{E}_{sn}^2+\mathrm{E}_{ns}^2).\end{aligned}\tag{3.2}$$

Next, E_{ij} refers to the components of the Green strain tensor, $\mathbf{E}$, defined in the local coordinate system where $i,j\in(f,s,n)$. Also, b_{ij} are material parameters. Furthermore, the Green strain tensor is defined in terms of the local material directions as

$$\mathrm{E}_{ij}=\mathrm{tr}\left(\mathbf{E}\left(\mathbf{V}_i\otimes\mathbf{V}_j\right)\right),\tag{3.3}$$

where, $\mathbf{V}_i$ and $\mathbf{V}_j$ are the material directions in the i^{th} and j^{th} directions respectively. Next, with the structural tensors referring to the three local material directions given as

$$\mathbf{M}_f=\mathbf{V}_f\otimes\mathbf{V}_f,\tag{3.4}$$

$$\mathbf{M}_s=\mathbf{V}_s\otimes\mathbf{V}_s,\tag{3.5}$$

$$\mathbf{M}_n=\mathbf{V}_n\otimes\mathbf{V}_n,\tag{3.6}$$

the term Q in Eq. 3.2 can be redefined in terms of the principle invariants of the Green strain tensor, $\mathbf{E}$, as

$$Q=a_1I_4^2+a_2I_6^2+a_3I_8^2+a_4I_5+a_5I_7+a_6I_9,\tag{3.7}$$

with

$$I_4 = \mathrm{tr}\left(\mathbf{M}_f\mathbf{E}\right), \tag{3.8}$$
$$I_5 = \mathrm{tr}\left(\mathbf{M}_f\mathbf{E}^2\right), \tag{3.9}$$
$$I_6 = \mathrm{tr}\left(\mathbf{M}_s\mathbf{E}\right), \tag{3.10}$$
$$I_7 = \mathrm{tr}\left(\mathbf{M}_s\mathbf{E}^2\right), \tag{3.11}$$
$$I_8 = \mathrm{tr}\left(\mathbf{M}_n\mathbf{E}\right), \tag{3.12}$$
$$I_9 = \mathrm{tr}\left(\mathbf{M}_n\mathbf{E}^2\right). \tag{3.13}$$

Eq. (3.7) can now be expressed in terms of the components of the Green strain tensor, $\mathbf{E}$, in the local coordinate system as

$$\begin{aligned} Q = &\left(a_1 + a_4\right)\mathrm{E}_{ff}^2 + \left(a_2 + a_5\right)\mathrm{E}_{ss}^2 + \left(a_3 + a_6\right)\mathrm{E}_{nn}^2 \\ &+ \frac{1}{2}\left(a_4 + a_5\right)\left(\mathrm{E}_{fs}^2 + \mathrm{E}_{sf}^2\right) \\ &+ \frac{1}{2}\left(a_4 + a_6\right)\left(\mathrm{E}_{fn}^2 + \mathrm{E}_{nf}^2\right) \\ &+ \frac{1}{2}\left(a_5 + a_6\right)\left(\mathrm{E}_{sn}^2 + \mathrm{E}_{ns}^2\right). \end{aligned} \tag{3.14}$$

Thus, equating like terms in Eq. (3.2) and Eq. (3.14), the newly defined material parameters, a_i (where $i \in (1, ..., 6)$), can be formulated in terms of the parameters, b_{jk} (where $j, k \in (f, s, n)$), yielding,

$$a_1 = b_{ff} - b_{fs} - b_{fn} + b_{sn}, \tag{3.15}$$
$$a_2 = b_{ss} - b_{fs} + b_{fn} - b_{sn}, \tag{3.16}$$
$$a_3 = b_{nn} + b_{fs} - b_{fn} - b_{sn}, \tag{3.17}$$
$$a_4 = b_{fn} + b_{fn} - b_{sn}, \tag{3.18}$$
$$a_5 = b_{fn} - b_{fn} + b_{sn}, \tag{3.19}$$
$$a_6 = -b_{fn} + b_{fn} + b_{sn}. \tag{3.20}$$

In Chap. 2, the second Piola-Kirchoff stress tensor, $\mathbf{S}$, was presented as a function of the nine invariants of the Green strain tensor, $\mathbf{E}$ (Eq. (2.66)). During diastolic filling, the total stress represented by the second Piola-Kirchoff stress tensor, has contributions from only the passive constitutive model, $\mathbf{S} = \mathbf{S}_\mathrm{P}$, where $\mathbf{S}_\mathrm{P}$ denotes the passive contribution.

Using Eq. (2.66), the total stress, $\mathbf{S}$, is derived as

$$\begin{aligned}\mathbf{S} =&\mathbf{S}_{\mathrm{P}} = \sum_{\alpha=1}^{\alpha=9} \frac{\partial \psi}{\partial I_\alpha} \frac{\partial I_\alpha}{\partial \mathbf{E}} \\ =&\frac{1}{2} A \exp^{Q} \frac{\partial Q}{\partial \mathbf{E}} \\ & A_{\mathrm{comp}} \left(\left[\frac{\partial}{\partial \mathbf{E}} \left(\det \left[\mathbf{U} \right] \right) \right] \ln \left(\det \left[\mathbf{U} \right] \right) + \frac{1}{\det \left[\mathbf{U} \right]} \frac{\partial}{\partial \mathbf{E}} \left(\det \left[\mathbf{U} \right] \right) - \frac{\partial}{\partial \mathbf{E}} \left(\det \left[\mathbf{U} \right] \right) \right) \\ =&\frac{1}{2} A \exp^{Q} \frac{\partial Q}{\partial \mathbf{E}} + A_{\mathrm{comp}} \left(\det \left[\mathbf{U} \right] \ln \left(\det \left[\mathbf{U} \right] \right) - \det \left[\mathbf{U} \right] + 1 \right) \left(\mathbf{C} \right)^{-1}, \end{aligned} \tag{3.21}$$

where $\mathbf{C}$ is the right Cauchy-Green deformation tensor and the tensor, $\frac{\partial Q}{\partial \mathbf{E}}$ is given by

$$\begin{aligned}\frac{\partial Q}{\partial \mathbf{E}} =&2a_1 I_4 \mathbf{M}_f + 2a_2 I_6 \mathbf{M}_s + 2a_3 I_8 \mathbf{M}_n \\ &+ a_4 \left(\mathbf{M}_f \mathbf{E} + \mathbf{E} \mathbf{M}_f \right) + a_5 \left(\mathbf{M}_s \mathbf{E} + \mathbf{E} \mathbf{M}_s \right) + a_6 \left(\mathbf{M}_n \mathbf{E} + \mathbf{E} \mathbf{M}_n \right), \end{aligned} \tag{3.22}$$

where $\mathbf{M}_i$ are the so called structural tensors in the i^{th} material direction, given by $\mathbf{M}_i = \mathbf{V}_i \otimes \mathbf{V}_i$.

3.5 Active Mechanical Behaviour of Myocardium

In Sec. 3.3, cardiac function was outlined by observing the behaviour of the myocardium of the left ventricle over the four phases of a complete beat of a cardiac cycle, namely, diastolic filling, isovolumetric contraction, ejection and isovolumetric relaxation. During the last three phases, a time varying active contraction of cardiac tissue in the ventricle is present. This active contraction is on the rise during isovolumetric contraction, causing the cavity pressure to rise in order to maintain static equilibrium and cavity volume. The active contraction increases for the greater part of ejection, but at some point, begins to decrease such that the ventricle relaxes during isovolumetric relaxation.

In order to select a constitutive model to describe the active tension development, it is important to understand the microscopic processes that occur within the myocardium. In Sec. 3.2.2.2, the electro-mechanics of the myocardium, that are initiated at the microscopic level were explored. To summarize this section, tension development is initiated when an electrical impulse passes over the sarcomere, causing ion channels to open, resulting in an increase of intra-cellular calcium concentration. This increase causes a cyclic binding of myosin heads with actin and subsequent sliding and overlapping of the myofilaments. Due to this sliding and overlapping, contraction of myocytes occurs. Consequently, the constitutive model employed, to model the active contraction of myocardium, must take into consideration the following:

- Firstly, an most importantly, intra-cellular calcium concentration.
- Sarcomere length.
- Time.

Since active contraction is initiated on the microscopic level, its development throughout the ventricle with respect to time is asynchronous. In order to simulate the asynchronous tension development in the ventricle, the chosen active tension model, can be coupled to a mathematical model of the electrical wave propagation, via a bi-domain model [27] or the Eikonal diffusion equation [17]. Using the bi-domain model or Eikonal diffusion equation, the timing of active contraction initiation can be calibrated such, that it appears physiologic, however, myofiber strain distribution in the ventricle, maybe unphysiologically inhomogeneous [30]. However, when the initiation of active contraction is assumed to be unphysiologically synchronous, then, the distribution of myofiber strain in the ventricle, can appear more homogeneous and physiologic [30]. Hence, as an initial starting point, it is reasonable to assume that the active contraction in the ventricle, occurs in a synchronous manner, even though this assumption is not physiologically true [17, 29]. For the purposes of this research, the active tension model proposed by Guccione et al. [29] is considered. The model was derived from bio-physical considerations of calcium dynamics and cross-bridge formation, and was developed as an approximation to a more complex model based on experimental data [29].

3.5.1 The Active Tension Model of Guccione et al [1993]

The active tension model by Guccione et al. [29] predicts the tension development, T_A, as

$$T_A = T_{\max} \frac{Ca_0^2}{Ca_0^2 + ECa_{50}^2(l)} C_t(l,t), \tag{3.23}$$

where, C_t is the time transient, which is dependant on time and sarcomere length (t and l respectively). Also, $T_{\max}$ is the maximum tension developed. In addition, ECa_{50} represents the calcium concentration at which tension is 50 percent. Furthermore, the time transient, C_t, is given by

$$C_t = \frac{1}{2}\left(1 - \cos\omega(l,t)\right). \tag{3.24}$$

Next, ECa_{50} is computed as

$$ECa_{50}(l) = \begin{cases} \frac{[Ca_0]_{\max}}{\sqrt{\exp[B(l-l_0)]-1}} & \text{for } l > l_0, \\ 0 & \text{for } l \leq l_0. \end{cases} \tag{3.25}$$

In Eq. (3.25), B is a constant that governs the shape of the peak isometric tension-sarcomere length relation. Also l_0 is the sarcomere length, below which no active tension develpment occurs. The sarcomere length, l, is a function of the fibre strain and the sarcomere-rest length, l_R, in the stress free configuration, and is computed as

$$l = l_R\sqrt{2\mathrm{E}_{ff}+1}, \tag{3.26}$$

where the fibre strain is computed as $\mathrm{E}_{ff} = \mathrm{tr}\,(\mathbf{M}_f\mathbf{E})$. Additionally, noting the formulation of the fourth invariant of the Green strain tensor, I_4, in Eq. (3.8), the sarcomere length can be expressed as

$$l = l_R\sqrt{2I_4+1}. \tag{3.27}$$

It is important to note that the sarcomere-rest length in the undeformed configuration is made to vary from a value of 1.78 μm, at the endocardium to a value of 1.91 μm ,at the epicardium (for a canine left ventricle [29]), thus accounting for residual strains in the myocardium (as explained in Sec. 3.4.2). Next, the time dependence of the time transient, C_t, is described by the term, $\omega\,(l,t)$ in Eq. (3.24). The term, $\omega\,(l,t)$ is computed by the a time dependant relationship given by

$$\omega\,(l,t) = \begin{cases} \pi\frac{t-u}{t_0} & \text{for } 0 \leq t < t_0, \\ \pi\frac{t-u-t_0}{t_r} & \text{for } 0 \leq t_0 < t < t_0+t_r, \\ 0 & \text{for } t \geq t_0+t_r, \end{cases} \tag{3.28}$$

where t_0 is the time to peak tension, u is the depolarization time and t_r is the duration of the relaxation period, the latter term described by

$$t_r = ml + b, \tag{3.29}$$

where m and b are constants relating to the slope and time intercepts, of the linear relaxation duration-sarcomere length relation, respectively. The second Piola-Kirchoff stress tensor, $\mathbf{S}$, can now be formulated. To do so, it is first acknowledged that during phases of the cardiac cycle where active contraction of the ventricle is present (i.e. isovolumetric contraction, ejection and isovolumetric relaxation), the second Piola-Kirchoff

stress tensor has passive and active contributions, i.e.

$$\mathbf{S} = \mathbf{S}_{\mathrm{P}} + \mathbf{S}_{\mathrm{A}}, \tag{3.30}$$

where, $\mathbf{S}_{\mathrm{P}}$ is the passive contribution to the second Piola-Kirchoff stress tensor (Eq. (3.21)). $\mathbf{S}_{\mathrm{A}}$ is the active contribution to the second Piola-Kirchoff stress tensor, which is given by

$$\mathbf{S}_{\mathrm{A}} = T_A \mathbf{M}_f. \tag{3.31}$$

The active stress formulation of the total stress, $\mathbf{S}$, is found by first substituting Eq. (3.24), into Eq. (3.31), then substituting the resulting equation, along with that for $\mathbf{S}_{\mathrm{P}}$ (Eq. (3.21)), into Eq. (3.30), yielding

$$\begin{aligned}\mathbf{S} =& \frac{1}{2} A \exp^{Q} \frac{\partial Q}{\partial \mathbf{E}} + A_{\mathrm{comp}} \left(\det\left[\mathbf{U}\right] \ln\left(\det\left[\mathbf{U}\right]\right) - \det\left[\mathbf{U}\right] + 1\right) \left(\mathbf{C}\right)^{-1} \\ &+ \left[T_{\max} \frac{Ca_0^2}{Ca_0^2 + ECa_{50}^2(l)} C_t(l,t)\right] \mathbf{M}_f,\end{aligned} \tag{3.32}$$

where the term, $\frac{\partial Q}{\partial \mathbf{E}}$, is given by Eq. (3.22).

3.6 The Circulatory System

Cardiac function is implicitly related to the output of blood into the circulatory system during ejection. Consequently, an understanding of the anatomy and means to model the coupled system is required.

3.6.1 Anatomy and Function

Referring to Fig. 3.11, blood travels on two different routes through the circulatory system. Firstly, de-oxygenated blood flows through the superior and inferior vena cava and enters the right atrium. When the blood pressure in the right atrium exceeds that in the right ventricle, the tricuspid valve opens and de-oxygenated blood flows into the latter. The instant the blood pressure in the right ventricle exceeds that in the pulmonary artery, the pulmonary semilunar valve opens, and the contracting right ventricle pushes blood out into the latter. The blood is then transported to the lungs so as to get oxygenated. This phase of circulation is referred to as pulmonary or venous circulation.

The pulmonary veins then return the oxygenated blood from the lungs to the left atrium. When the blood pressure in the left ventricle, falls below that in the left atrium, the

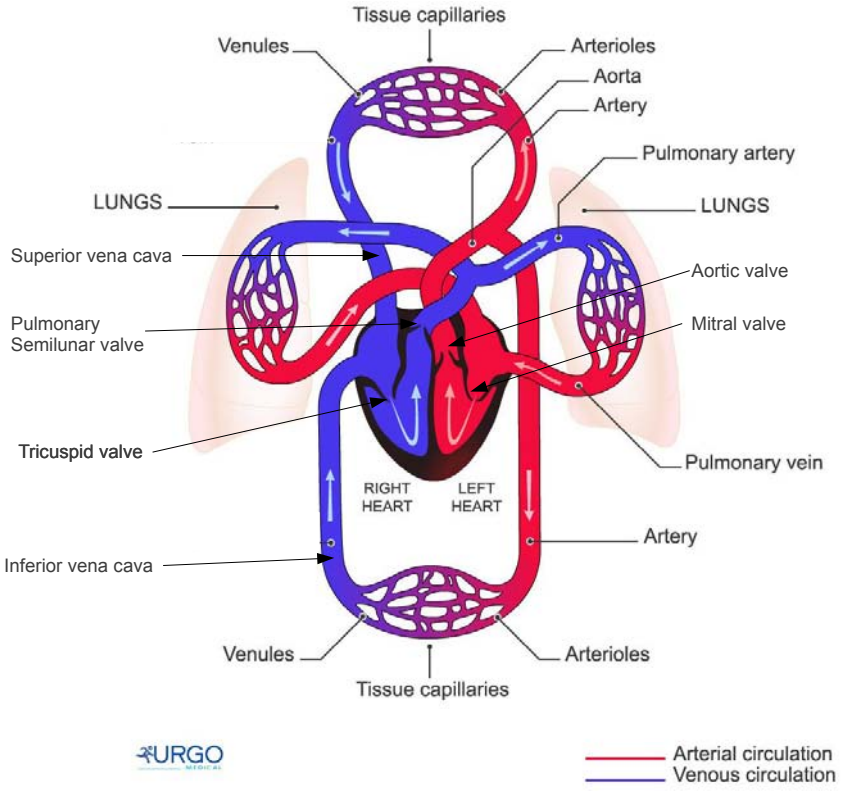

FIGURE 3.11: A schematic illustration of the circulatory sytem [10].

mitral valve opens, and oxygenated blood flows from the left atrium into the left ventricle (diastolic filling). The moment the pressure in the left ventricle exceeds the aortic pressure, the aortic valves open, and the oxygenated blood is ejected into the aorta due to the contracting ventricle (i.e. ejection). The oxygenated blood then flows through the aorta to the rest of the body. This phase of circulation is referred to as systemic or arterial circulation. De-oxygented blood from the body is then returned to the superior and inferior vena cava, signalling the start of pulmonary circulation.

In the context of cardiac mechanics modelling, systemic or arterial circulation is of particular interest, as the latter portion of this phase of circulation is coupled to the ejection phase of the cardiac cycle. Therefore, the mechanical response of the left ventricle during ejection needs to be coupled to an appropriate model for the hemodynamics of systemic circulation [17].

3.6.2 Modelling Systemic Circulation

Systemic circulation is modelled via an appropriate model of the arterial system, which can be done in various ways, using Lumped models [42], tube models [43] and anatomically based distributed models [44]. In this work, Lumped or Windkessel models will

be considered as as a means of modelling the arterial system, due to the relative ease in which they can be used to describe the load faced by the heart, in pumping blood through the systemic arterial tree.

3.6.3 Windkessel Models

Windkessel or Lumped models describe the hemodynamics of the whole arterial system, in terms of pressure-flow relations at its entrance, using Windkessel parameters that have a physiological meaning, such arterial compliance and resistance [11]. However, based on such models it is not possible to study phenomena that occur within the arterial tree, such as wave travel and reflection of waves [11]. Despite this disadvantage, the most useful attribute of Windkessel models, is that they can be readily applied to describe the load faced by the left ventricle, in pumping blood through the systemic arterial system, i.e., they can be used to describe or predict left ventricular pressure during the ejection phase of the cardiac cycle. Various Windkessel models can be found in the literature. However, the most common are the two, three and four element Windkessel models [11]. Consequently, the following sections will explore the various models available.

3.6.3.1 The Two Element Windkessel Model

The two element Windkessel model is the earliest model, developed by the German physiologist Otto Frank [45]. It models the heart and systemic arterial tree, using a closed hydraulic circuit comprising of a pump (i.e. the heart) connected to a chamber (i.e. the aorta), which is in turn connected to a resistor. Fig. 3.12 below illustrates the schematic set-up of the two-element Windkessel model. Referring to Fig. 3.12, the hydraulic circuit is filled with blood except for a pocket of air in the chamber (i.e. Windkessel). As blood is pumped into the latter (from the heart pump), the air pocket is compressed, and blood is pushed out of the chamber back into the pump. The air pocket in the chamber simulates the elasticity and extensibility of the major artery (aorta), as blood is pumped into it by the left ventricle, an effect which is commonly referred to as arterial compliance. The resistance that blood encounters while leaving the aorta and flowing back into the heart, simulates the resistance to flow that blood experiences when travelling through the arterial tree. This resistance is commonly referred to as the peripheral resistance.

The governing differential equation relating blood pressure in the aorta to aortic flow,

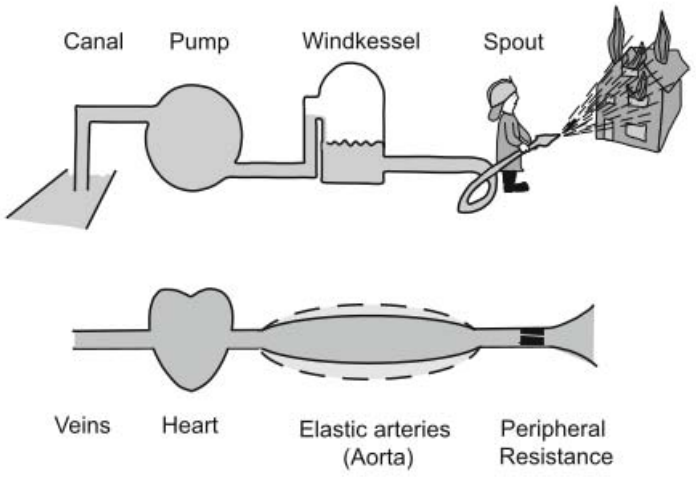

FIGURE 3.12: The concept of the two element Windkessel. The aorta acts as the chamber (i.e. Windkessel),the heart acts as a pump and the peripheral resistance represents the resistance of the arterial tree [11].

based on the two element Windkessel is given by

$$I(t)_{ao} = \frac{P(t)_{ao}}{R} + C\frac{dP(t)_{ao}}{dt}, \tag{3.33}$$

where, t refers to time, R the peripheral resistance, C the arterial compliance, $I(t)_{ao}$ the aortic flow and $P(t)_{ao}$ the aortic pressure. A similar equation governs the relationship between current, $I(t)$ and time varying potential, $V(t)$, in an electrical circuit. In light of this last remark, the two element Windkessel model can be schematically represented by the following circuit analogy of the arterial system, with an arterial compliance in parallel with a peripheral resistance, as illustrated in Fig. 3.13. During diastole, the flow of blood into the aorta is zero, i.e., $I(t)_{ao} = 0$, in which case the two element Windkessel can be solved exactly for the aortic pressure. The result is an exponential decay function for the aortic pressure described by

$$P(t)_{ao} = P_{es}\exp^{\frac{-t}{RC}}, \tag{3.34}$$

where P_{es} is the end-systolic aortic pressure. The above relationship is observed to correspond well to experimental measurements of aortic pressure during diastole, which is illustrated in Fig. 3.14.

Shortcomings of the Two Element Windkessel Model

With the development of the electromagnetic flow meter, which allowed measurement of aortic flow, it became apparent that during systole the relationship between aortic pressure and flow was poorly predicted by the two element Windkessel model [11]. This was observed by using measured aortic pressure and flow signals to compute what is referred to as the input impedance, Z_c, that varies with frequency. From these computations,

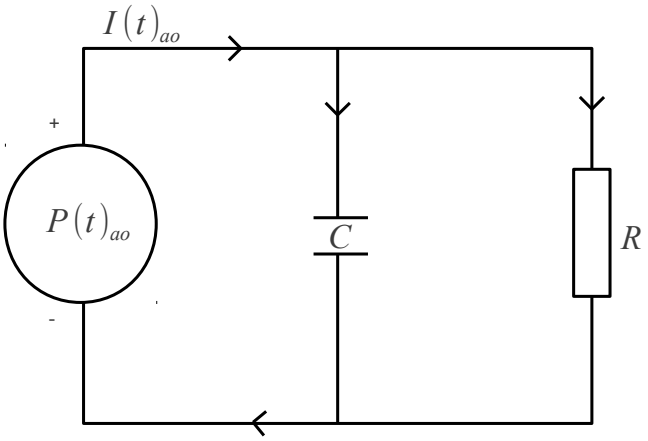

FIGURE 3.13: A closed hydraulic circuit analogy of arterial system based on the two element Windkessel model [11].

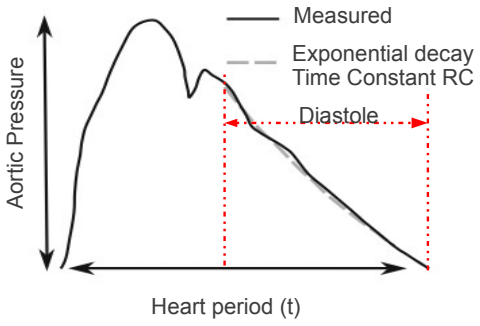

FIGURE 3.14: The exponential decay of aortic pressure during diastole predicted by the two element Windkessel (grey dotted lines) and that observed experimentally (solid black lines) [11].

the reasons for the shortcoming of the two element Windkessel was unveiled. For high frequencies, it is observed that the input impedance modulus reduces to negligible values and its phase angle reaches -90 degrees, whereas for mammals the measured input impedance modulus decreases to a plateau and its phase angle hovers around zero for higher frequencies [11] (see Fig. 3.15). Referring to Fig. 3.15, the more or less constant

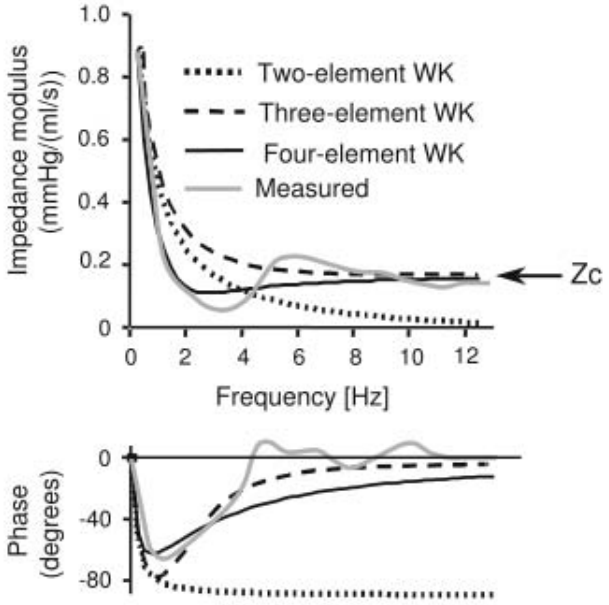

FIGURE 3.15: Illustration of the short comings of the two-element Windkessel model [11]. The top figure illustrates the input impedance modulus measured for various frequencies as well as that predicted by various Windkessel models. The bottom figure illustrated the measured phase angle at various frequencies observed experimentally as well as that predicted by various Windkessel models.

level of input impedance at higher frequencies was found to be equal to the characteristic impedance of the proximal aorta [11]. This information resulted in the addition of the characteristic impedance to the two element Windkessel model, forming what is commonly referred to as the three element Windkessel.

3.6.3.2 The Three Element Windkessel Model

The three element Windkessel model adds a characteristic impedance, Z_c, to the two element Windkessel model, between the pump and the air chamber (see Fig. 3.16). This characteristic impedance can be seen as the link between the Lumped Windkessel model and wave travel aspects of the arterial system [11]. The characteristic impedance has the same dimensions as resistance and can be replaced by a resistive element, between the pump and the chamber [11]. This additional resistor is included to capture resistance to blood flow, due to the aortic or pulmonary valve and allows the model to capture

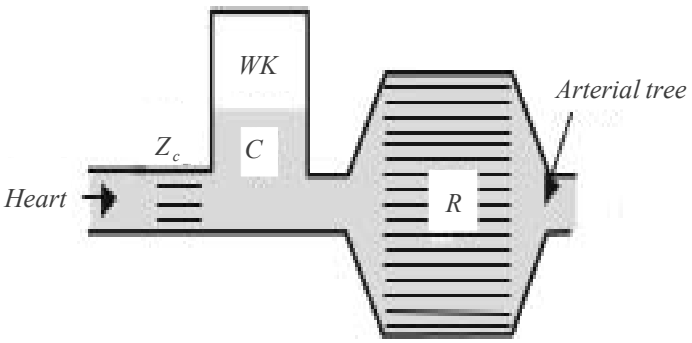

FIGURE 3.16: The concept of the three element Windkessel [11].

wave effects of the arterial system [11]. The additional resistor is referred to as a flow resistance. Fig. 3.17 below is an illustration of the closed hydraulic circuit analogy. of the three element Windkessel model, where the flow resistance, R_0, is in series with the peripheral resistance. Based on the above circuit analogy, the governing differential

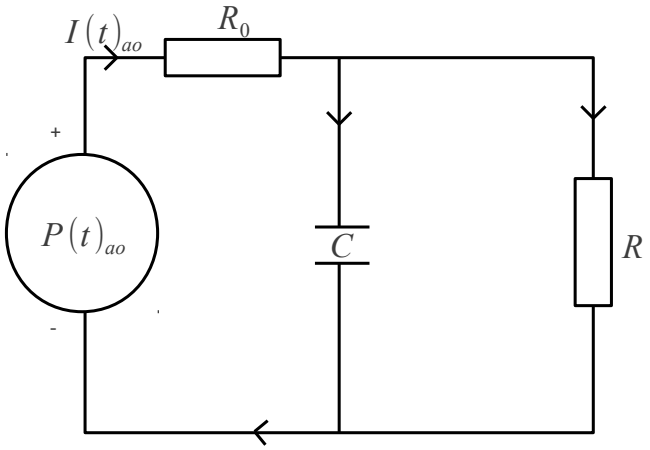

FIGURE 3.17: A closed hydraulic circuit analogy of arterial system based on the three-element Windkessel model [11].

equation for the three element Windkessel model is given by

$$(CR_0)\frac{dI_{ao}}{dt} + \left(1 + \frac{R_0}{R}\right) I_{ao} = (C)\frac{dP_{ao}}{dt} + \left(\frac{1}{R}\right) P_{ao}, \tag{3.35}$$

where,

- C is the arterial compliance,
- R is the peripheral resistance,
- R_0 is the flow resistance,
- P_{ao} is again the aortic pressure,
- and I_{ao} is the aortic flow rate.

During diastole, the aortic flow is zero. Thus, the three element Windkessel model can be solved to find an explicit form for the aortic pressure given by

$$P\left(t\right)_{ao} = P_{es}\exp^{\frac{-t}{RC}}. \tag{3.36}$$

Therefore, it is observed that Eq. (3.36) predicts the exponential decay of aortic pressure during diastole, in exactly the same manner as the two element Windkessel model.

Shortcomings of the Three-Element Windkessel Model

Referring to Fig. 3.15, it is noticed that the three element Windkessel model over predicts the impedance modulus in the low frequency range [11]. In order to reduce these errors, a fourth element to the Windkessel model was proposed by Stergiopulos et al. [46]. This fourth element is an inertance, which is equal to the addition of all inertances in the arterial segments, i.e., the total arterial inertance [11]. This inertance term only affects the very low frequency behaviour of the input impedance, which is where the three element Windkessel is inaccurate. However, in practice, it turns out that the inertance is very difficult to estimate, which is an argument to prefer the three element Windkessel [11]. In addition, using the three element Windkessel model, the overall predicted waveforms are close to the measured pressures and the global aspects of pressure and the input impedance are well described [11]. Consequently, for the purposes of this research, and following the recommendations of Westerhof et al. [11], the three element Windkessel model is chosen to model the hemodynamics of the arterial system.

Chapter 4

The Idealized Ellipsoidal Canine Left Ventricle Model

4.1 Introduction

In previous chapters, an understanding of the continuum mechanics approach used to model the stresses and deformations in the left ventricle were developed (chapter 2). In addition, the structure of myocardial tissue was explored, along with the observed behaviour of the left ventricle over a full cardiac cycle. Furthermore, the mathematical basis for a continuum mechanics approach to modelling the passive and active mechanical behaviour of the myocardium, was explained (chapter 3).

In order to properly simulate left ventricular mechanical function over a full cardiac cycle, its geometry must be modelled correctly. For the purposes of this research, an idealized truncated prolate ellipsoidal approximation of a canine left ventricle is utilized [17]. Furthermore, a key component for simulating the mechanics of the myocardium of the left ventricle, is the modelling of the fibrous nature of the tissue, which gives rise to its anisotropic mechanical behaviour [13]. Consequently, this chapter will explore the geometry and coordinate system of the idealized ellipsoidal canine left ventricle model. Additionally, the manner in which the fibrous nature of myocardium is modelled will be outlined.

4.2 Model Geometry

The parameters used to model the idealized canine ellipsoidal geometry are summarized in Tab. 4.1 below and are in accordance with Bovendeerd et al. [12] and Kerckhoffs

et al. [17].

TABLE 4.1: Model Parameters for an Idealized Ellipsoidal Canine Left Ventricle Geometry

Parameter	Values
$H(mm)$	24.8
$C(mm)$	43.0
η_{epi}	0.68
η_{endo}	0.37

In Tab. 4.1, H refers to the truncation height, which is the height from the centre of the ellipsoid to the base. Also, C is the prolate constant and η_{epi} and η_{endo}, are the parameters that define the epicardial and endocarial surface coordinates.

It is convenient to express the Cartesian coordinates of the ellipsoidal model geometry, in terms of prolate spheroid coordinates, η, θ and ϕ, which are part of a curvilinear coordinate system [47]. The relation between Cartesian and prolate spheroid coordinates is given by

$$x = C\sinh(\eta)\sin(\theta)\cos(\phi), \tag{4.1}$$

$$y = C\sinh(\eta)\sin(\theta)\cos(\phi), \tag{4.2}$$

$$z = C\cosh(\eta)\sin(\theta), \tag{4.3}$$

where, η is equal to η_{epi} for the epicardium and η_{endo} for the endocardium. Additionally, $\theta \in (0,\, \pi)$, $\phi \in (0,\, 2\pi)$ and x, y and z are the usual Cartesian coordinates. Fig. 4.1(a) illustrates the truncation height, H, the prolate constant, C and the prolate spheroid coordinate, ϕ, all on a Cartesian coordinate system (x, y, z). Furthermore, Fig. 4.1(b) illustrates the variation of the prolate spheroid coordinate, θ and the values of the prolate spheroid coordinate, η, at the endocardium and epicardium surfaces (η_{endo} and η_{epi} respectively).

Referring to Fig. 4.1, for the half plane, x-y, that passes through the origin (O), ϕ is set equal to zero, an θ is equal to $\frac{\pi}{2}$. Hence, the maximum radius, x, on this plane can be computed for the endocardium, with η equal to 0.37 as

$$x = C\sinh(\eta) = 16.28\,\text{mm}. \tag{4.4}$$

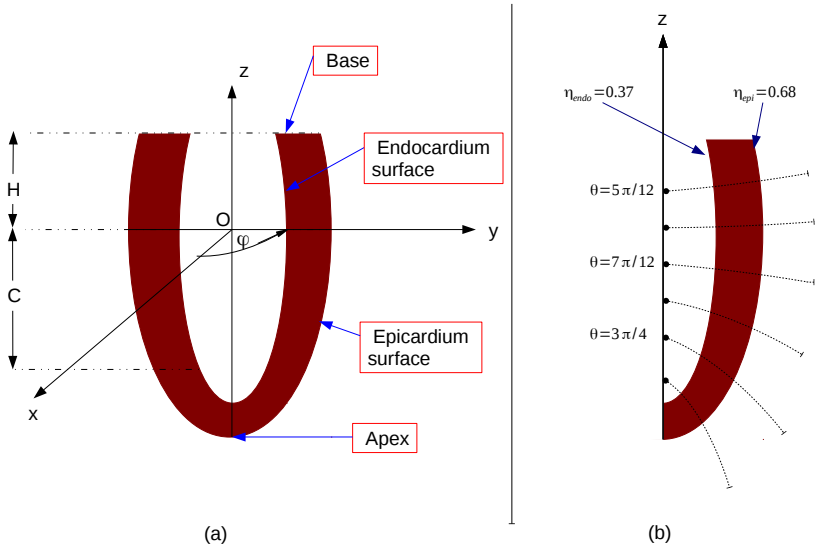

FIGURE 4.1: (a)Illustration of parameters C and truncation height H as well as the prolate spheroid coordinate ϕ, which are in the idealized ellipsoidal canine left ventricle geometry [12]. (b)Illustration of the variation of the prolate spheroid coordinate θ, and the values of the prolate spheroid coordinate η at the endocardium and epicardium surfaces [12].

Similarly, the maximum radius, x, can be computed for the epicardium, with η equal to 0.68 as

$$x = C\sinh(\eta) = 31.55\,\text{mm}. \tag{4.5}$$

Next, for the x-z plane, θ is equal to zero, consequently, the maximum height, z, from the origin to the apex can be computed for the endocardium, with η equal to 0.37 as

$$z = C\cosh(\eta) = 43.98\,\text{mm}. \tag{4.6}$$

Also, the maximum height, z, from the origin to the apex can be computed for the epicardium, with η equal to 0.68 as

$$z = C\cosh(\eta) = 53.33\,\text{mm}. \tag{4.7}$$

Now, considering the general equation for an ellipse in Cartesian coordinates,

$$\left(\frac{x}{\text{b}}\right)^2 + \left(\frac{y}{\text{c}}\right)^2 = 1, \tag{4.8}$$

it is clear that for the endocardium surface, b and c are 16.28 mm and 43.98 mm, respectively (as calculated in Eq. (4.4) and Eq. (4.6), respectively). Similarly, for the epicardium surface, b and c are 31.55 mm and 53.33 mm, respectively (as calculated in

Eq. (4.5) and Eq. (4.7), respectively). The values of b and c for the endocardium and epicardium surfaces are summarized in Tab. 4.2.

TABLE 4.2: Ellipsoidal constants b and c for endocardium and epicardium surfaces

Location	b(mm)	c(mm)
Endocardium	16.28	43.98
Epicardium	31.55	53.33

Consequently, if we allow the term, $\left(\frac{x}{\mathrm{b}}\right)^2$, in Eq. (4.8) to be equal to t, where $t \in (0,1)$, then we get that x is equal to $\mathrm{b}\sqrt{t}$ and z is equal to $\mathrm{c}\sqrt{1-t}$. Thus, the parametric equation representing either the endocardium or epicardium surfaces is

$$f(t) = (\mathrm{b}\sqrt{t}, 0, \mathrm{c}\sqrt{1-t}). \tag{4.9}$$

4.2.1 Model Generation

Making use of the parametric equation derived in the previous section (Eq. (4.9)), the truncated canine ellipsoidal model geometry, is generated using the pre-processing software GID (http://www.gid-usa.com). Fig. 4.2 illustrates the three dimensional model generated in GID, that is utilized for all mechanics simulations of the canine left ventricle.

FIGURE 4.2: Three dimensional representation of the truncated ellipsoidal canine left ventricle generated in GID.

4.2.2 Auxiliary Coordinate System

It is convenient to utilize the prolate spheroid coordinate system outlined in the previous section, to define an auxiliary coordinate system that describes both the transmural and longitudinal positions in the left ventricle model [13]. This auxiliary coordinate system is illustrated diagrammatically in Fig. 4.3.

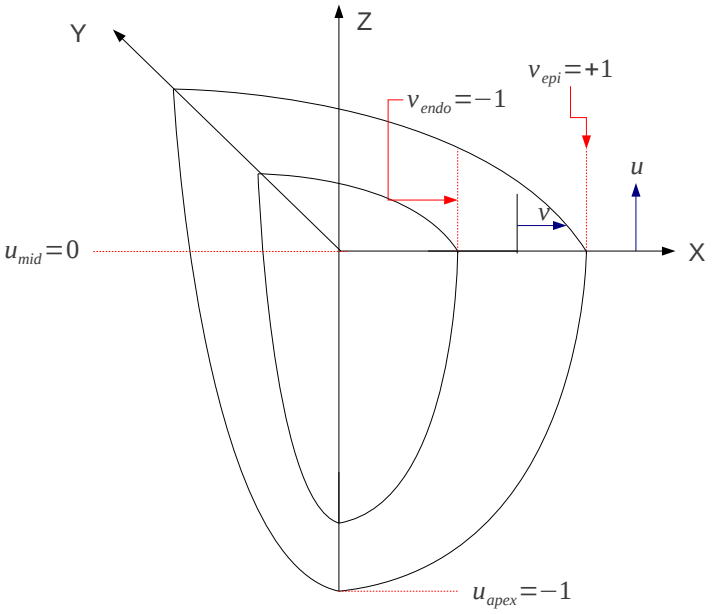

FIGURE 4.3: Diagrammatic representation of the auxiliary coordinate system used.

Referring to Fig. 4.3, it is clear that the transmural coordinate, v, is made to vary monotonically from a value of $v_{\text{endo}} = -1$ at the endocardium, to a value of $v_{\text{epi}} = +1$ at the epicardium. Similarly, the longitudinal coordinate, u, is made to vary monotonically from a value of $u_{\text{apex}} = -1$ at the apex, to a value $u_{\text{mid}} = 0$ at the equator.

Using the following relations, the transmural prolate spheroid coordinate, η, and the longitudinal prolate spheroid coordinate, θ, are shifted so that they are centred on zero. Then coordinates are scaled so that they vary from -1 to +1 in the correct direction

$$v = \bar{\eta} = \eta - \eta_{\text{mid}} \frac{\bar{\eta}_{\text{diff}}}{\eta_{\text{diff}}}, \tag{4.10}$$

where

$$\eta_{\text{diff}} = \eta_{\text{epi}} - \eta_{\text{endo}}, \tag{4.11}$$

$$\bar{\eta}_{\text{diff}} = \bar{\eta}_{\text{epi}} - \bar{\eta}_{\text{endo}} = 2, \tag{4.12}$$

$$\eta_{\text{mid}} = \frac{\eta_{\text{epi}} + \eta_{\text{endo}}}{2}. \tag{4.13}$$

In addition,

$$u = \bar{\theta} = \theta - \theta_{\text{mid}} \frac{\bar{\theta}_{\text{diff}}}{\theta_{\text{diff}}} = -\left(\theta - \frac{\pi}{2}\right) \frac{2}{\pi}, \tag{4.14}$$

where

$$\theta_{\text{diff}} = \theta_{\text{apex}} - \theta_{\text{base}} = \pi, \tag{4.15}$$

$$\bar{\theta}_{\text{diff}} = \bar{\theta}_{\text{apex}} - \bar{\theta}_{\text{base}} = -2, \tag{4.16}$$

$$\eta_{\text{mid}} = 2. \tag{4.17}$$

From the above equations, it is clear that η_{epi} and η_{endo} are required as inputs for the truncated ellipsoidal left ventricle model, of the animal being studied. Also, it should be noted that the new coordinates, vary monotonically rather than linearly with transmural and longitudinal distance.

4.3 Fibre Orientations

For the ellipsoidal model considered in this study, the orientation of cardiac muscle fibres, is quantified by the helix fibre angle, α_h and the transverse fibre angle, α_t [13] (refer to Fig. 4.4).

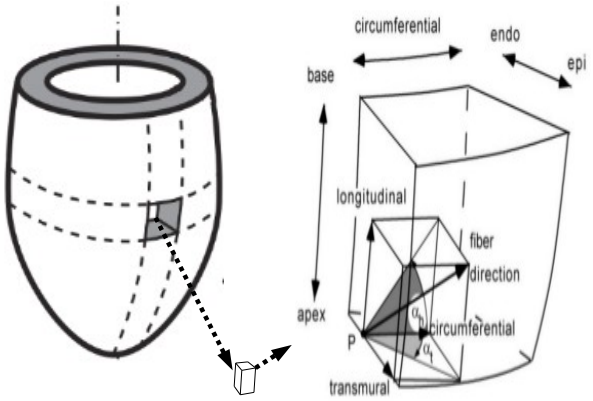

FIGURE 4.4: Quantification of muscle fibre orientation by helix fibre andge α [12]

The helix fibre angle, α_h and the transverse fibre angle, α_t, vary according to

$$\alpha_h(v) = p_1 + p_2 v, \tag{4.18}$$

$$\alpha_t(u, v) = p_3 u(1 - v^2), \tag{4.19}$$

where, u and v are the normalized longitudinal and transmural coordinates respectively, as defined in the previous section. Also, p_1, p_2 and p_3 are constants defined in the Tab. 4.3 [13].

TABLE 4.3: Parameter values for fibre direction model [13]

p_1(rad)	p_2(rad)	p_3(rad)
0.3533429174	-1.147553983	0.291993585

The constants in Tab. 4.3 were optimized by Rijcken et al. [13], in order to produce a near homogeneous distribution of fibre strain during ejection, by minimizing the an objective function of the form

$$G(\mathbf{p}) = \text{variance}(\epsilon_{\text{be}} + \epsilon_{\text{ee}}) + w \cdot \text{variance}(\epsilon_{\text{be}} - \epsilon_{\text{ee}}), \tag{4.20}$$

where ϵ_{be} and ϵ_{ee} are fibre strains at the beginning and end of ejection. In addition w is a weighting factor. Via the minimization of Eq. (4.20), Rijcken et al. [13] arrived at a set of parameters for p_1, p_2 and p_3 (refer to Tab. 4.3), which yield fibre orientations, $\mathbf{V}_f$, that produce a near homogeneous distribution of the average fibre strain between the beginning and end of ejection, which is an observation that the authors repute to be physiologically observed. Using the constants p_1, p_2 and p_3, the variation of the helix and transverse fibre angles, due to variations in the normalized longitudinal and transmural coordinates (u and v), can be found. Fig. 4.5 illustrates the model fibre direction angles (solid lines, α_h and α_t) as a function of normalized longitudinal and transmural position. Fig. 4.5 also illustrates the experimental helical fibre angle [48, 49]. By observing Fig. 4.5, it is evident that the model value corresponds well with the experimental.

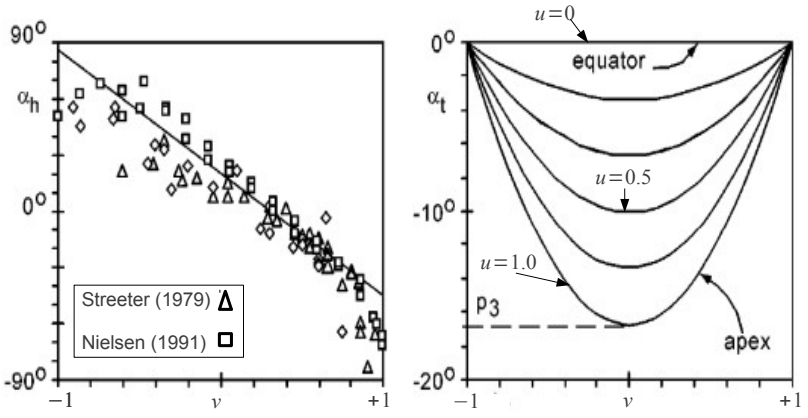

FIGURE 4.5: Quantification of muscle fibre orientation by helix fibre andge α [13]

In order to compute the fibre direction at an arbitrary point within the left ventricle model, three basis vectors need to be defined for the circumferential, transmural and longitudinal directions, i.e., $\boldsymbol{\phi}$, $\boldsymbol{\eta}$ and $\boldsymbol{\theta}$, respectively. These basis vectors are given by

$$\begin{aligned}\boldsymbol{\eta} =& \frac{1}{\sqrt{\cosh^2\eta\sin^2\theta + \sinh^2\eta\cos^2\theta}} \\ & \times (\cosh\eta\sin\theta\cos\phi\mathbf{e}_1 + \cosh\eta\sin\theta\sin\phi\mathbf{e}_2 + \sinh\eta\cos v\mathbf{e}_3), \qquad (4.21)\\ \boldsymbol{\phi} =& \frac{1}{\sqrt{\sinh\eta\sin\theta}} \\ & \times (-\sinh\eta\sin\theta\sin\phi\mathbf{e}_1 + \sinh\eta\sin\theta\cos\phi\mathbf{e}_2 + 0\mathbf{e}_3), \qquad (4.22)\\ \boldsymbol{\theta} =& \frac{-1}{\sqrt{\cosh^2\eta\sin^2\theta + \sinh^2\eta\cos^2\theta}} \\ & \times (\sinh\eta\cos\theta\cos\phi\mathbf{e}_1 + \sinh\eta\cos\theta\sin\phi\mathbf{e}_2 + \cosh\eta\sin v\mathbf{e}_3). \qquad (4.23)\end{aligned}$$

The fibre direction, $\mathbf{V}_f$, at any point within the left ventricle, can then be computed as

$$\mathbf{V}_f = \tan\alpha_t\boldsymbol{\eta} + \tan\alpha_h\boldsymbol{\phi} + \boldsymbol{\theta}. \qquad (4.24)$$

Fig. 4.6, shows the projected fibre directions at various points throughout the idealized ellipsoidal canine left ventricle.

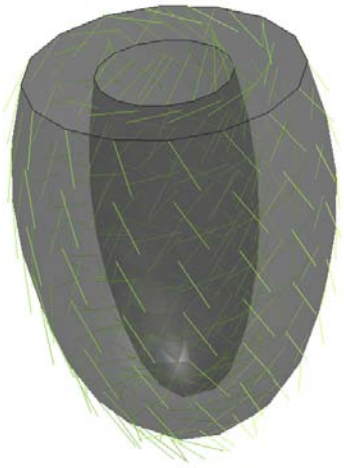

FIGURE 4.6: Fibre directions at various points throughout the canine ellipsoidal left ventricle geometry

The remaining material directions, $\mathbf{V}_s$ and $\mathbf{V}_n$, in the sheet and sheet-normal directions respectively, are computed as

$$\mathbf{V}_n = \mathbf{V}_f \times \boldsymbol{\eta}, \qquad (4.25)$$

$$\mathbf{V}_s = \mathbf{V}_n \times \mathbf{V}_f. \qquad (4.26)$$

Chapter 5

Numerical Modelling of Cardiac Function

5.1 Introduction

Chapter four explored the idealized canine ellipsoidal left ventricle model, which will be used for numerical modelling in the context of this research. Hence, it is possible to integrate the information that was gathered in the previous chapters and use it to outline the implementation of the element-free Galerkin method (EFGM), which forms the core of the in-house modelling software SESKA. SESKA was chosen for the purposes of this work over commercial finite element packages such as ABAQUS, because the formers development as a software is the current focus of the Computational Continuum Mechanics Group (CCM) at the University of Cape Town, which is the research group that this work belongs to. The software is being developed for the purposes of studying left ventricular behaviour, along with other scientific and engineering phenomena. Hence, in summary, SESKA serves as the basis for all mechanical modelling of left ventricular function throughout this dissertation. In itself, SESKA is a cross-platform numerical modelling software, comprising of over 200,000 lines of code. The software is implemented in the C++ language, and has links to many useful external packages such as PETSc (*Portable Extensible Tool Kit*). The power of the software is its ability to run on large computing clusters, enabling the user to simulate large problems with highly refined particle distributions.

5.1.1 Implementation of the Element-Free Galerkin Method

The EFGM method is a meshless method. Thus, instead of using a mesh, as in finite element methods (FEM), a particle distribution is used to calculate the shape functions. In addition, in SESKA, the moving least squares (MLS) method is used to compute the shape functions and approximate solutions over the domain. For more information on the EFGM and MLS methods, the reader is directed to the paper by Belytschko et al. [18].

5.1.2 Outline of Topics Covered in this Chapter

In the succeeding sections of this chapter, attention will be paid to the following aspects of the EFGM method, implemented in the in-house modelling software SESKA:

- **Variational formulation:** The variational formulation (weak form) of continuum mechanics.
- **Assembly:** Firstly, the assembly of the discrete equation system can be outlined. Next, with regards to modelling cardiac function, the variations in the assembly of the discrete equation system, will be explored for each phase of the cardiac cycle, namely, diastolic filling, isovolumetric relaxation and contraction and finally ejection.
- **Implementation:** Several contributions made by the author, towards the in-house modelling software SESKA will be explained.

5.2 Variational Formulation

In chapter two, the law of linear momentum conservation was derived, and Eq. (2.37) represented the Lagrangian equilibrium equation for a constant velocity field. Eq. (2.37) represents the governing differential equation for the cardiac mechanics problem. In addition, both traction and displacement boundary conditions are necessary for realistic modelling of cardiac function. Hence, the general strong form for the cardiac mechanics problem is

$$\mathrm{div}\mathbf{P} + \mathbf{b}\left(\mathbf{X}, t\right) = \mathbf{0} \qquad \text{in } \mathcal{B}, \tag{5.1}$$

$$\mathbf{P}\mathbf{n} = \mathbf{t}^{(\mathbf{n})} \text{ on } \partial\mathcal{B}_N, \tag{5.2}$$

$$\mathbf{u} = \bar{\mathbf{u}} \text{ on } \partial\mathcal{B}_D. \tag{5.3}$$

Referring to the above strong form, the first equation represents the equilibrium equation of Newton's second law, with negligible acceleration. The second equation expresses the traction force, $\mathbf{t}^{(\mathbf{n})}$, in the Lagrangian description, acting on the Neumann boundary, $\partial\mathcal{B}_N$. Finally, the last equation expresses the displacement boundary condition, $\bar{\mathbf{u}}$, on the Dirichlet boundary, $\partial\mathcal{B}_D$.

The variational formulation (or weak form) of continuum mechanics states that the external potential, $\mathcal{W}_{\text{ext}}$, is equal to the internal potential, $\mathcal{W}_{\text{int}}$ [50]. The external potential corresponds to the work done by the external forces such as a body force, $\mathbf{b}$, acting over the volume of the body and traction forces, $\mathbf{t}^{(\mathbf{n})}$, acting on the surface of the body. The external potential is given by

$$\mathcal{W}_{ext} = \int_{\mathcal{B}} \rho_0 \mathbf{b} \cdot \delta\mathbf{u} dV + \int_{\partial\mathcal{B}_N} \mathbf{t}^{(\mathbf{n})} \cdot \delta\mathbf{u} dA, \tag{5.4}$$

where $\delta\mathbf{u}$ is the variation of displacement. The internal potential is given by

$$\mathcal{W}_{int} = \int_{\mathcal{B}} \mathbf{S} : \delta\mathbf{E} dV, \tag{5.5}$$

where $\delta\mathbf{E}$ is the variation of the Green strain tensor. The full variational formulation, which is the sum of internal and external potential can be stated as

$$\begin{aligned} \delta\Psi =& \mathcal{W}_{int} - \mathcal{W}_{ext} = 0 \\ =& \int_{\mathcal{B}} \mathbf{S} : \delta\mathbf{E} dV - \int_{\mathcal{B}} \mathbf{b} \cdot \delta\mathbf{u} dV - \int_{\partial\mathcal{B}_N} \mathbf{t}^{(\mathbf{n})} \cdot \delta\mathbf{u} dA = 0. \end{aligned} \tag{5.6}$$

It can be shown that the variational formulation in Eq. (5.6), is equivalent to the strong form. The equivalence of the strong and variational formulations, will not be illustrated in this work, but the reader is directed to the book, *Nonlinear finite element methods* [50], for further details. Furthermore, in cardiac mechanics, the body force is usually neglected. Hence, the weak form can be simplified to

$$\delta\Psi = \int_{\mathcal{B}} \mathbf{S} : \delta\mathbf{E} dV - \int_{\partial\mathcal{B}_N} \mathbf{t}^{(\mathbf{n})} \cdot \delta\mathbf{u} dA = 0. \tag{5.7}$$

Additionally, the surface traction, $\mathbf{t}^{(\mathbf{n})}$, is applied to the inner wall of the left ventricle and is given by

$$\mathbf{t}^{(\mathbf{n})} = P\mathbf{n} \quad \text{on } \partial\mathcal{B}_{tN}. \tag{5.8}$$

From Eq. (5.8), it is important to note that the pressure, P, which is exerted by blood on the endocardium wall, acts in the current configuration, denoted by $\partial\mathcal{B}_{tN}$. The corresponding term in the variational formulation is the external potential, $\mathcal{W}_{ext}$, which

under the assumption of negligible body force becomes

$$\mathcal{W}_{ext} = \int_{\partial\mathcal{B}_{tN}} P\mathbf{n} \cdot \delta\mathbf{u} da. \tag{5.9}$$

The equation above represents an integral over the Neumann boundary conditions in the current configuration. Consequently, Nanson's formula, which relates surface elements in the current and reference configuration, is used to convert the integral to one over Neumann boundary conditions in the reference configuration, yielding,

$$\mathcal{W}_{ext} = \int_{\partial\mathcal{B}_N} PJ\mathbf{F}^{-\mathrm{T}}\mathbf{n} \cdot \delta\mathbf{u} dA. \tag{5.10}$$

Consequently, the final weak form for the cardiac mechanics problem is formulated as

$$\delta\Psi = \int_{\mathcal{B}} \mathbf{S} : \delta\mathbf{E} dV - \int_{\partial\mathcal{B}_N} PJ\mathbf{F}^{-\mathrm{T}}\mathbf{n} \cdot \delta\mathbf{u} dA = 0. \tag{5.11}$$

5.3 Assembly

At the assembly stage, the objective is to obtain a discrete set of equations, that can be subsequently solved for the unknown field such as displacement. The discrete equation system usually takes the form of

$$\mathbf{K}\Delta\mathbf{u} - [\mathbf{f}_{ext} - \mathbf{f}_{int}] = \mathbf{0}, \tag{5.12}$$

where,

- $\mathbf{K}$ is the stiffness or tangent matrix,
- $\Delta\mathbf{u}$ is the unknown displacement field increment,
- $\mathbf{f}_{ext}$ is the external loading vector,
- and $\mathbf{f}_{int}$ is the internal reaction vector.

However, for non-linear problems, such as the cardiac mechanics problem considered in this work, there is no equality in Eq. (5.12), hence, an iterative procedure is required to find the unknown displacement field. This iterative procedure, involves Newton's method.

5.3.1 Newton's Method

At each iteration step, i, of a loading or time step, n, the discrete equation system is assembled from the linearisation of the weak form in Eq. (5.11). The weak form is linearised using a first order Taylor series expansion, about an initial guess for the displacement vector, $\mathbf{u}_n^{i-1}$, which yields,

$$\delta\Psi\left(\mathbf{u}\right) = \delta\Psi\left(\mathbf{u}_n^{i-1} + \Delta\mathbf{u}\right) = \delta\Psi\left(\mathbf{u}_n^{i-1}\right) + \frac{\partial\delta\Psi\left(\mathbf{u}\right)}{\partial\mathbf{u}}|_{\mathbf{u}_n^{i-1}} \cdot \Delta\mathbf{u}, \tag{5.13}$$

where $\Delta\mathbf{u}$ is the incremental displacement field such that $\mathbf{u} = \mathbf{u}_n^{i-1} + \Delta\mathbf{u}$. The linearization of $\delta\Psi\left(\mathbf{u}\right)$ (i.e. Eq. (5.13)) is then approximately zero

$$\delta\Psi\left(\mathbf{u}_n^{i-1}\right) + \frac{\partial\delta\Psi\left(\mathbf{u}\right)}{\partial\mathbf{u}}|_{\mathbf{u}_n^{i-1}} \cdot \Delta\mathbf{u} \approx 0. \tag{5.14}$$

Eq. (5.14) is used to give contributions to the discrete equation set in Eq. (5.12). The contributions to the discrete equation set in Eq. (5.12), that are derived from Eq. (5.14), will not be illustrated in this dissertation, as it falls outside its scope. Rather, the reader is directed to a previous work of the research group that this dissertation belongs to in [51], in which the contributions to the discrete equation set from Eq. (5.14) are clearly derived. Nevertheless, the discrete equation system takes the form of

$$\mathbf{K}_n^i \Delta\mathbf{u}_n^i = \left[\mathbf{f}_{ext}\right]_n^i - \left[\mathbf{f}_{int}\right]_n^i = \mathbf{R}_n^i, \tag{5.15}$$

where, $\mathbf{R}_n^i$ is the residual of the of the discrete equation system. Also, $\mathbf{K}_n^i = \mathbf{K}\left(\mathbf{u}_n^{i-1}\right)$, and $\mathbf{R}_n^i = \mathbf{R}\left(\mathbf{u}_n^{i-1}\right)$, i.e., the tangent matrix and residual vector respectively, are evaluated at the initial guess for the displacement field, $\mathbf{u}_n^{i-1}$. Next, the discrete equation system is solved for the unknown displacement increment, $\Delta\mathbf{u}_n^i$, via

$$\Delta\mathbf{u}_n^i = \left[\mathbf{K}_n^i\right]^{-1}\mathbf{R}_n^i, \tag{5.16}$$

Therefore, the displacement field, $\mathbf{u}_n^i$, is updated at each iteration step using

$$\mathbf{u}_n^i = \mathbf{u}_n^{i-1} + \Delta\mathbf{u}_n^i. \tag{5.17}$$

The updated displacement vector, $\mathbf{u}_n^i$, is then used as the initial guess for the next iteration $(i+1)$. It is important to note that for the first iteration step $(i=0)$, the initial guess of the displacement vector, is taken to be the displacement solution from the previous converged calculation step, i.e., $\mathbf{u}_n^{i-1} = \mathbf{u}_{n-1}$.

The iterative procedure is terminated when the Euclidean norm of residual vector, $||\mathbf{R}_n^i||$,

falls below a desired threshold, δ_r. $||\mathbf{R}_n^i||$, is computed as

$$||\mathbf{R}_n^i|| = \sqrt{[\mathbf{R}_n^i]^{\mathrm{T}}\,\mathbf{R}_n^i}. \tag{5.18}$$

When $||\mathbf{R}_n^i||$ falls below the desired threshold, it implies that $[\mathbf{f}_{ext}]_n^i \approx [\mathbf{f}_{int}]_n^i$, which represents static equilibrium between externally applied loads and internal reactions of the body. In this scenario, the calculation then proceeds to the next loading step (i.e. $n := n + 1$).

5.3.2 Specific Aspects of Assembly for Cardiac Modelling

In the the succeeding sections, some specific aspects of the assembly stage, shall be explored for the cardiac mechanics problem.

5.3.2.1 Incremental Loading Approach to Modelling the Diastolic Filling

In the current framework of SESKA, the diastolic filling phase is usually pressure driven. Therefore, the cavity pressure is increased from a starting pressure, up to some user specified end-diastolic maximum. In order to solve the variational formulation for a specified end-diastolic pressure, it is necessary to solve the discrete equation system for incrementally increasing load, because Newton's method will only converge if the initial guess for the displacement vector, is close enough to the actual solution.

The incremental loading procedure is achieved by multiplying a final pressure, P_0 (usually initialized to unity), by a Neumann loading factor, λ_n, that increases from an initial value of 0, upto some user defined maximum loading (i.e. the required end-diastolic pressure). The blood pressure, P, exerted on the endocardium is therefore expressed as

$$P = \lambda_n P_0. \tag{5.19}$$

Furthermore, λ_n in Eq. (5.19) is computed as

$$\lambda_n = \lambda_{n-1} + \Delta\lambda_n, \tag{5.20}$$

where $\Delta\lambda_n$ is the loading factor increment for the current calculation step. By expressing the pressure as a loading factor multiplied by a final pressure, the corresponding term in the variational formulation is reformulated as

$$\int_{\partial\mathcal{B}_N} PJ\mathbf{F}^{-\mathrm{T}}\mathbf{n}\cdot\delta\mathbf{u}dA = \lambda_n \int_{\partial\mathcal{B}_N} P_0 J\mathbf{F}^{-\mathrm{T}}\mathbf{n}\cdot\delta\mathbf{u}dA, \tag{5.21}$$

The contributions to the external loading vector, $[\mathbf{f}_{ext}]_n^i$, come from the integral on the left hand side of Eq. (5.21). The integral on the right hand side of Eq. (5.21), gives contributions to the final external loading vector, $[\mathbf{f}_{ext,0}]$. Multiplying the final external loading vector by the current loading factor, λ_n, provides an alternate interpretation for $[\mathbf{f}_{ext}]_n^i$, given by

$$[\mathbf{f}_{ext}]_n^i = \lambda_n \left[\mathbf{f}_{ext,0}\right], \tag{5.22}$$

Hence, for an incremental loading approach, the discrete equation system in Eq. (5.15) is reformulated as

$$\mathbf{K}_n^i \Delta \mathbf{u}_n^i = \lambda_n \left[\mathbf{f}_{ext,0}\right] - [\mathbf{f}_{int}]_n^i = \mathbf{R}_n^i. \tag{5.23}$$

5.3.2.2 Isovolumetric Modelling

During the phases of isovolumetric contraction and relaxation, the mitral and aortic valves are closed. Simultaneously, an active contraction is present in the left ventricle, which is increasing during isovolumetric contraction and decreasing during isovolumetric relaxation. The closure of the valves during isovolumetric phases, and consequent entrapment of blood within the left ventricle, implies that from a physiological point of view, the cavity volume remains more or less constant. Hence, due to the active contraction in the ventricle, which attempts to deform it, the cavity pressure in the ventricle must adjust in manner that ensures no change in cavity volume and hence static equilibrium.

The requirement of no volume change ($\Delta V \approx 0$), implies that at each iteration step, i, of a time step, n, a suitable pressure adjustment, $\Delta P = \Delta\lambda_n^i P_0$, is required such that the residual of the discrete equation system vanishes. For isovolumic phases, the discrete equation system takes the form of

$$\mathbf{K}_n^i \Delta \mathbf{u}_n^i = \Delta\lambda_n^i \left[\mathbf{f}_{ext,0}\right] + \mathbf{R}_n^i. \tag{5.24}$$

Eq. (5.24) is then rearranged to solve for the unknown displacement increment at each iteration step

$$\begin{aligned} \Delta \mathbf{u}_n^i =& \Delta\lambda_n^i \left[\mathbf{K}_n^i\right]^{-1} \left[\mathbf{f}_{ext,0}\right] + \left[\mathbf{K}_n^i\right]^{-1} \mathbf{R}_n^i \\ =& \Delta\lambda_n^i \Delta \mathbf{u}_{ext,n}^i + \Delta \mathbf{u}_{\mathbf{R},n}^i, \end{aligned} \tag{5.25}$$

where, $[\mathbf{f}_{ext,0}]$ is the the reference external loading vector corresponding to some final pressure, P_0, the latter usually initialized to unity. Additionally, $\Delta\mathbf{u}^i_{ext,n}$ can be interpreted as the displacement increment for an iteration step, due to full force loading with no internal reaction. Additionally, $\Delta\mathbf{u}^i_{\mathbf{R},n}$ is the displacement increment for an iteration step, due to the residual vector. Furthermore, the reference external loading vector needs to be close enough to the current loading in order to ensure rapid convergence, of the iterative Newton's method, used to solve for the displacement increment. Thus, at the beginning of each time step, the reference external loading vector, $[\mathbf{f}_{ext,0}]$, is adjusted using an initial loading correction, $\Delta\lambda^0_n$, via

$$\begin{aligned}\mathbf{f}^*_{ext,0} =&\Delta\lambda^0_n\left[\mathbf{f}_{ext,0}\right]\\ =&\lambda_{ref}\left[\mathbf{f}_{ext,0}\right],\end{aligned} \tag{5.26}$$

where $\Delta\lambda^0_n$ corresponds to the loading correction found from the previous time step. As a side note, it is important that the reference loading factor, λ_{ref}, is positive. Consequently, the displacement increment in Eq. (5.25), is reformulated in terms of the adjusted reference external loading vector as

$$\begin{aligned}\Delta\mathbf{u}^i_n =&\lambda^{-1}_{ref}\left(\Delta\lambda^i_n\left[\mathbf{K}^i_n\right]^{-1}\mathbf{f}^*_{ext,0}\right)+\left[\mathbf{K}^i_n\right]^{-1}\mathbf{R}^i_n\\ =&\lambda^{-1}_{ref}\Delta\lambda^i_n\Delta\mathbf{u}^i_{ext,n}+\Delta\mathbf{u}^i_{\mathbf{R},n}.\end{aligned} \tag{5.27}$$

The first step to obtaining the sought after loading factor increment, $\Delta\lambda^i_n$, is to obtain an expression for the total volume change, ΔV_n, over a calculation step. This expression is derived as

$$\Delta V_n=\sum_{k=0}^{i-1}\left[\Delta V^k_n\left(\Delta\mathbf{u}^k_n\right)\right]+\left[\lambda^{-1}_{ref}\Delta\lambda^i_n\Delta V^i_{ext,n}\left(\Delta\mathbf{u}^i_{ext,n}\right)+\Delta V^i_{\mathbf{R},n}\left(\Delta\mathbf{u}^i_{\mathbf{R},n}\right)\right], \tag{5.28}$$

where, $\sum_{k=0}^{i-1}\left[\Delta V^k_n\left(\Delta\mathbf{u}^k_n\right)\right]$, represents the accumulated volume change for the time step, which includes contributions from iteration step k=0 to k=i-1. The sum of the second and third term, on the right hand side of Eq. (5.28), represents the volume change accumulated during the current iteration step. $\Delta V^i_{ext,n}\left(\Delta\mathbf{u}^i_{ext,n}\right)$ represents the volume change due to to a preliminary displacement correction, $\mathbf{u}^i_{ext,n}$ and $\Delta V^i_{\mathbf{R},n}\left(\Delta\mathbf{u}^i_{\mathbf{R},n}\right)$, represents that due to a preliminary displacement correction, $\Delta\mathbf{u}^i_{\mathbf{R},n}$. The sought after loading factor, $\Delta\lambda^i_n$, can be found by rearranging Eq. (5.28), which yields,

$$\Delta\lambda^i_n=\lambda_{ref}\frac{\Delta V_n-\sum_{k=0}^{i-1}\left[\Delta V^k_n\left(\Delta\mathbf{u}^k_n\right)\right]-\Delta V^i_{\mathbf{R},n}\left(\Delta\mathbf{u}^i_{\mathbf{R},n}\right)}{\Delta V^i_{ext,n}\left(\Delta\mathbf{u}^i_{ext,n}\right)}, \tag{5.29}$$

where $\Delta V_n = 0$ for isovolumetric contraction and relaxation phases. Next, the loading factor is updated as

$$\lambda_n^i = \lambda_n^{i-1} + \Delta\lambda_n^i, \tag{5.30}$$

where, λ_n^{i-1} is the loading factor from the previous iteration step. The final displacement correction, $\Delta\mathbf{u}_n^i$, is then given by

$$\Delta\mathbf{u}_n^i = \Delta\mathbf{u}_{\mathbf{R},n}^i + \lambda_{ref}^{-1}\Delta\lambda_n^i\Delta\mathbf{u}_{ext,n}^i, \tag{5.31}$$

incorporating contributions due to the residual from the previous iteration step as well as the loading correction, the latter ensuring the specified volume increment ΔV_n (which is equal to zero for isovolumic phases).

The required or target cavity volume, $V_{targ,n}$, at each time step, n, is given by

$$V_{targ,n} = V_{n-1} + \Delta V_n, \tag{5.32}$$

where ΔV_n is zero only for the isovolumic phases. Hence, when the actual computed cavity volume, V_{act}, is within 1 percent of the target volume, then the loading factor, λ_n^i, is no longer updated and is held constant for the remaining iteration steps. In this case, a time step has converged if the Euclidean norm of the residual vector (Eq. (5.18)), falls below its desired threshold, δ_r.

5.3.2.3 Ejection Modelling

To model the ejection phase of the cardiac cycle, the form of the discrete equation system is identical to that used for the isovolumetric phases, implying that

$$\mathbf{K}_n^i\Delta\mathbf{u}_n^i = \Delta\lambda_n^i\,[\mathbf{f}_{ext,0}] + \mathbf{R}_n^i. \tag{5.33}$$

Therefore, the ejection phase is modelled in exactly the same iterative manner as the isovolumetric phases (see Sec. 5.3.2.2). However, the difference is that the target volume increment for a calculation step is not zero ($\Delta V_n \neq 0$) and is rather obtained using an iterative procedure, involving the three element Windkessel model. The target volume increment obtained at each iteration step is then utilized to find a suitable pressure adjustment, $\Delta\lambda_n^i$, via Eq. (5.29), such that the actual cavity volume, approximately matches the target volume, the latter computed as

$$V_{target,n}^i = V_{n-1} + \Delta V_n \approx V_{actual,n}^i, \tag{5.34}$$

where, $V^i_{target,n}$ and $V^i_{actual,n}$, represent the target and actual cavity volumes, for a particular iteration step of a time step. Also, V_{n-1} represents the the cavity volume from the previous converged time step.

In Chap. 3, the three element Windkessel model was explored, which is used to model the global hemodynamics of the arterial system. Eq. (3.35) represented the governing differential equation for the three element Windkessel model. Since the three element Windkessel model is used in an iterative procedure, subscripts i and superscripts n, representing the iteration number and time step, respectively, are added to Eq. (3.35), yielding,

$$(CR_0)\left[\frac{dI_{ao}}{dt}\right]^i_n + \left(1+\frac{R_0}{R}\right)[I_{ao}]^i_n = (C)\left[\frac{dP_{ao}}{dt}\right]^i_n + \left(\frac{1}{R}\right)[P_{ao}]^i_n\,, \tag{5.35}$$

where,

- C is the arterial compliance,
- R is the peripheral resistance,
- R_0 is the flow resistance,
- $[P_{ao}]^i_n$ is the aortic pressure,
- and $[I_{ao}]^i_n$ is the aortic flow rate.

Additionally, during ejection, the aortic flow rate, is equal to the negative of the rate of change of left ventricular cavity volume. Thus, the aortic flow rate is given by

$$[I_{ao}]^i_n = [I_{lv}]^{\mathrm{i}}_n = -\left[\frac{dV_{lv}}{dt}\right]^i_n. \tag{5.36}$$

Also, during ejection, the aortic pressure is approximately equal to the left ventricular cavity pressure [3]. Hence, the aortic pressure is simplified to

$$[P_{ao}]^i_n \approx [P_{lv}]^i_n\,. \tag{5.37}$$

Noting the above two simplifications, the three element Windkessel model is now reformulated in terms of left ventricular cavity pressure and volume, which yields,

$$(CR_0)\left[\frac{dI_{lv}}{dt}\right]^i_n + \left(1+\frac{R_0}{R}\right)[I_{lv}]^i_n = (C)\left[\frac{dP_{lv}}{dt}\right]^i_n + \left(\frac{1}{R}\right)[P_{lv}]^i_n\,. \tag{5.38}$$

Provided that the time increment is small, it follows that $dt \approx \Delta t$, $dV_{lv} \approx \Delta V_{lv}$ and $dP_{lv} \approx \Delta P_{lv}$. Therefore, the time discretization of Eq. (5.38) is given by

$$(CR_0)\left[\frac{\Delta I_{lv}}{\Delta t}\right]_n^i + \left(1+\frac{R_0}{R}\right)[I_{lv}]_n^i = (C)\left[\frac{\Delta P_{lv}}{\Delta t}\right]_n^i + \left(\frac{1}{R}\right)[P_{lv}]_n^i . \tag{5.39}$$

Furthermore, at each iteration step, the left ventricular pressure can be expressed in terms of some final pressure, P_0 (usually initialized to unity), multiplied by the un-updated Neumann loading factor for the iteration, λ_n^i. However, for the first iteration step (i=0), the un-updated Neumann loading factor is equal to that from the previous converged time step, λ_{n-1}. Noting the comments made in this paragraph, the left ventricular pressure is formulated as

$$[P_{lv}]_n^i = \begin{cases} \lambda_{n-1}P_0 \text{ for } i=0, \\ \lambda_n^i P_0 \text{ for } i>0, \end{cases} \tag{5.40}$$

In addition, the rate of change of flow rate is approximated via backward differences at each iteration step, using the relation

$$\left[\frac{\Delta I_{lv}}{\Delta t}\right]_n^i = \frac{[I_{lv}]_n^i - [I_{lv}]_{n-1}}{\Delta t}, \tag{5.41}$$

where $[I_{lv}]_{n-1}$, is the flow rate from the previous converged time step. Similarly, the rate of change of left ventricular pressure, is approximated via backward differences. However, for the first iteration step (i=0), this derivative is estimated to be equal to that from the previous converged time step, i.e., $\left[\frac{\Delta P_{lv}}{\Delta t}\right]_{n-1}$. Noting these comments, the rate of change of left ventricular pressure is formulated as

$$\left[\frac{\Delta P_{lv}}{\Delta t}\right]_n^i = \begin{cases} \left[\frac{\Delta P_{lv}}{\Delta t}\right]_{n-1} \text{ for } i=0, \\ P_0\frac{\lambda_n^i-\lambda_{n-1}}{\Delta t} \text{ for } i>0. \end{cases} \tag{5.42}$$

Next, Eq. (5.40), Eq. (5.41) and Eq. (5.42), are substituted in to the Windkessel equation (Eq. (5.39)). The resulting equation is then rearranged to solve for the flow rate update, $[I_{lv}]_n^i$, for the current iteration step. The updated flow rate for the current iteration is given by

$$[I_{lv}]_n^i = \begin{cases} \left[\frac{1}{\frac{CR_0}{\Delta t}+\left(1+\frac{R_0}{R}\right)}\right]\left(CP_0\left[\frac{\Delta P_{lv}}{\Delta t}\right]_{n-1} + \frac{P_0\lambda_{n-1}}{R} + \frac{CR_0}{\Delta t}I_{n-1}\right) \text{ for } i=0, \\ \left[\frac{1}{\frac{CR_0}{\Delta t}+\left(1+\frac{R_0}{R}\right)}\right]\left(CP_0\left[\frac{\lambda_n^i-\lambda_{n-1}}{\Delta t}\right] + \frac{P_0\lambda_n^i}{R} + \frac{CR_0}{\Delta t}I_{n-1}\right) \text{ for } i>0. \end{cases} \tag{5.43}$$

Referring to Eq. (5.43), for i=0, the flow rate is computed using an explicit one-step Euler backward prediction. However, for i >0, the flow rate is computed using a one-step backward implicit Euler correction. Next, the target volume increment, ΔV_n, is then updated at each iteration step via

$$\Delta V_n = -\Delta t \left[I_{lv}\right]_n^i , \tag{5.44}$$

where the negative sign in Eq. (5.44) is necessary to enforce the decrease in cavity volume that occurs in ejection, due to the contracting ventricle. Consequently, this target volume increment is substituted in to Eq. (5.29), to find the loading factor increment, $\Delta\lambda_n^i$. Then, the loading factor, λ_n^i, is updated using Eq. (5.30). Furthermore, the final displacement increment, $\Delta\mathbf{u}_n^i$, is computed using Eq. (5.31).

The iterative procedure of solving for the target volume increment is conducted up until the absolute value of the residual of the Windkessel differential equation, falls below a certain tolerance, i.e., $|\mathrm{R}_{\mathrm{Windkessel}}| \leq \delta_w$. The residual of the Windkessel differential equation is given by

$$\begin{aligned} \mathrm{R}_{\mathrm{Windkessel}} = &\left[(C)\left[\frac{\Delta P_{lv}}{\Delta t}\right]_n^i + \left(\frac{1}{R}\right)\left[P_{lv}\right]_n^i\right] \\ &- \left[(CR_0)\left[\frac{\Delta I_{lv}}{\Delta t}\right]_n^i + \left(1+\frac{R_0}{R}\right)\left[I_{lv}\right]_n^i\right]. \end{aligned} \tag{5.45}$$

It is important to note that in Eq. (5.45), $[I_{lv}]_n^i$, is the un-updated flow rate for the iteration. Similarly, $[P_{lv}]_n^i$ is the un-updated cavity pressure for the iteration. Once the residual of the Windkessel differential equation falls below the specified tolerance, then the target volume increment is no longer updated and is held constant for the remaining iteration steps of a time step. Furthermore, a calculation step in the ejection phase is only terminated once the residual of the Windkessel differential equation and the Euclidean norm of the residual vector (Eq. (5.18)), fall below their desired thresholds.

For clarity sake, it is important to note that the Windkessel differential equation is solved as an initial value problem, with initial flow rate and rate of change of flow rate equal to zero at the instant of vale opening. Also, the initial cavity pressure and rate of change of pressure are equal to that at the end of isovolumetric contraction. Finally, the ejection phase terminates when the flow rate changes sign, indicating that the left ventricle has ceased to contract and push blood into the aorta, at which point the aortic valves shut and isovolumetric relaxation begins.

5.4 Implementations

The author has made several contributions to the cardiac mechanics module of the in-house modelling software SESKA. These contributions are listed below:

- Coupling of the three element Windkessel model to the existing cardiac mechanics model.
- Implementation of a new calculation control mode, allowing systole (isovolumetric contraction and ejection) to be run in one simulation.
- Implementation of a new calculation control mode, allowing systole and isovolumetric relaxation to be run in one simulation.

Details of the implementation of the mentioned calculation control modes, will be given in the following sections.

5.4.1 Systolic Calculation Control Mode

This calculation control mode allows one to model the systolic phase of the cardiac cycle, without having to perform a restart of the simulation after isovolumetric contraction has terminated. Fig. 5.1 is a schematic illustration of the implementation and workings of the systolic calculation control mode. In Fig. 5.1, it is important to note that λ_{ao}, refers to the desired level of aortic pressure, at which isovolumetric contraction is terminated.

Referring to Fig. 5.1, initial loading conditions and particle degrees of freedom for the left ventricle, are read out of the calculation output file generated at the end of diastolic filling. Boundary conditions for the model and material parameters are additionally read in from an input file.

For a time step n , the cavity pressure (which is equal to the Neumann loading factor), is initialized to the value from the previous converged time step. Similarly, the initial particle degrees of freedom are set to those from the previous converged time step. Thus, all components of the discrete equation system can be assembled, i.e., $\mathbf{K}_n^i$, $\mathbf{R}_n^i$ and $\mathbf{f}_{ext,0}$. Next, the Neumann loading factor increment, $\Delta\lambda_n^i$, is solved for, and to do so a distinction must first be made as to which phase of systole the time step resides within. If the Neumann loading at the beginning of a time step, is less than the level of aortic pressure required to initiate ejection, then the time step resides within the isovolumetric contraction phase. Conversely, if the loading factor is greater than the

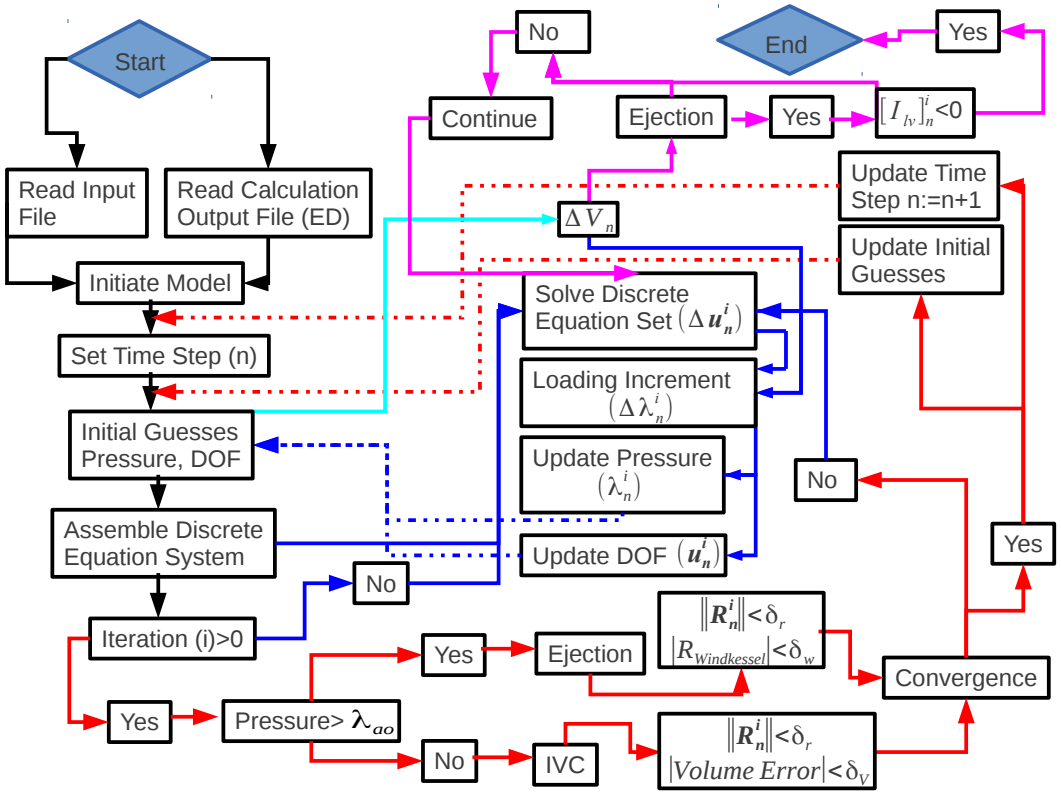

FIGURE 5.1: Schematic of the systole calculation control mode.

required level of aortic pressure, then the time step resides within the ejection phase. For a time step in isovolumetric contraction, the Neumann loading increment is computed using Eq. (5.29), with $\Delta V_n = 0$. However, for a time step in the ejection phase, ΔV_n is computed from the three element Windkessel model, using Eq. (5.44), which is in-turn substituted into Eq. (5.29), to find the sought after loading increment. Next, the loading factor is updated using the solved for loading increment and Eq. (5.30). The degrees of freedom are then updated using the displacement increment computed using Eq. (5.31). The next iteration steps of a time step follow the same procedure, except with new initial guesses for the loading factor and particle degrees of freedom.

The termination criteria for a time step can be considered. Once again, a distinction must be made as to which phase of systole a time step is located in. A time step within the isovolumetric contraction phase, has converged only if both the Euclidean norm of the residual of the discrete equation system, $||\mathbf{R}_n^i||$, and the absolute value of the percentage volume error (between target and actual cavity volume), have fallen below desired thresholds of δ_r and δ_V respectively. Alternatively, for a time step in the ejection phase, convergence is achieved only if both $||\mathbf{R}_n^i||$ and the absolute value of the residual of the three element Windkessel differential equation, $|\mathrm{R}_{\mathrm{Windkessel}}|$, have fallen below desired thresholds of δ_r and δ_w respectively. Once a time step has converged, the loading

factor and particle degrees of freedom are no longer updated. The stored loading factor and particle degrees of freedom are then used as initial guesses for the next time step.

The simulation terminates, when the computed flow rate during ejection, falls below zero, thus the cavity volume has ceased to decrease, causing the aortic valves to close, which signals the start of the last phase of the cardiac cycle, isovolumetric relaxation.

5.4.2 Systole and Isovolumetric Relaxation Control Mode

This calculation control mode allows a user to simulate the systolic and isovolumetric relaxation phases of the cardiac cycle, without having to perform a restart between isovolumetric contraction and ejection or between ejection and isovolumetric relaxation. Fig. 5.2 is a schematic representation of the workings of this calculation control mode. It should be noted that λ_{EIVR}, denotes the desired final pressure at the end of isovolumetric relaxation.

Referring to Fig. 5.2, it is clear that the algorithm for running systole and isovol-

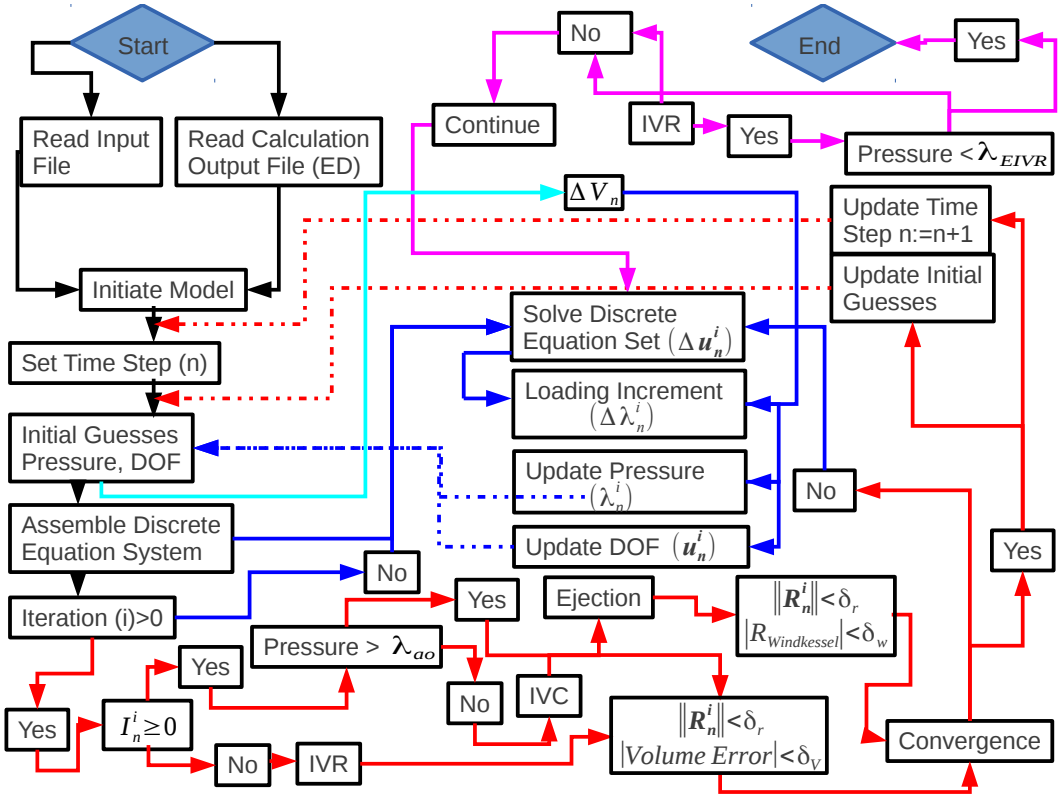

FIGURE 5.2: Schematic of the systole and isovolumetric relaxation control mode.

umetric relaxation, is very similar to that used to simulate systole. There are however three primary differences. The first difference pertains to the calculation of the Neumann

loading factor update. In the case of the systole and isovolumetric relaxation calculation control mode, the algorithm adaptively decides whether a time step is within isovolumetric contraction ($I_{n-1} \geq 0$ and $\lambda_{n-1} \leq \lambda_{ao}$), ejection ($I_{n-1} \geq 0$ and $\lambda_{n-1} \geq \lambda_{ao}$) or isovolumetric relaxation ($I_{n-1} \leq 0$). If a time step is within isovolumetric contraction or relaxation, then the volume increment, ΔV_n, used for the computation of the sought after loading increment is zero. If however, the time step is within ejection, then the volume increment used to compute the loading increment, is derived from the three element Windkessel model (i.e. Eq. (5.44)).

The second difference has to do with the time step convergence criteria. If a time step is within isovolumetric relaxation or contraction, then it has converged only if both the Euclidean norm of the residual of the discrete equation system, $||\mathbf{R}_n^i||$, and the absolute value of the percentage volume error (between target and actual cavity volume), have fallen below desired thresholds of δ_r and δ_V respectively. Alternatively, if a time step is in the ejection phase, then it has converged only if both $||\mathbf{R}_n^i||$ and the absolute value of the residual of the three element Windkessel differential equation, $|\mathrm{R}_{\mathrm{Windkessel}}|$, have fallen below desired thresholds of δ_r and δ_w respectively.

The last difference pertains to the simulation termination criteria. For this calculation control mode, the simulation is terminated only when the Neumann loading factor, during isovolumetric relaxation, has fallen below a desired final pressure λ_{EIVR}.

Chapter 6

Inverse Modelling

6.1 Introduction

Constitutive equations that govern both passive and active mechanical behaviour of healthy myocardium were explored. In addition, the three element Windkessel model was introduced in order to couple the left ventricular mechanical response, to the systemic response of the arterial system in the ejection phase of the cardiac cycle. Furthermore, the idealized canine ellipsoidal model of the left ventricle, which is utilized to simulate left ventricular mechanical response, was explored. Also, the manner in which left ventricular cardiac function is modelling in the in-house modelling software SESKA, was outlined in chapter five.

The methods available for obtaining material parameters (active tension, passive and Windkessel parameters), that produce a best fit between simulated left ventricular mechanical response and that found experimentally, will be outlined. Identification of material parameters can be obtained using numerical simulations (e.g. FEM or EFGM methods), in combination with measurements of variables, such as stresses, strains, pressure, volume or displacements, obtained from experimental testing. Such methods of solving for material parameters are referred to as inverse methods [19]. Various inverse methods are presented in the literature, however, Avril et al. [19] identified two broad categories, namely, updating and non-updating methods.

6.2 Methods for Inverse Modelling

In the following sections, two broad categories of inverse modelling will be explained, namely, updating and non-updating methods. Furthermore, the chosen method for

inverse modelling, in the context of this research, will be revealed.

6.2.1 Updating Methods of Inverse Modelling

These methods involve the computation of displacement, strain, stress, pressure or volume fields, via a numerical analysis, using initial guesses for material parameters. Actual material parameters are then identified by minimizing an objective cost function, which compares available measurements, to the corresponding simulated results. This method of finding constitutive parameters is advantageous, as full-field measurements are not required, rather, only a few measurements distributed across the volume of interest are necessary.

Updating methods however, involve iterative computational procedures and are thus CPU intensive. In addition, such methods are sensitive to measurement noise if based on scarce experimental data [19]. The updating method of inverse modelling is shown schematically in Fig. 6.1.

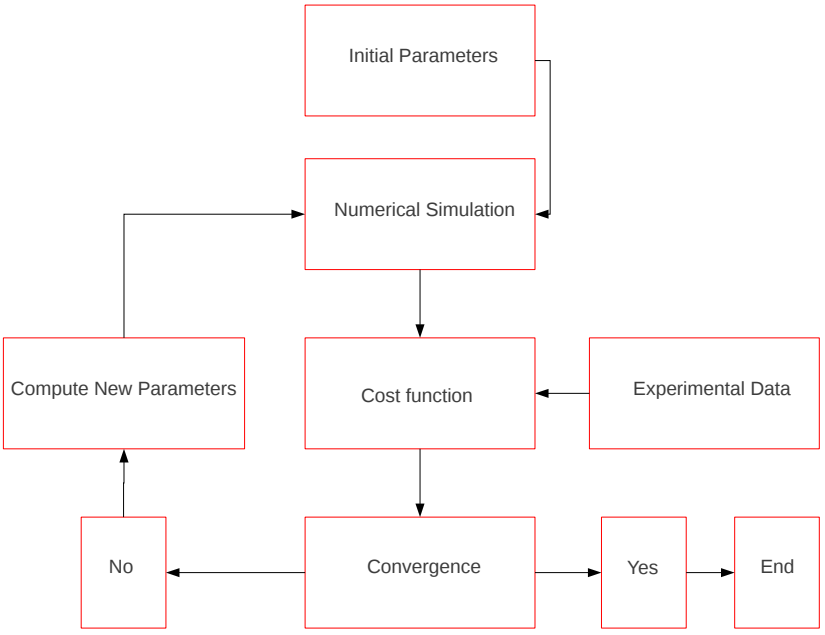

FIGURE 6.1: updating method of inverse modelling.

6.2.2 Non-Updating Methods of Inverse Modelling

In non-updating methods of inverse modelling [19], simulated fields (e.g. strain, stress, pressure or volume fields) are evaluated from experimentally measured fields, using a postulated material model, whose parameters are to be identified and depend implicitly

upon the latter. Equilibrium equations then provide the relationships that need to be solved, in order to obtain the sought after constitutive parameters. These methods are advantageous, as they are usually associated with faster computational time. However, such methods require full-field experimental data, with very sharp spatial resolution.

6.2.3 Chosen Method of Inverse Modelling

For the purpose of this research, updating methods of inverse modelling will be used, in order to solve for material parameters, that best simulate the mechanical response of the left ventricle. The main reason for this choice of inverse method is due to the scarcity of detailed full-field experimental data, in the literature (such as stress, strain or displacement data, distributed across the volume of interest). However, as was mentioned earlier, an objective cost function that is to be minimized, must be formulated for use in updating methods of inverse modelling and the nature of this cost function will be explored in the following sections.

6.3 Formulation of an Objective Cost Function

In the previous section, it was explained that the core of updating methods of inverse modelling, involves minimizing a cost function, that compares simulated to experimental data. Furthermore, the nature of the cost function can be single or multi objective and this aspect will be outlined in detail in the following sections.

6.3.1 Single objective Cost Functions

A single objective cost function, is constructed from only one objective function. In the context of least squares curve fitting applications, the cost function compares an experimentally observed vector, $\hat{\mathbf{f}}$, to a corresponding simulated vector, $\mathbf{f}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a vector of parameters. Hence, a single objective cost function can be constructed as the sum of the squares, of the difference between experimental and simulated data [52] as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2} \sum_{i=0}^{i=N-1} \left[\left(\frac{\hat{\mathrm{f}}_i - \mathrm{f}(\boldsymbol{\theta})_i}{\mathrm{w}_i^{\hat{\mathbf{f}}}} \right)^2 \right], \tag{6.1}$$

where $i \in (0, N-1)$ and $\hat{\mathrm{f}}_i$ are the components of the experimental vector, $\hat{\mathbf{f}}$. In addition, $\mathrm{f}(\boldsymbol{\theta})_i$ are the components of the simulated vector, $\mathbf{f}(\boldsymbol{\theta})$, and $\frac{1}{\left(\mathrm{w}_i^{\hat{\mathbf{f}}}\right)^2}$ are diagonal

components of a weight matrix, $\mathbf{W}^{\hat{\mathbf{f}}}$, whose off-diagonal components are zero. Furthermore, N refers to the number of data comparison points. Alternatively, the cost function represented in Eq. (6.1) can be expressed in vector-matrix notation as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\left[\left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{f}}} \left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)\right]. \tag{6.2}$$

Furthermore, Eq. (6.2) can be expressed in index notation as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}\left[\left(\hat{\mathrm{f}}_i - \mathrm{f}_i(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathrm{W}_{ij}^{\hat{\mathbf{f}}} \left(\hat{\mathrm{f}}_j - \mathrm{f}_j(\boldsymbol{\theta})\right)\right], \tag{6.3}$$

where $i, j \in (0, \text{N-1})$. Consequently, in terms of material parameter identification, via minimization of a single objective cost function, a mathematical minimization problem is constructed as

$$\begin{aligned} &\text{minimize } J^2(\boldsymbol{\theta}), \\ &\text{subject to } \boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}, \end{aligned} \tag{6.4}$$

where $\boldsymbol{\theta}^{\min}$ and $\boldsymbol{\theta}^{\max}$ represent the lower and upper bound of the parameter space.

6.3.2 Multi Objective Cost Functions

In certain cases, it may be desired to construct the cost function out of more than one objective. For example, biaxial testing of myocardial tissue involves the measurement of displacements in both the fibre (or circumferential) and sheet (or longitudinal) directions. Consequently, reproducing the behaviour of the tissue using numerical modelling such as Finite Element Modelling, would first involve identifying a set of material parameters that properly models the tissue, in both the fibre an sheet directions. In such a case, the cost function would be constructed from both the simulated and experimental fibre and sheet displacements. This cost function would be an example of a bi-objective cost function, as it would be constructed from two objectives, namely, fibre and sheet displacements.

Considering the general case, a cost function vector is constructed, whose elements are the individual single objective cost functions to be minimized [14]. The cost function vector is given by

$$[\boldsymbol{J}(\boldsymbol{\theta})]^{\mathrm{T}} = \left[\left[J^2(\boldsymbol{\theta})\right]_1, ..., \left[J^2(\boldsymbol{\theta})\right]_m\right], \tag{6.5}$$

where, m refers to the number of single objective functions and $\boldsymbol{J}(\boldsymbol{\theta})$ is the cost function vector, whose elements are $\left[J^2(\boldsymbol{\theta})\right]_m$. In terms of material parameter identification, a mathematical minimization problem can be constructed as

$$\begin{aligned} &\text{minimize } \boldsymbol{J}(\boldsymbol{\theta}), \\ &\text{subject to } \boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}. \end{aligned} \tag{6.6}$$

6.3.2.1 The Concept of Pareto Optimality

$\boldsymbol{J}(\boldsymbol{\theta})$ is a vector, hence, if any of its components are conflicting, there is no unique solution to the minimization problem in Eq. (6.6) [14]. Thus, the concept of non-inferiority or Pareto optimality, must be utilized in order to characterize the objectives [53]. A non-inferior or Pareto optimal solution is defined as one in which an improvement in one objective results in a degradation of another [14].

Definition: a solution $\boldsymbol{\theta}$ ($\boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}$), is a non-inferior solution, if for some neighbourhood of $\boldsymbol{\theta}$, there does not exist a parameter increment, $\delta\boldsymbol{\theta}$, that satisfies conditions given by

$$\begin{aligned} &\boldsymbol{\theta}^{\min} \leq \delta\boldsymbol{\theta} + \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}, \\ &\left[J^2(\boldsymbol{\theta} + \delta\boldsymbol{\theta})\right]_i \leq \left[J^2(\boldsymbol{\theta})\right]_i \quad \text{where,} \quad i = 1, ..., m, \\ &\left[J^2(\boldsymbol{\theta} + \delta\boldsymbol{\theta})\right]_j < \left[J^2(\boldsymbol{\theta})\right]_j \quad \text{for some } j. \end{aligned} \tag{6.7}$$

Fig. 6.2 illustrates the concept of Pareto optimality, for a bi-objective cost function, $[\boldsymbol{J}(\boldsymbol{\theta})]^{\mathrm{T}} = [\mathrm{F}_1, \mathrm{F}_2]$. The set of non-inferior solutions lies on the curve between C and D. Points A and B represent specific non-inferior points. In addition, Λ denotes the feasible region for the objective function space. Referring to Fig. 6.2, points A and B are

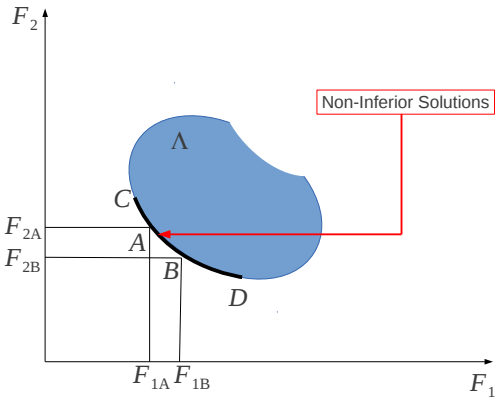

FIGURE 6.2: Illustration of the concept of Pareto optimality [14].

clearly non-inferior solutions, because an improvement in one objective, F_1, requires a

degradation in the other objective, F_2, i.e., $F_{1B} < F_{1A}$, $F_{2B} > F_{2A}$ [14]. In the context of this research, very little attention will be paid to the concept of Pareto optimality. The reason for this assumption is due to the fact that for the considered applications, it is expected that a unique solution exists that minimizes all objective functions.

6.3.2.2 The Weighted Sum Method

There are numerous methods available for multi-objective minimization, however, the most common techniques involve formulating a single-objective minimization problem, such that optimal solutions to this problem, are Pareto optimal solutions to the multi-objective minimization problem [14, 54].

The weighted sum approach to multi-objective minimization, converts the multi-objective problem of minimizing the vector, $\boldsymbol{J}(\boldsymbol{\theta})$, into a scalar problem, by constructing a weighted sum of all the objectives [54]. Hence, a mathematical minimization problem is constructed as

$$\begin{aligned} &\text{minimize } J^2(\boldsymbol{\theta}) = \sum_{i=1}^{i=m} \omega_i \left[J^2(\boldsymbol{\theta})\right]_i, \\ &\text{subject to } \boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}, \end{aligned} \tag{6.8}$$

where $J^2(\boldsymbol{\theta})$ is a scalar cost function. Considering the geometric representation of the objective function space, for the bi-objective case in Fig. 6.3, a line $L = \boldsymbol{\omega}^{\mathrm{T}}\boldsymbol{J}(\boldsymbol{\theta}) = c$ is drawn, where $\boldsymbol{\omega}$ is a vector containing the weighting coefficients, ω_i, of each objective function, $\left[J^2(\boldsymbol{\theta})\right]_i$ (where $i \in (1, 2)$). The minimization of Eq. (6.8), can be interpreted as finding the value of c, for which L just touches the boundary of the feasible objective function space, Λ, as it proceeds outwards from the origin [14]. Therefore, the weighting coefficients, ω_i, define the slope of L, which in turn leads to the solution point where the latter touches the boundary of Λ. This method however is not without its draw-backs

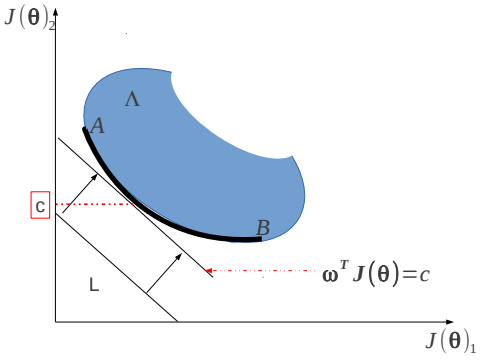

FIGURE 6.3: Geometric representations of the weighted sum method [14].

[14]. Firstly, different weights, ω_i, will produce different Pareto optimal solutions. In addition, the weighting coefficients do not necessarily correspond directly to the relative importance of the objectives or allow trade-off's between the objectives to be expressed [54]. Secondly, and most importantly, a convexity problem can arise when the lower boundary of the feasible region for the objective function is non-convex, in which case, the set of non-inferior solutions between A and B is not available. The non-convexity problem is illustrated schematically in Fig. 6.4, which pertains to the minimization of a bi-objective cost function.

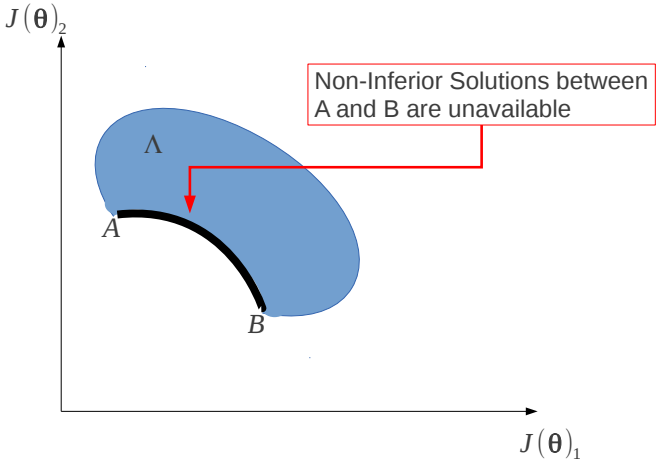

FIGURE 6.4: Illustration of the non-convexity problem that may be encountered when using the weighted sum method for multi objective minimization [14].

In the context of this research, multi-objective minimization will be performed on a bi-objective cost function, when identifying active tension and Windkessel material parameters. Hence, the construction of a generic bi-objective cost function, using the weighted sum approach will be detailed. Firstly, consider the objective function, $\left[J^2(\boldsymbol{\theta})\right]_1$, constructed from the sum of the squares of the difference between an experimentally observed vector, $\hat{\mathbf{f}}$, and a corresponding simulated vector, $\mathbf{f}(\boldsymbol{\theta})$, i.e.

$$\left[J^2(\boldsymbol{\theta})\right]_1 = \frac{1}{2}\sum_{i=0}^{i=N-1}\left(\frac{\hat{\mathrm{f}}_i - \mathrm{f}_i(\boldsymbol{\theta})}{\mathrm{w}_i^{\hat{\mathbf{f}}}}\right)^2, \tag{6.9}$$

where $i \in (0, N-1)$ and $\hat{\mathrm{f}}_i$ are the components of the experimental vector, $\hat{\mathbf{f}}$. Also, $\mathrm{f}(\boldsymbol{\theta})_i$ are the components of the simulated vector, $\mathbf{f}(\boldsymbol{\theta})$. In addition, $\frac{1}{\left(\mathrm{w}_i^{\hat{\mathbf{f}}}\right)^2}$ are diagonal components of a weight matrix, $\mathbf{W}^{\hat{\mathbf{f}}}$, whose off-diagonal terms are all zero. In addition, N again represents the number of data points. Alternatively, Eq. (6.9) can be expressed

in vector-matrix notation as

$$\left[J^2(\boldsymbol{\theta})\right]_1 = \frac{1}{2}\left[\left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{f}}} \left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)\right]. \tag{6.10}$$

Next, consider another separate objective function, $\left[J^2(\boldsymbol{\theta})\right]_2$, constructed from the sum of the squares of the difference between an experimentally observed vector, $\hat{\mathbf{g}}$, and a corresponding simulated vector, $\mathbf{g}(\boldsymbol{\theta})$

$$\left[J^2(\boldsymbol{\theta})\right]_2 = \frac{1}{2}\sum_{i=0}^{i=N-1}\left(\frac{\hat{\mathrm{g}}_i - \mathrm{g}_i(\boldsymbol{\theta})}{\mathrm{w}_i^{\hat{\mathbf{g}}}}\right)^2, \tag{6.11}$$

where, $\hat{\mathrm{g}}_i$ are the components of the experimental vector, $\hat{\mathbf{g}}$, and $\mathrm{g}(\boldsymbol{\theta})_i$ are the components of the simulated vector, $\mathbf{g}(\boldsymbol{\theta})$. Additionally, $\frac{1}{\left(\mathrm{w}_i^{\hat{\mathbf{g}}}\right)^2}$ are diagonal components of a weight matrix, $\mathbf{W}^{\hat{\mathbf{g}}}$, whose off-diagonal terms are all zero. Eq. (6.11) can be expressed in vector-matrix notation as

$$\left[J^2(\boldsymbol{\theta})\right]_2 = \frac{1}{2}\left[\left(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{g}}} \left(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta})\right)\right]. \tag{6.12}$$

Consequently, a bi-objective cost function, $J^2(\boldsymbol{\theta})$, can be constructed as the sum of the weighted objective functions, $\omega_1\left[J^2(\boldsymbol{\theta})\right]_1$ and $\omega_2\left[J^2(\boldsymbol{\theta})\right]_2$, yielding,

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\left[\omega_1\left[\left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{f}}} \left(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})\right)\right] + \omega_2\left[\left(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{g}}} \left(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta})\right)\right]\right], \tag{6.13}$$

where ω_1 and ω_2 are the weights of the objective functions $\left[J^2(\boldsymbol{\theta})\right]_1$ and $\left[J^2(\boldsymbol{\theta})\right]_2$ respectively. Eq. (6.13) can be alternatively expressed in index notation as

$$\begin{aligned} J^2(\boldsymbol{\theta}) =& \frac{1}{2}\omega_1 \sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}\left[\left(\hat{\mathrm{f}}_i - \mathrm{f}_i(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathrm{W}_{ij}^{\hat{\mathbf{f}}} \left(\hat{\mathrm{f}}_j - \mathrm{f}_j(\boldsymbol{\theta})\right)\right] \\ &+ \frac{1}{2}\omega_2 \sum_{i=0}^{i=N-1}\sum_{j=0}^{j=N-1}\left[\left(\hat{\mathrm{g}}_i - \mathrm{g}_i(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathrm{W}_{ij}^{\hat{\mathbf{g}}} \left(\hat{\mathrm{g}}_j - \mathrm{g}_j(\boldsymbol{\theta})\right)\right], \end{aligned} \tag{6.14}$$

where, $i, j \in (0, N-1)$.

6.3.2.3 The ϵ-Constraint Method

A method that overcomes the non-convexity problem associated with the weighted sum method, is the so-called ϵ-constraint method. This method involves minimization of a primary objective, $\left[J^2(\boldsymbol{\theta})\right]_p$, and expressing the other objective functions in terms of inequality constraints [54]. Consequently, a mathematical minimization problem can be

constructed as

$$\begin{aligned} &\text{minimize} \quad \left[J^2(\boldsymbol{\theta})\right]_p, \\ &\text{subject to} \quad \left[J^2(\boldsymbol{\theta})\right]_i \leq \epsilon \quad \text{for} \quad i \neq p, \\ &\text{subject to } \boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta} \leq \boldsymbol{\theta}^{\max}. \end{aligned} \tag{6.15}$$

Fig. 6.5 is a schematic representation of the ϵ-constraint method for the case of a bi-objective cost function, $[\boldsymbol{J}(\boldsymbol{\theta})]^{\mathrm{T}} = \left[\left[J^2(\boldsymbol{\theta})\right]_1, \left[J^2(\boldsymbol{\theta})\right]_2\right]$.

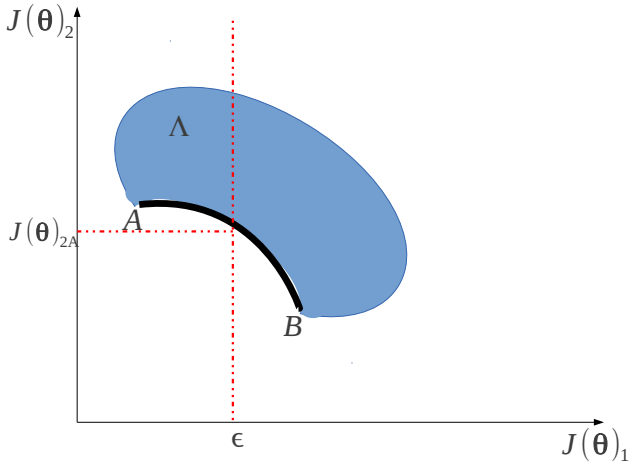

FIGURE 6.5: Illustration of the ϵ-constraint method for multi objective minimization [14].

Referring to Fig. 6.5, it is clear that using the ϵ-constraint method, the set of non-inferior solutions between A and B is now available for the non-convex problem, which is not the case for the weighted sum method [14]. In addition, different non-inferior solutions can be obtained using different constraints for the parameter ϵ [14].

The ϵ-constraint method also has certain short-comings [14]. The major short-coming of this method is that it requires an appropriate choice of the parameter ϵ, in order for a feasible solution to be obtained, which may be difficult in the case of lack of insight into an optimization problem. Furthermore, implementation of the ϵ-constraint method requires knowledge of a branch of statistics known as decision theory or game theory [14], which are topics that fall outside the scope of this thesis. However, the ability of the method to handle the non-convexity problem makes it a possible area for further research. Finally, the chosen method for multi-objective cost function minimization, in the context of this research, is the popular weighted sum approach.

6.4 Material Parameter Identification

Material parameter identification via an inverse approach, usually involves the use of an optimization algorithm [20]. These optimization algorithms can be split into two broad categories:

- Gradient based algorithms [23].
- Non-gradient based algorithms [14, 23].

Gradient based algorithms, are usually associated with faster convergence towards a parameter set that minimizes a cost function. However, these methods do not guarantee convergence towards a global minima and are thus reliant on the initial starting parameter. Examples of gradient-based optimizers include Steepest descent, Gauss-Newton and Levenberg-Marquardt algorithms, among many others [23]. Next, non-gradient based algorithms tend to display slower convergence towards a parameter set that minimizes a cost function and this is usually attributed to the fact that a large initial population size, of different parameter sets over a search space is required [14]. However, these optimizers are reputed to be more adept at handling noisy experimental data, than gradient based optimizers and are also more efficient means of finding a global minima in a search space [14]. Examples of non-gradient based algorithms include Genetic Search and Particle Swarm algorithms, among many others. For the purposes of this research, a gradient based optimizer based on the Levenberg-Marquardt algorithm is considered as a means of minimizing cost functions, in order to obtain desired material parameters. This choice is guided by the fact that the algorithm is devoted to non-linear least squares curve fitting applications [20].

The implementation of the Levenberg-Marquardt algorithm is however not the conventional formulation, originally suggested by Levenberg [52] and later Marquardt [55], as these formulations were geared mainly to the solution of unconstrained non-linear least squares minimization problems. In this research, it is desired that parameters be restricted to a feasible space, thus requiring a constrained minimization approach. Levenberg-Marquardt type algorithms for constrained optimization have been presented in the literature [24, 25]. Kanzow et al. [24], have outlined two such methods. The first approach solves a strictly convex minimization problem at each iteration, whereas the second method, which is referred to as a projected Levenberg-Marquardt algorithm, only solves one equation system at each iteration. Both methods have been shown to converge near quadratically under an error bound assumption, and can be globalized in an easy manner [24]. The projected Levenberg-Marquardt method, requires a significantly lower

amount of computational effort, provided that constraints can be applied efficiently [24]. In light of these remarks, the projected Levenberg-Marquardt algorithm is selected as the non-linear least squares minimization algorithm. Before the algorithm is outlined, it should be noted that k and t refer to the major and minor iteration number, respectively and are both initialized to zero at the start of the optimization. Next, a general framework for the algorithm described by Kanzow et al. [24] is presented below:

- Evaluate the cost function, $J^2(\boldsymbol{\theta}^k)$, at an initial parameter vector, $\boldsymbol{\theta}^k$.
- Compute the damping parameter, μ^k.
- Evaluate the gradient of the cost function, as well as the gradient of the simulated vector, i.e., $\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)$ and $\boldsymbol{\nabla}\mathbf{f}(\boldsymbol{\theta}^k)$ (and $\boldsymbol{\nabla}\mathbf{g}(\boldsymbol{\theta}^k)$ for the case of bi-objective cost functions), respectively.
- Compute a search direction, $\delta\boldsymbol{\theta}_u^k$, based on an unconstrained system of equations.
- Apply parameter constraints, using an appropriate projection (P), i.e., $\boldsymbol{\theta}^{k+1} = \mathrm{P}\left[\boldsymbol{\theta}^k + \delta\boldsymbol{\theta}_u^k\right]$, such that the new parameter set is within the feasible space ($\boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta}^{k+1} \leq \boldsymbol{\theta}^{\max}$).
- Evaluate the cost function at the new parameter set, $J^2(\boldsymbol{\theta}^{k+1})$.
- Compute the descent, $\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k$, with $\delta\boldsymbol{\theta}^k = \mathrm{P}\left[\boldsymbol{\theta}^k + \delta\boldsymbol{\theta}_u^k\right] - \boldsymbol{\theta}^k$.
- If the descending condition is satisfied (i.e. $J^2(\boldsymbol{\theta}^{k+1}) < J^2(\boldsymbol{\theta}^k)$), then accept the new parameter vector and continue to the next major iteration ($k := k + 1$).
- If the descending condition is not satisfied, but the descent condition is not violated (i.e. $\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k < 0$), then adjust the search direction, $\delta\boldsymbol{\theta}^k$, using a line search to find a step size, α, that produces an acceptable reduction in the cost function. After the line search, the minor iteration number, t, which in this case represents the number of required line search iterations, is set to zero. Furthermore, the new parameters are accepted and the algorithm proceeds to the next major iteration ($k := k + 1$).
- If both the descending and the descent conditions are violated, then a projected gradient search is initiated, to produce an acceptable reduction in the cost function. After the projected gradient search, the minor iteration number, t, which in this case represents the number of required projected gradients steps, is set to zero. Furthermore, the new parameters are accepted and the algorithm proceeds to the next major iteration ($k := k + 1$).

- Terminate the algorithm if the convergence criteria for the cost function, cost function gradient vector or parameter increment vector is satisfied, i.e., $J^2(\boldsymbol{\theta}^{k+1}) < \epsilon_1$, $||\boldsymbol{\nabla} J^2(\boldsymbol{\theta})|| < \epsilon_2$ and $\frac{||\delta\boldsymbol{\theta}^k||}{||\boldsymbol{\theta}^k||} < \epsilon_3$, where, ϵ_1, ϵ_2, and ϵ_3 are user defined. The algorithm is also terminated if the number of major iterations, k, exceeds a user defined maximum, k_{max}.

The succeeding sections will consider some of the numerical aspects of the projected Levenberg-Marquardt algorithm, that were mentioned above. Furthermore, some additional numerical and conceptual complexities of the algorithm will be outlined.

6.4.1 Computation of the Search Direction

In the previous section, it was illustrated that at the core of the projected Levenberg-Marquardt algorithm is the computation of a search direction based on an unconstrained minimization problem, i.e., $\delta\boldsymbol{\theta}_u^k$. In the following sections this search direction will be derived for the following two cases:

- Single objective cost function minimization [24, 52, 55, 56].
- Bi-objective cost function minimization, with cost functions constructed using the weighted sum approach [24, 52, 55, 56].

In the following sections, the symbol, k, which denotes the major iteration number has been omitted from the derivations, for the sake of clarity.

6.4.1.1 Single Objective Cost function Minimization

Consider the generic single objective cost function in Eq. (6.2). The Levenberg-Marquardt algorithm flexibly updates the material parameters, between the Steepest descent and Gauss-Newton update methods [56]. The Steepest descent method, is a general minimization technique, which updates parameter values in the direction opposite to the gradient of the cost function. The general equation for the gradient of the single objective cost function is

$$\frac{\partial J^2(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \boldsymbol{\nabla} J^2(\boldsymbol{\theta}) = -(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{f}}} \mathbf{J}^{\mathbf{f}}, \tag{6.16}$$

where

$$\mathbf{J}^{\mathbf{f}} = \boldsymbol{\nabla} \mathbf{f}(\boldsymbol{\theta}) = \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{6.17}$$

$\mathbf{J^f}$ denotes the Jacobian matrix, representing the sensitivity of the simulated vector, $\mathbf{f}(\boldsymbol{\theta})$, to perturbations in the material parameters. The material parameter perturbation, $\delta\boldsymbol{\theta}$, that moves parameters in the direction of steepest descent, is given by the negative cost function gradient, which yields,

$$\delta\boldsymbol{\theta} = -\alpha\boldsymbol{\nabla}J^2(\boldsymbol{\theta}) = \alpha\left[(\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))\right], \tag{6.18}$$

where α determines the length of the step in the steepest descent direction. Next, the Gauss-Newton method presumes that the cost function (Eq. (6.2)), is quadratic in the parameters near the optimal solution. The simulated vector with perturbed parameters, $\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$, is first locally approximated through a first-order Taylor series expansion, yielding,

$$\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \mathbf{f}(\boldsymbol{\theta}) + \mathbf{J^f}\delta\boldsymbol{\theta}. \tag{6.19}$$

Then, replacing $\mathbf{f}(\boldsymbol{\theta})$ in Eq. (6.2), with the approximation for the perturbed vector, $\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$ (Eq. (6.19)), the cost function value with perturbed parameters, $J^2(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$, is found to be

$$\begin{aligned} L(\delta\boldsymbol{\theta}) = J^2(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) = &\frac{1}{2}\hat{\mathbf{f}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\hat{\mathbf{f}} + \frac{1}{2}[\mathbf{f}(\boldsymbol{\theta})]^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{f}(\boldsymbol{\theta}) \\ &- \hat{\mathbf{f}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{f}(\boldsymbol{\theta}) - \left[(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}\right]\delta\boldsymbol{\theta} \\ &+ \frac{1}{2}(\delta\boldsymbol{\theta})^{\mathrm{T}}\left[(\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}\right]\delta\boldsymbol{\theta}. \end{aligned} \tag{6.20}$$

Eq. (6.20), illustrates that the cost function with perturbed parameters, is quadratic with regards to the perturbation, $\delta\boldsymbol{\theta}$ and that the approximation of the Hessian matrix, $\mathbf{H}$, is equal to $[(\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}]$. Next, the derivative of $L(\delta\boldsymbol{\theta})$ with respect to the perturbation, $\delta\boldsymbol{\theta}$, is derived as

$$\frac{\partial L(\delta\boldsymbol{\theta})}{\partial(\delta\boldsymbol{\theta})} = -\left[(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}\right] + (\delta\boldsymbol{\theta})^{\mathrm{T}}\left[(\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}\right]. \tag{6.21}$$

Therefore, the perturbation, $\delta\boldsymbol{\theta}$, that minimizes the cost function is found by setting Eq. (6.21) equal to zero. Consequently, the resulting normal equations for the Gauss-Newton parameter perturbation are

$$\left[(\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J^f}\right]\delta\boldsymbol{\theta} = (\mathbf{J^f})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})). \tag{6.22}$$

Furthermore, the Levenberg-Marquardt algorithm adaptively varies the parameter updates between the Steepest descent and Gauss-Newton updates via

$$\left(\left[(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}}\right] + \mu\mathbf{I}\right)\delta\boldsymbol{\theta} = (\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})), \tag{6.23}$$

where $\mathbf{I}$ is the identity matrix. Also, it can be noted that small values of the algorithmic parameter, μ, result in a approximate Gauss-Newton update and large values of μ, result in a approximate Steepest descent update. At a large distance from the cost function minimum, the steepest descent method is utilized to provide steady and convergent progress toward the solution. As the solution approaches the minimum, μ is adaptively decreased, and the Levenberg-Marquardt method approaches the Gauss-Newton method, and the solution typically converges rapidly to the local minimum [56]. The manner in which the algorithmic parameter, μ, is updated, is one of the numerical complexities of the Levenberg-Marquardt algorithm, and a section later in this chapter will be devoted solely to this topic. Next, Eq. (6.23), is simplified to the following system of linear equations:

$$\mathbf{A}\delta\boldsymbol{\theta} = \mathbf{b}. \tag{6.24}$$

Provided that the computed Jacobian matrix has full rank, then the matrix, $\mathbf{A}$, is positive definite symmetric, consequently, the solved for parameter increment, $\delta\boldsymbol{\theta}$, is a search direction (satisfying the descent condition, $\left[\boldsymbol{\nabla}J^2(\boldsymbol{\theta}^{\mathrm{k}})\right]^{\mathrm{T}}\delta\boldsymbol{\theta}^{\mathrm{k}} < 0$). Hence, for some small positive step along the search direction, a reduction in the objective function is provided. The equation system (Eq. (6.24)) is usually solved using methods such as the LU factorization. In this regard, the author takes advantage of the PETSc equation solver libraries, which are integrated in SESKA. Nevertheless, the search direction is derived. Finally, this search direction is a solution to an unconstrained minimization problem, i.e., $\delta\boldsymbol{\theta} = \delta\boldsymbol{\theta}_u$. The subscript, u, is employed to emphasise that the descent direction is the solution to an unconstrained problem.

6.4.1.2 Minimization of Bi-Objective Cost Functions Constructed using the Weighted Sum Approach

Consider the generic bi-objective cost function in Eq. (6.13), which has been constructed using the weighted sum method. The general equation for the gradient of this cost function is

$$\frac{\partial J^2(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} = \boldsymbol{\nabla}J^2(\boldsymbol{\theta}) = -\omega_1(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}} - \omega_2(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}, \tag{6.25}$$

where

$$\mathbf{J}^{\mathbf{f}} = \boldsymbol{\nabla}\mathbf{f}(\boldsymbol{\theta}) = \frac{\partial \mathbf{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \tag{6.26}$$

and

$$\mathbf{J}^{\mathbf{g}} = \boldsymbol{\nabla}\mathbf{g}(\boldsymbol{\theta}) = \frac{\partial \mathbf{g}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}. \tag{6.27}$$

$\mathbf{J}^{\mathbf{f}}$ and $\mathbf{J}^{\mathbf{g}}$ denote the first and second Jacobian matrices, representing the sensitivity of the vector fields, $\mathbf{f}(\boldsymbol{\theta})$ and $\mathbf{g}(\boldsymbol{\theta})$, to perturbations in the material parameters. The material parameter perturbation, $\delta\boldsymbol{\theta}$, that moves parameters in the direction of steepest descent, is given by the negative cost function gradient

$$\delta\boldsymbol{\theta} = -\alpha\boldsymbol{\nabla}J^2(\boldsymbol{\theta}) = \alpha[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta})) + \omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta}))], \tag{6.28}$$

where α again determines the length of the step in the steepest descent direction. To obtain the normal equations for the Gauss-Newton method, the first simulated vector, with perturbed material parameters, $\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$, is again locally approximated through a first-order Taylor series expansion, yielding,

$$\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \mathbf{f}(\boldsymbol{\theta}) + \mathbf{J}^{\mathbf{f}}\delta\boldsymbol{\theta}. \tag{6.29}$$

Similarly, the second simulated vector, evaluated with perturbed material parameters, $\mathbf{g}(\boldsymbol{\theta}+\delta\boldsymbol{\theta})$, may also be locally approximated through a first-order Taylor series expansion, via

$$\mathbf{g}(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) \approx \mathbf{g}(\boldsymbol{\theta}) + \mathbf{J}^{\mathbf{g}}\delta\boldsymbol{\theta}. \tag{6.30}$$

Next, $\mathbf{f}(\boldsymbol{\theta})$ and $\mathbf{g}(\boldsymbol{\theta})$ in Eq. (6.13), are replaced by the approximations for $\mathbf{f}(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$ and $\mathbf{g}(\boldsymbol{\theta} + \delta\boldsymbol{\theta})$ (Eq. (6.29) and Eq. (6.30), respectively), yielding the cost function value with perturbed parameters, which is given by

$$\begin{aligned} L(\delta\boldsymbol{\theta}) = J^2(\boldsymbol{\theta} + \delta\boldsymbol{\theta}) =& \frac{\omega_1}{2}\hat{\mathbf{f}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\hat{\mathbf{f}} + \frac{\omega_2}{2}\hat{\mathbf{g}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\hat{\mathbf{g}} + \frac{\omega_1}{2}[\mathbf{f}(\boldsymbol{\theta})]^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{f}(\boldsymbol{\theta}) \\ &+ \frac{\omega_2}{2}[\mathbf{g}(\boldsymbol{\theta})]^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{g}(\boldsymbol{\theta}) - \omega_1\hat{\mathbf{f}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{f}(\boldsymbol{\theta}) - \omega_2\hat{\mathbf{g}}^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{g}(\boldsymbol{\theta}) \\ &- \left[\omega_1(\hat{\mathbf{f}} - \mathbf{f}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}} + \omega_2(\hat{\mathbf{g}} - \mathbf{g}(\boldsymbol{\theta}))^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right]\delta\boldsymbol{\theta} \\ &+ \frac{1}{2}(\delta\boldsymbol{\theta})^{\mathrm{T}}\left[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}} + \omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right]\delta\boldsymbol{\theta}. \end{aligned} \tag{6.31}$$

Eq. (6.31), illustrates once again that the cost function is quadratic with regards to the perturbation, $\delta\boldsymbol{\theta}$, and that the approximation of the Hessian matrix, $\mathbf{H}$, is equal to $[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}} + \omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}]$. Next, the derivative of $L(\delta\boldsymbol{\theta})$ with respect to the

perturbation, $\delta\boldsymbol{\theta}$, is found to be

$$\begin{aligned}\frac{\partial L(\delta\boldsymbol{\theta})}{\partial(\delta\boldsymbol{\theta})} = &-\left[\omega_1(\hat{\mathbf{f}}-\mathbf{f}(\boldsymbol{\theta}))\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}}+\omega_2(\hat{\mathbf{g}}-\mathbf{g}(\boldsymbol{\theta}))\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right]\\ &+(\delta\boldsymbol{\theta})^{\mathrm{T}}\left[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}}+\omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right].\end{aligned} \tag{6.32}$$

The perturbation, $\delta\boldsymbol{\theta}$, that minimizes the cost function is found by setting Eq. (6.32) equal to zero. Consequently, the resulting normal equations for the Gauss-Newton parameter perturbation are

$$\begin{aligned}\left[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}}+\omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right]\delta\boldsymbol{\theta} =&\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}}-\mathbf{f}(\boldsymbol{\theta}))\\ &+\omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}(\hat{\mathbf{g}}-\mathbf{g}(\boldsymbol{\theta})).\end{aligned} \tag{6.33}$$

As was the case for the single objective minimization, the Levenberg-Marquardt algorithm adaptively varies the parameter updates, between the Steepest descent and Gauss-Newton updates via

$$\begin{aligned}\left(\left[\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}\mathbf{J}^{\mathbf{f}}+\omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}\mathbf{J}^{\mathbf{g}}\right]+\mu\mathbf{I}\right)\delta\boldsymbol{\theta} =&\omega_1(\mathbf{J}^{\mathbf{f}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{f}}}(\hat{\mathbf{f}}-\mathbf{f}(\boldsymbol{\theta}))\\ &+\omega_2(\mathbf{J}^{\mathbf{g}})^{\mathrm{T}}\mathbf{W}^{\hat{\mathbf{g}}}(\hat{\mathbf{g}}-\mathbf{g}(\boldsymbol{\theta})).\end{aligned} \tag{6.34}$$

Eq. (6.34) can be simplified to the following set of linear equations:

$$\mathbf{A}\delta\boldsymbol{\theta}=\mathbf{b}. \tag{6.35}$$

The equation system is solved using the LU factorization yielding a parameter perturbation, $\delta\boldsymbol{\theta}$, which is again the solution to an unconstrained minimization problem, i.e., $\delta\boldsymbol{\theta}=\delta\boldsymbol{\theta}_{\mathrm{u}}$. Finally, provided that the Jacobian matrices have full rank, the matrix, $\mathbf{A}$, is positive definite symmetric, which means that once again the solved for parameter perturbation is in fact a search direction.

6.4.2 Parameter Scaling

The issue of parameter scaling is crucial, especially when the parameters to be identified differ in magnitude, perhaps even by several orders. Firstly, Eq. (6.23) and Eq. (6.34), from the previous two sections can be represented in general form by

$$\left[\mathbf{H}^k+\mu\mathbf{D}^k\right]\delta\boldsymbol{\theta}_u^k=-\boldsymbol{\nabla}J^2(\boldsymbol{\theta}^k), \tag{6.36}$$

where $\mathbf{H}^k$ is the Hessian matrix, $\mathbf{D}^k$ is the so-called damping matrix, $\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)$ is the cost function gradient vector and k refers to the major iteration number. In the current derivations (Eq. (6.23) and Eq. (6.34)), the damping matrix is in the form of the identity matrix, $\mathbf{I}$, which is the recommendation initially put forward by Levenberg [52]. However, when an initial parameter set is far from the required minima, this definition of the damping matrix can cause sluggish behaviour of the Levenberg-Marquardt algorithm. This behaviour is most noticeable when the parameter values differ significantly in scale and arises because the properties of steepest descent do not show invariance to parameter scale [15]. Consequently, in order to improve the algorithm's convergence, Marquardt [55] suggested that the damping matrix should have diagonal components, equal to the diagonal components of the Hessian matrix, i.e.

$$\mathrm{D}_{ij}^k = \begin{cases} \mathrm{H}_{ij}^k \text{ for } i = j, \\ 0 \text{ for } i \neq j. \end{cases} \tag{6.37}$$

Marquardt's suggested damping matrix essentially introduces scale invariance, which means that, if parameters of the fit are replaced by $\hat{\theta}_i = \lambda_i \theta_i$, for some scaling factors λ_i, then the sequence of iterates (values of the cost function) would be left unchanged [15]. The damping matrix proposed by Marquardt is observed to improve convergence, however, the introduced scale invariance can result in the parameter identification being susceptible to parameter evaporation [15]. Transtrum and Sethna [15] provided a very accessible description of the idea of parameter evaporation. Basically, this phenomenon can occur when parameters encounter a ridge or plateau in a cost function surface (refer to Fig. 6.6), characterised by small gradients. Hence, algorithms lost on such a plateau, will push parameters to infinite values. Such solutions correspond to saddle points or local minima at infinite parameter values and thus have little or no meaning.

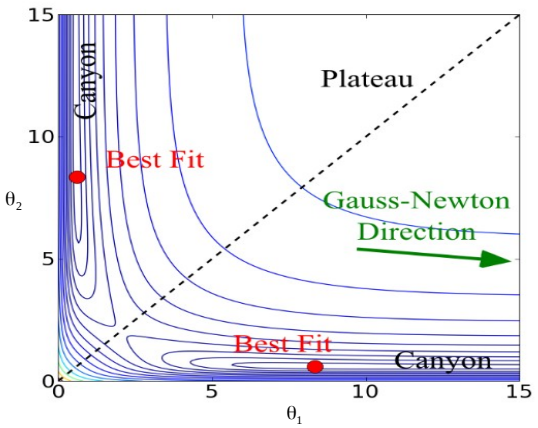

FIGURE 6.6: Illustration of a ridge or plateau in a cost function surface of two variables θ_1 and θ_2 [15].

Consequently, a more robust selection for the damping matrix was later put forward by Móre's [57]. Móre's, recognized that with Marquardt's definition of the damping matrix, the algorithm can take unreliable parameter steps, when the cost function gradient is approaching zero [57]. Thus, Móre's suggested that diagonal components of the damping matrix, be chosen as the maximum values of the diagonal elements of the Hessian matrix yet encountered, yielding the following formulation:

$$\mathrm{D}_{ij}^{k} = \begin{cases} \max\{\mathrm{H}_{ij}^{k}, \mathrm{H}_{ij}^{k-1}\} \text{ for } i = j, \\ 0 \text{ for } i \neq j. \end{cases} \tag{6.38}$$

Móre's formulation of the damping matrix is more robust at avoiding parameter evaporation, and still retains the fast convergence behaviour exhibited using Marquardt's suggestion. Hence, in this work, the damping matrix in Eq. (6.36) is computed using Eq. (6.38.), following the recommendations made by Móre.

6.4.3 Application of Parameter Constraints

As an initial starting point for the implementation of the Levenberg-Marquardt algorithm, the feasible parameter space ($\boldsymbol{\theta}^{\min} \leq \boldsymbol{\theta}^{k+1} \leq \boldsymbol{\theta}^{\max}$), has been described using simple box constraints. In this case, parameters obtained by minimizing the cost function, are bounded by a hyperbox $\left[\boldsymbol{\theta}^{\min}, \boldsymbol{\theta}^{\max}\right]$. Hence, the goal of parameter identification, is to generate a sequence of candidate solutions for $\boldsymbol{\theta}$, that all strictly lie within the confines described by this hyperbox. Since the solution for the parameter perturbation, $\delta\boldsymbol{\theta}_u$, at each Levenberg-Marquardt iteration, k, is a solution to an unconstrained minimization problem, additional box constraint handling mechanisms are required to restrict the parameter vector, to lie within the feasible space represented by the hyperbox. Kanzow et al. [24] suggested that parameters that attempt to violate their constraints, simply need to be projected onto the feasible space. When the feasible space is described by box constraints, this projection can be carried out efficiently, with very little computational effort [24]. The ease with which the projection can be carried out in the case of box constraints, is due to the fact that the required mapping, is simply a Euclidean projection described by

$$\mathrm{P}\left[\left(\theta_u^{k+1}\right)_i\right] = \begin{cases} \theta_i^k + \left(\delta\theta_u^k\right)_i \text{ for } \theta_i^{\min} \leq (\theta_u)_i^{k+1} \leq \theta_i^{\max}, \\ \theta_i^{\max} \text{ for } (\theta_u)_i^{k+1} > \theta_i^{\max}, \\ \theta_i^{\min} \text{ for } (\theta_u)_i^{k+1} < \theta_i^{\min}, \end{cases} \tag{6.39}$$

where, $\left(\theta_u^{k+1}\right)_i$ is the solution for the parameter vector components, based on the unconstrained minimization problem (i.e $\left(\theta_u^{k+1}\right)_i = \theta_i^k + \left(\delta\theta_u^k\right)_i$). Also, $\mathrm{P}\left[\left(\theta_u^{k+1}\right)_i\right]$, represents

the Euclidean projection of the unconstrained parameter solution, onto the feasible parameter space, i.e., θ_i^{k+1}. Eq. (6.39) can be expressed in general form as

$$\theta_i^{k+1} = \min\left[\max\left(\theta_i^{\min}, \left(\theta_u^{k+1}\right)_i\right), \theta_i^{\max}\right]. \tag{6.40}$$

In the case where box constraints have been applied, the descent direction is recomputed, using the constrained set of new material parameters as

$$\delta\theta_i^k = \theta_i^{k+1} - \theta_i^k. \tag{6.41}$$

A draw back of the projection of parameters onto the feasible space, is the potential loss of descent direction, which corresponds to $\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k > 0$. This situation is rare, but can occur, consequently, Kanzow et al. [24] suggested that this situation be remedied by using a projected gradient search technique. The method for conducting the projected gradient search will be explored in a later section of this chapter.

Based on the new parameter obtained from Eq. (6.40), the cost function, $J^2(\boldsymbol{\theta}^{k+1})$, is computed from Eq. (6.2) for the single objective case or Eq. (6.13) for the bi-objective case. The parameters are accepted and the next iteration of the projected Levenberg-Marquardt algorithm is commenced (i.e. $k := k + 1$), if the following conditions are satisfied:

$$J^2(\boldsymbol{\theta}^{k+1}) < J^2(\boldsymbol{\theta}^k), \tag{6.42}$$

$$\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k < 0, \tag{6.43}$$

where, $\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k < 0$ is the descent condition, and $J^2(\boldsymbol{\theta}^{k+1}) < J^2(\boldsymbol{\theta}^k)$ is the descending condition. The gradient of the cost function at the initial parameters, $\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)$, is computed from Eq. (6.16) for the single objective case or Eq. (6.25) for the bi-objective case. The descent condition is automatically satisfied if a new parameter has brought about a reduction in the cost function (i.e. when the descending condition is satisfied). If the above conditions are not met, then either a line search or projected gradient search is initiated, the nature of which will be explored in the subsequent sections.

6.4.4 Line Search Procedures

The conditions under which a line search iteration is initiated are

$$J^2(\boldsymbol{\theta}^{k+1}) \geq J^2(\boldsymbol{\theta}^k), \tag{6.44}$$

$$\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k < 0, \tag{6.45}$$

where, $\delta\boldsymbol{\theta}^k$ is again the potentially adjusted search direction computed using Eq. (6.41). Hence, it is clear that the line search is initiated only when the descending condition is violated, provided that the descent condition is still satisfied. Numerous line search procedures are available, however, two popular techniques are listed below:

- Cubic interpolation [14].
- Inexact backtracking Armijo type line search [24].

The above line search techniques are explored in the following sections, as both are implemented in the in-house modelling software SESKA and have both been used in this work. The inexact backtracking Armijo type line search is chosen for implementation, as it is recommended by Kanzow et al. [24] in their implementation of the projected Levenberg-Marquardt algorithm. In addition, the cubic line search is implemented as a possible alternative, following Matlab's implementation of the Levenberg-Marquardt algorithm for non-linear least squares curve fitting, i.e. the Matlab function *lsqnonlin* [14].

6.4.4.1 Cubic Interpolation

Cubic interpolation involves an approximation of the cost function behaviour, over the range of the descent direction, $\delta\boldsymbol{\theta}^k$, using a uni-variant cubic polynomial of the following form:

$$J^2(\alpha) = a\alpha^3 + b\alpha^2 + c\alpha + d, \tag{6.46}$$

where, a, b, c and d are parameters of the fit. The gradient of the uni-variant fit is given by the following quadratic equation:

$$\frac{dJ^2(\alpha)}{d\alpha} = 3a\alpha^2 + 2b\alpha + c. \tag{6.47}$$

Furthermore, the second derivative of the cubic polynomial is

$$\frac{d(J^2(\alpha))^2}{d^2\alpha} = 6a\alpha + 2b. \tag{6.48}$$

In addition, α represents the step size, with $J^2(\alpha = 0)$ corresponding to $J^2(\boldsymbol{\theta}^k)$. In order to determine parameters of the uni-variant cubic fit, additional information is required, pertaining to the behaviour of the cost function over the range of the descent direction. The gradient of the cost function at $\boldsymbol{\theta}^k$, is readily available, as this was required in order to compute the search direction. Using this gradient, which is a vector, the gradient of the uni-variant cubic fit at $\alpha = 0$ (α_0), can be determined exactly at no significant computational cost, using the following relation:

$$\frac{dJ^2(\alpha)}{d\alpha}|_{\alpha_0} = \left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k. \tag{6.49}$$

However, one additional condition is required in order to be able to fully characterize the cubic polynomial fit. For this, it is recommended that the gradient of the uni-variant fit, is evaluated at the step size, α_{k+1}, corresponding to the set of parameters, $\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha_{k+1}\delta\boldsymbol{\theta}^k$. It should be noted that for the first iteration of the line search, α_{k+1} is set to unity. The gradient of the uni-variant fit is evaluated using forward differences as

$$\frac{dJ^2(\alpha)}{d\alpha}|_{\alpha_{k+1}} = \frac{J^2([\eta+1]\,\boldsymbol{\theta}^{k+1}) - J^2(\boldsymbol{\theta}^{k+1})}{\eta}, \tag{6.50}$$

where η is a small positive scalar, say 10^{-6}. Referring to Eq. (6.50), in order to evaluate this gradient at α_{k+1}, the simulation needs to be run once in order to compute the cost functional at the scaled parameter vector, $[\eta+1]\,\boldsymbol{\theta}^{k+1}$, i.e., $J^2([\eta+1]\,\boldsymbol{\theta}^{k+1})$. It should be noted that the value of the cost function, $J^2(\boldsymbol{\theta}^{k+1})$, is already available at each line search iteration, and does not need to be recomputed. Now, all the information is available to determine the parameters of the fit for the uni-variant cubic polynomial. These parameters are computed by solving the following system of linear equations (usually with LU factorization):

$$\begin{bmatrix} J^2(\alpha_0) \\ J^2(\alpha_{k+1}) \\ \frac{dJ^2(\alpha)}{d\alpha}|_{\alpha_0} \\ \frac{dJ^2(\alpha)}{d\alpha}|_{\alpha_{k+1}} \end{bmatrix} = \begin{bmatrix} \alpha_0^3 & \alpha_0^2 & \alpha_0 & 1 \\ \alpha_{k+1}^3 & \alpha_{k+1}^2 & \alpha_{k+1} & 1 \\ 3\alpha_0^2 & 2\alpha_0 & 1 & 0 \\ 3\alpha_{k+1}^2 & 2\alpha_{k+1} & 1 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix}. \tag{6.51}$$

Next, the extremum of the cubic polynomial, are determined by finding the roots of the gradient of the uni-variant fit (Eq. (6.47)), using the quadratic formula

$$\alpha_{1,2} = \frac{-2b \pm \sqrt{4b^2 - 12ac}}{6a}. \tag{6.52}$$

To find the roots that produce a minimum extrema, the roots, α_1 and α_2, are subjected to the second derivative test. The root, α_i (wher $i \in (1, 2)$), that produces a positive value in Eq. (6.48), is that which produces a minimum, and is thus the selected step size. Once, a suitable step size, α, is chosen based on the above procedure, a new material parameter set is computed as

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha\delta\boldsymbol{\theta}^k. \tag{6.53}$$

The cost function is then evaluated at the new material parameter set, i.e., $J^2(\boldsymbol{\theta}^{k+1})$. If the new cost function value is less than the old ($J^2(\boldsymbol{\theta}^k)$), then the new parameter set is accepted, the line search is terminated, and the next iteration of the projected Levenberg-Marquardt algorithm is commenced. However, if the computed step size does not bring about a desired reduction of the cost function at the new parameter set, then the next line search iteration is begun, with $\alpha_{k+1} = \alpha$. In addition, the cost function value produced from the new set of parameters is used in the next iteration of the line search.

As a final note, certain robustness features have been incorporated in to the line search procedure. Firstly, in rare cases, involving highly non-linear function behaviour, a negative step size may be computed, in this case the step size is increased to a value of 2 [14]. In addition, in certain situations, a negative square root may be encountered when computing the roots using Eq. (6.52), in such a case, a step size cannot be computed. Consequently, under this circumstance the algorithm switches to bisection, where the step size is continuously halved until a reduction in the cost function is produced.

6.4.4.2 Inexact Backtracking Armijo Type Line Search Procedure

The exact line search technique, based on cubic interpolation may require many additional functional evaluations at each line search iteration, that do not necessarily get one closer to the minimum [48]. This can result in large computational cost at each line search iteration, especially if highly refined particle distributions are employed, which are exactly what is required for convergence of the element free Galerkin method, that SESKA is based on. In an attempt to side step some of these problems, it is common to employ inexact line search techniques, that are usually backtracking, in order to reduce

the cost function along the search direction, $\delta\boldsymbol{\theta}^k$, in a less computationally taxing manner [24]. The most popular choice for such a method is the Armijo type line search. In this technique, an iteration counter (which also represents the minor iteration number), t, is set equal to zero in the first line search iteration. A step size is then computed as

$$\alpha = \alpha_0 \delta^t, \tag{6.54}$$

where, α_0 and δ ($\alpha_0, \delta \in (0,1)$) are line search constants, with typical values of 0.5 and 0.9, respectively [24]. Based on Eq. (6.54), it is clear that the step size is decreased geometrically, as t increases from zero until a user defined maximum number of line search iterations. This method is advantageous as it promotes super-linear convergence as opposed to previous methods, which involve halving the step size at each iteration, resulting in associated linear convergence [24]. New parameters for the line search iteration are then computed as

$$\boldsymbol{\theta}^{k+1} = \boldsymbol{\theta}^k + \alpha\delta\boldsymbol{\theta}^k. \tag{6.55}$$

The cost function, $J^2(\boldsymbol{\theta}^{k+1})$, is then evaluated at the new parameter set. The new parameters are accepted if the following Armijo-Goldstein condition is satisfied:

$$J^2(\boldsymbol{\theta}^{k+1}) \leq J^2(\boldsymbol{\theta}^k) + \alpha\sigma \left[\delta\boldsymbol{\theta}^k\right]^{\mathrm{T}} \boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k), \tag{6.56}$$

where σ is a positive scalar $\in (0,1)$, with a typical value of 10^{-4}. The conditions in Eq. (6.56), ensures a sufficient decrease in the cost function, for some step size $\alpha > 0$. If the Armijo-Goldstein conditions are satisfied, then the new parameter set, $\boldsymbol{\theta}^{k+1}$, is accepted, the line search is terminated, and the algorithm moves to the next iteration of the projected Levenberg-Marquardt algorithm. However, if the conditions are not met, the new parameter set is discarded, the iteration counter is incremented by one ($t := t + 1$) and the algorithm moves to the next iteration of the line search.

6.4.4.3 Protection against Infinite Loops and Small Step Sizes

As with any iterative technique, the line search needs to be protected against an infinite loop. This is done by setting a maximum number of line search iterations, $t_{\max}$, for both the cubic and inexact backtracking line searches. If this maximum number of iteration steps are exceeded, then the parameters are reset to the old values, $\boldsymbol{\theta}^k$, and a projected gradient search is initiated, in order to attempt to reduce the cost function value, in the direction opposite to the gradient of the objective function. Additionally, in certain circumstances, it may be found that the step size required to provide a sufficient reduction in the cost function is extremely small (less than 10^{-3} for example). This is

an undesirable outcome of the line search. In this case the parameters are reset to the old values, $\boldsymbol{\theta}^k$, and a projected gradient search is again initiated.

6.4.5 Projected Gradient Search

If the new parameters, $\boldsymbol{\theta}^{k+1}$, do not produce a reduction in the cost function, and the descent condition is violated, then a projected gradient search technique is utilized to find new parameters, that produce an acceptable reduction in the cost function. It should be noted that if the descent condition is violated, it automatically implies that the new parameter set has not reduced the cost function value, the converse is not true however, i.e., a new parameter set may not reduce the cost function, but the descent condition may still be satisfied, if this where not the case, then the line search techniques would not be readily applicable. The projected gradient search is initiated under the conditions given by

$$J^2(\boldsymbol{\theta}^{k+1}) \geq J^2(\boldsymbol{\theta}^k), \tag{6.57}$$

$$\left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \delta\boldsymbol{\theta}^k \geq 0, \tag{6.58}$$

where, $\delta\boldsymbol{\theta}^k$ is the potentially adjusted search direction computed using Eq. (6.41). Additionally, the projected gradient search may be initiated in the case where the line search has failed to produce new parameters, as the step size has fallen below a minimum threshold or the maximum defined number of line search iterations is exceeded. In such cases, new material parameters are computed as

$$\boldsymbol{\theta}^{k+1} = \mathrm{P}\left[\boldsymbol{\theta}^k - \alpha \boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right], \tag{6.59}$$

where, $\mathrm{P}\left[\boldsymbol{\theta}^k - \alpha \boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]$ represents the projection of the parameters obtained from the search (Eq. (6.39), onto the feasible parameter space, bounded by the hyperbox, $[\boldsymbol{\theta}^{\min}, \boldsymbol{\theta}^{\max}]$ (refer to Fig. 6.7). In addition, α is a step size computed via Eq. (6.54) (with t in Eq. (6.54) set to zero for the first projected gradient iteration). The cost function at the new parameter set, $J^2(\boldsymbol{\theta}^{k+1})$, is then computed. The new parameters are accepted if the following condition is satisfied [24]:

$$J^2(\boldsymbol{\theta}^{k+1}) \leq J^2(\boldsymbol{\theta}^k) + \sigma \left[\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right]^{\mathrm{T}} \left[\mathrm{P}\left[\boldsymbol{\theta}^k - \alpha \boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)\right] - \boldsymbol{\theta}^k\right], \tag{6.60}$$

where σ is a again positive scalar $\in (0, 1)$, with a typical value of 10^{-4}. When the condition in Eq. (6.60) is not satisfied, then the new parameters are rejected, t is incremented by one, and the next projected gradient iteration step is initiated. Conversely, if the condition is met, then the new parameters are accepted and used for the next iteration

of the projected Levenberg-Marquardt algorithm.

It is clear that the projected gradient search is simply a constrained form of the steepest descent algorithm (refer to Eq. (6.18) and Eq. (6.28)), as parameters are updated in the direction opposite to the gradient of the cost function. Consequently, for a sufficiently small step size, the projected gradient search is guaranteed to find an acceptable decrease in the cost function. Fig. 6.7 is a schematic illustration of the workings of a projected gradient search technique, for an objective function of two variables, θ_1 and θ_2, bounded by box constraints of the form, $\theta_i^{\max} \leq \theta_i \leq \theta_i^{\min}$, which form the feasible space, Λ. Fig. 6.7 also illustrates geometrically how parameters that violate box constraints, are projected back onto the feasible search space, Λ.

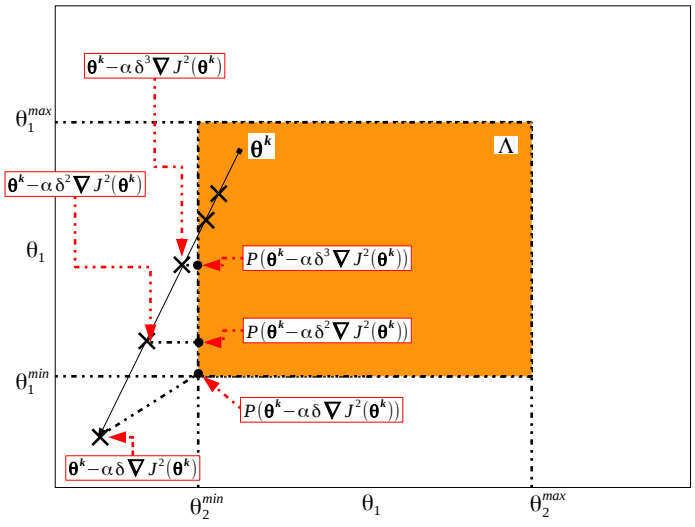

FIGURE 6.7: A schematic illustration of the projected gradient search technique.

As a final note, it needs to be emphasized that the projected gradient step is crucial to the success and robustness of the projected Levenberg-Marquardt algorithm, mainly because the projection of parameters onto the feasible space can cause loss of descent direction.

6.4.6 Updating the Damping Parameter μ

At the core of the search direction computation, $\delta\boldsymbol{\theta}_u$, is the algorithmic parameter, μ, commonly referred to as the damping parameter. Small values of the algorithmic parameter, μ, result in a approximate Gauss-Newton type update and large values result in an approximate gradient (steepest) descent type update [56]. The choice of μ therefore affects both the size and direction of the descent direction, $\delta\boldsymbol{\theta}_u$, computed using the

projected Levenberg-Marquardt algorithm. If the solution of the parameter vector, $\boldsymbol{\theta}$, is close to the optimal solution that minimizes the cost function, then smaller values of μ are desired, such that the algorithm benefits from the rapid final convergence exhibited by the Gauss-Newton method [23]. Conversely, if the parameter vector is far from the optimal solution, that minimizes the cost function, then larger values of μ are required, such that the algorithm benefits from the steady and robust convergence behaviour of the steepest descent method [23]. Kanzow et al. [24] suggested that the value of the cost function is utilized as a means of sizing the damping parameter. When a set of parameters is far from the optimal solution, this distance is reflected in the magnitude of the cost function, the converse is true for parameters close to the minimum. Consequently, at the start of each iteration step, k, of the projected Levenberg-Marquardt algorithm, the damping term is computed as

$$\mu_k = \mu_0 \left[J^2(\boldsymbol{\theta}^k) \right], \tag{6.61}$$

where, $\boldsymbol{\theta}^k$ represents the parameter vector at the start of a major iteration, k. In addition, μ_0 is a positive constant ($\mu_0 > 0$). From Eq. (6.61), when the parameters are close to the minimum, the cost function value will be small, resulting in a small damping term, which in-turn leads to an approximate Gauss Newton step. Conversely, when parameters are far from the optimal solution, the cost function will be large, resulting in a large damping term, which shall result in an approximate steepest descent step. This method of sizing the damping term implicitly causes the projected Levenberg-Marquardt algorithm, to switch between steepest descent and Gauss Newton type updates. This method of updating the damping term, is reported to have significantly more robust behaviour than the commonly employed delayed gratification technique [15, 23, 58], which involves increasing the damping term by a constant term, β, if a new parameter set does not bring about a reduction in the cost function, and decreasing the damping term by another constant, $\gamma < \beta$, if a reduction of the cost function is observed.

6.4.7 Numerical Approximation of the Jacobian Matrices

The Jacobian matrices, $\mathbf{J^f}$ and $\mathbf{J^g}$ (the later is computed only for bi-objective cost function minimization), represent the sensitivity of the simulated vectors, $\mathbf{f}(\boldsymbol{\theta})$ and $\mathbf{g}(\boldsymbol{\theta})$, to small perturbations in the material parameters. These Jacobian matrices are approximated numerically using forward differences or alternatively using central differences, the latter method ensuring better approximations, despite greater computation effort. The formula used for the forward difference approximation of the Jacobian matrix, $\mathbf{J^f}$,

is given by

$$\mathrm{J}^{\mathbf{f}}_{ij} = \frac{\mathrm{f}_i(\boldsymbol{\theta} + \delta\boldsymbol{\theta}_j) - \mathrm{f}_i(\boldsymbol{\theta})}{\|\delta\boldsymbol{\theta}_j\|}, \tag{6.62}$$

similarly, the forward difference approximation of $\mathbf{J^g}$ is given by

$$\mathrm{J}^{\mathbf{g}}_{ij} = \frac{\mathrm{g}_i(\boldsymbol{\theta} + \delta\boldsymbol{\theta}_j) - \mathrm{g}_i(\boldsymbol{\theta})}{\|\delta\boldsymbol{\theta}_j\|}. \tag{6.63}$$

Alternatively using central differences, the approximation for $\mathbf{J^f}$ is

$$\mathrm{J}^{\mathbf{f}}_{ij} = \frac{\mathrm{f}_i(\boldsymbol{\theta} + \frac{1}{2}\delta\boldsymbol{\theta}_j) - \mathrm{f}_i(\boldsymbol{\theta} - \frac{1}{2}\delta\boldsymbol{\theta}_j)}{\|\delta\boldsymbol{\theta}_j\|}, \tag{6.64}$$

and similarly, the approximation for $\mathbf{J^g}$ is given by

$$\mathrm{J}^{\mathbf{g}}_{ij} = \frac{\mathrm{g}_i(\boldsymbol{\theta} + \frac{1}{2}\delta\boldsymbol{\theta}_j) - \mathrm{g}_i(\boldsymbol{\theta} - \frac{1}{2}\delta\boldsymbol{\theta}_j)}{\|\delta\boldsymbol{\theta}_j\|}. \tag{6.65}$$

In Eq. (6.62), Eq. (6.63), Eq. (6.64) and Eq. (6.65), $\delta\boldsymbol{\theta}_j$ is a vector of parameter perturbations, whose j^{th} element is the only non zero term. The j^{th} element of $\delta\boldsymbol{\theta}_j$ is computed as

$$(\delta\theta_j)_j = \epsilon_2(|\min(\theta_i)|), \tag{6.66}$$

where $\min(\theta_i)$ refers to the smallest component of the parameter vector, $\boldsymbol{\theta}$. Additionally, ϵ_2 is a small positive scalar, of magnitude say 10^{-6}. For the case of forward difference computations of the Jacobian, the simulated vectors, with perturbed material parameters, $\mathbf{f}(\boldsymbol{\theta}^{\text{perturbed}})$ and $\mathbf{g}(\boldsymbol{\theta}^{\text{perturbed}})$ (or simply $\mathbf{f}(\boldsymbol{\theta}^{\text{perturbed}})$ if the cost function is of single objective form), need to be computed j times, where j is equal to the size of the parameter vector, $\boldsymbol{\theta}$. At the j^{th} iteration of the Jacobian calculation, only the j^{th} component of the parameter vector, $\boldsymbol{\theta}$, is perturbed according to

$$\theta_i^{\text{perturbed}} = \theta_i + (\delta\theta_j)_j \quad \text{for } i = j, \tag{6.67}$$

$$\theta_i^{\text{perturbed}} = \theta_i \quad \text{for } i \neq j, \tag{6.68}$$

where θ_i are the components of the unperturbed material parameter vector. When the Jacobian is computed using central differences the simulated vectors, with perturbed parameters now need to be computed n times, where $n = 2 \times j$. At the n^{th} iteration of the Jacobian calculation, if $n \leq j$, the j^{th} component of the perturbed parameter vector

is computed as

$$\theta_i^{\text{perturbed}} = \theta_i + \frac{1}{2}(\delta\theta_j)_j \quad \text{for } i = j, \tag{6.69}$$

$$\theta_i^{\text{perturbed}} = \theta_i \quad \text{for } i \neq j. \tag{6.70}$$

In addition, if $n > j$, the j^{th} component of the perturbed parameter vector is

$$\theta_i^{\text{perturbed}} = \theta_i - \frac{1}{2}(\delta\theta_j)_j \quad \text{for } i = j, \tag{6.71}$$

$$\theta_i^{\text{perturbed}} = \theta_i \quad \text{for } i \neq j. \tag{6.72}$$

Consequently, for large problems, with hundreds (or even thousands of parameters), or for moderate sized problems that employ a very refined particle distribution (the later being essential to achieve convergence of the element free Galerkin method), whether the Jacobian is computed via forward or central differences, the projected Levenberg-Marquardt algorithm can becomes a computationally expensive and time consuming means of identifying material parameters. Methods are available in the literature for combating this problem, and usually involve approximating the Jacobian, using a Broyden rank 1 update [58], at no significant computational cost. Broyden's rank 1 update is essentially a generalized secant procedure. Based on the author's experience with this method, it appears to work well for unconstrained minimization problems. However, in the case of constrained optimization, it is usually required that accurate cost function gradients are provided, especially if the constraint region is narrow. Consequently, for this work, as an initial starting point for the algorithm implementation, the Jacobian matrices are always evaluated using a finite difference method at each iteration. However, it is recommended that in future Broyden's rank 1 update is integrated into the algorithm, in order to speed up computational time. With regards to this implementation, care needs to be taken as to when the secant procedure is applicable.

Another draw back of the finite difference scheme of computing the Jacobian matrices, is that it limits the size of parameter perturbation, $(\delta\theta_j)_j$, that can be used. To elaborate, if extremely small parameter perturbations are used of the order say, 10^{-10} or smaller, then computationally, one may encounter numerical rounding errors, which may severely hamper the accuracy of the computed Jacobian, which may inevitably lead to a diverging calculation.

6.4.8 Algorithm Stopping Criteria

Four stopping criteria are recommended by Kanzow et al. [24], at which the minimization routine is terminated. These stop criteria are given by

$$J^2(\boldsymbol{\theta}^{k+1}) < \epsilon_1, \tag{6.73}$$

$$||\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)|| < \epsilon_2, \tag{6.74}$$

$$\frac{||\delta\boldsymbol{\theta}^k||}{||\boldsymbol{\theta}^k||} < \epsilon_3, \tag{6.75}$$

$$k < \mathrm{k}^{max}. \tag{6.76}$$

The projected Levenberg-Marquardt algorithm is terminated, if any of the above stop criteria are achieved. The first stop criterion refers to the value of the cost function at the new parameter set. The second stop criterion refers to the norm of the cost function gradient vector. The third stop criterion refers to the ratio of the norm of the parameter increment, to that of the original parameter vector. The final stop criterion, sets a maximum number of major algorithm iterations for the optimization. If the optimization is terminated by maximum number of major iterations being exceeded, it means that the obtained parameter set does not represent a stationary point that minimizes the cost function. Furthermore, ϵ_1, ϵ_2, and ϵ_3 are user defined thresholds for the first, second and third stop criterion's respectively.

6.4.9 Further Numerical and Conceptual Complexities

Now that the major aspects of the projected Levenberg-Marquardt algorithm are outlined, some further numerical and conceptual complexities, regarding non-linear least squares curve fitting applications will be explored.

6.4.9.1 The Number of Data Points (N)

Experimental data always contains a certain degree of noise due to measurement errors, which may affect convergence of any optimization algorithm employed [20, 59]. Hence, one should make use of repeated measurements of a particular experiment, to generate an averaged set of data, in order to reduce the effect of measurement noise and inaccuracy due to outliers in experimental data. In addition, if the experimental data relationship is approximately linear, then fewer data points at which simulated and experimental data are compared, can be used. However, if the experimental data relationship is non-linear, then more data comparison points should be concentrated in areas where the

curvature is observed to change rapidly, and less can be allocated in portions where the relationship is linear or less non-linear.

6.4.9.2 Local versus Global Convergence

Non-linear least squares problems can have objective cost functions that contain multiple minima across there domain, i.e., more than one set of parameters may exist that minimizes the cost function (see figure 6.8). Consequently, optimization algorithms such as those mentioned in the previous sections will converge to different local minima, depending on the initial parameter guess, the level of noise in the experimental data and the values of the algorithmic parameters such as μ_0 [56]. Hence, in the absence of physical insight into a particular curve fitting application, it is required that the parameter space is first searched manually, in order to find an estimate of parameters that are in the neighbourhood of an optimal solution, that minimizes the cost function. In addition, there appears to be no full proof manner of determining whether a determined local minima is in fact a global minima, except by using several different initial parameter guesses, and observing whether the optimization algorithm converges to this same minima. If several minima are identified within a parameter search space, then the best estimate of the global minimum in this space, is the the one that produces the lowest cost function value. Fig. 6.8, is an illustration of a uni-modal cost function (i.e. dependant on one parameter only) with three minima, two are local minima, and one is a global minima. Fig. 6.8 illustrates that depending on the starting point for the parameter, θ_1, a gradient based minimiser such as the projected Levenberg-Marquardt algorithm, can converge to different local minima.

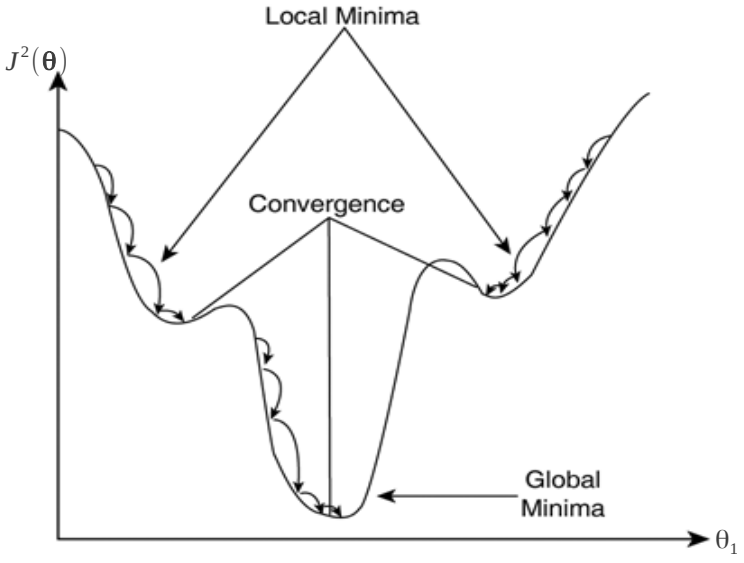

FIGURE 6.8: Schematic illustration of the existence of multiple minima.

6.4.9.3 The Boundary Parameter Problem

In previous sections it was illustrated that box constraints are utilized in order to constrain material parameters to a predefined parameter search space, i.e., Eq. (6.40). This method of constraining parameters has one shortcoming that is of particular interest. In order to illustrate this shortcoming consider a single objective cost function, $J^2(\boldsymbol{\theta})$, where the parameter vector, $\boldsymbol{\theta}$, has only one element, namely, θ_1. In addition, θ_1 is constrained by upper and lower bound values of $\theta_1^{\max}$ and $\theta_1^{\min}$, respectively. Fig. 6.9, illustrates an idealized cost function variation over a range of values for the parameter θ_1. In this figure, point A represents a desired global minimum in the cost function, and B represents a local minimum, the latter existing outside the parameter search space. Additionally, points C and D represent two different starting parameters, utilized in two separate optimization attempts. Also, point C' and D' represent the converged parameters from each optimization attempt, starting at C and D, respectively. The solid purple line represents the direction in which the parameter, θ_1, evolves during optimization, starting from point C and ending at point C'. Similarly, the solid orange line represents the direction in which the parameter, θ_1, evolves during optimization, starting from point D and ending at point D'.

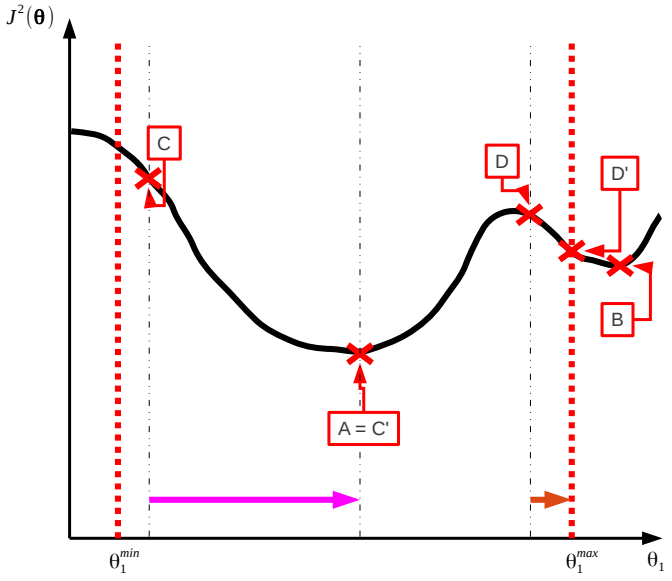

FIGURE 6.9: Schematic illustration of the boundary parameter problem associated with gradient based optimization algorithms such as the bounded Levenberg-Marquardt algorithm.

Referring to Fig 6.9, with the starting parameter, θ_1, at C, steady and convergent progress is achieved towards the global minimum in the cost function at A. However,

starting at an initial parameter of θ_1 at point D, parameters move in a downhill direction, towards the local minimum at point B. However, the upper boundary of the parameter search space, $\theta_1^{\max}$, prevents parameters from converging towards this local minimum, hence, parameters cease to converge further when they encounter this parameter boundary. Consequently, the achieved parameters that are found starting at point D are meaningless as they do not represent a minimum in the cost function. This problem is derived from the fact that the projected Levenberg-Marquardt algorithm is a gradient based optimizer, hence a good initial parameter guess is required such that parameters converge to at least a local minimum in the cost function, within a predefined parameter search space. The boundary value problem, also indicates that the parameter boundaries need to be chosen carefully, based on physical insight into a particular parameter optimization application, as well as insight into the material model that parameters are derived from. To illustrate the later point, consider the case of linear elasticity, with the material model governed by parameters E and ν, representing Young's modulus and Poisson's ratio. In typical finite element simulations, if ν is larger than 0.5, then the bulk modulus is infinite, hence, it would be prudent to restrict this parameter to a maximum value of 0.5 during optimization.

Chapter 7

Results and Discussion

7.1 Introduction

In chapter six, the projected Levenberg-Marquardt algorithm was introduced, as a means of material parameter identification, for inverse problems. This algorithm was first implemented by the author in the *Test* module, of the in-house modelling software SESKA. At this stage, the algorithm was tested on several non-linear least squares curve fitting applications [56], and was found to display robust convergence behaviour. Once the author was satisfied with the algorithm performance, it was coupled to the EFGM method implemented in SESKA. Following this implementation, the algorithm was tested on additional curve fitting applications.

In this chapter, one example shall be explored, which captures the behaviour of the projected Levenberg-Marquardt algorithm well, through a parametric study. The considered problem, is the identification of elastic material parameters for a cantilever beam problem. Next, the projected Levenberg-Marquardt algorithm, was used to calibrate material parameters for myocardial tissue, in order to achieve a closed loop pressure-volume curve, for a full beat of a cardiac cycle. The initial portion of this section, will present results of this model calibration. Furthermore, the latter sections, will include a mechanical analysis of the deformation of the left ventricle. These observations, will be utilized to highlight several key issues pertaining to the manner in which left ventricular cardiac function is modelled.

7.2 Illustrative Example

In the following sections, the projected Levenberg-Marquardt algorithm will be utilized to calibrate elastic material parameters, for a cantilever beam problem.

7.2.1 Problem Setup

Fig. 7.1(a) illustrates a cantilever beam, 10 m long, subjected to a downward surface traction, $\boldsymbol{\lambda}$, at the tip. In terms of displacement boundary conditions, the left hand side of the beam was clamped. The beam was modelled as linear elastic, using a $St.Venant-$

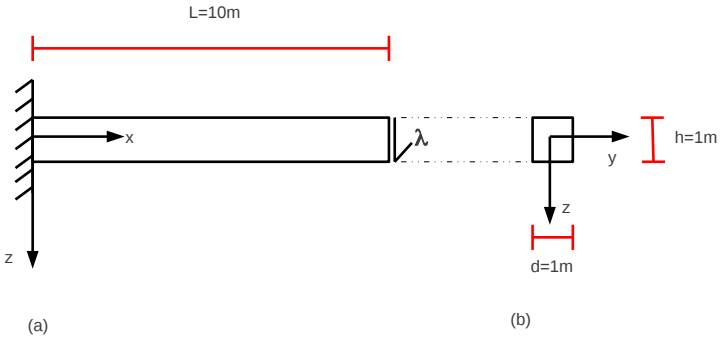

FIGURE 7.1: (a)Long section of the a cantilever beam of length L, with a surface traction, $\boldsymbol{\lambda}$, applied at the end. (b)Cross sectional view of the cantilever beam, with, d, denoting the width, and h, denoting the height.

$Kirchoff$ constitutive model (refer to Eq. (2.49)). The beam model was generated in the software GID, and discretized into 80 hexahedral elements with 189 nodes, the latter corresponding to the number of particles used in the EFGM method implemented in SESKA. The discretized beam model is illustrated in Fig. 7.2. In addition, a set of assumed parameters were taken as

$$\boldsymbol{\theta}^{\text{true}} = \begin{pmatrix} \text{E} \\ \nu \end{pmatrix} = \begin{pmatrix} 2100\,\text{N/m}^2 \\ 0.3 \end{pmatrix}. \tag{7.1}$$

Using the set of assumed parameters, $\boldsymbol{\theta}^{\text{true}}$, pseudo experimental data was generated, which included the vertical tip displacement, at various loads, as the latter increased from zero to 1 N/m^2, in increments of 0.25 N/m^2. The pseudo experimental data used for the inverse problem is illustrated in Fig. 7.3. For clarity, the pseudo experimental

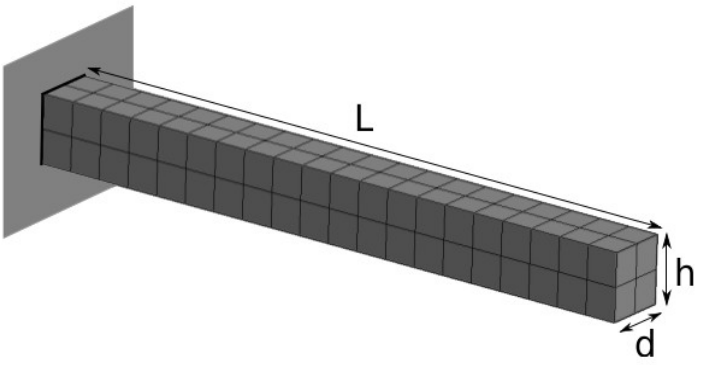

FIGURE 7.2: The cantilever beam model consisting of 189 particles.

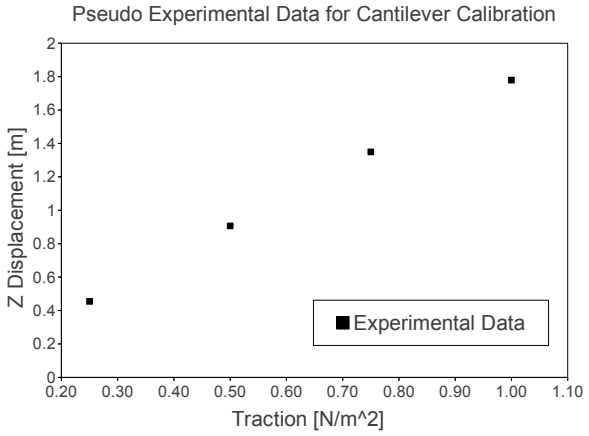

FIGURE 7.3: Pseudo experimental data used for the cantilever calibration.

data vector, $\hat{\mathbf{u}}$, has components given by

$$\hat{\mathbf{u}} = \begin{pmatrix} 0.455\,\mathrm{m} \\ 0.907\,\mathrm{m} \\ 1.350\,\mathrm{m} \\ 1.779\,\mathrm{m} \end{pmatrix}, \tag{7.2}$$

which are all derived from Fig. 7.3. Also, we define $\mathbf{u}(\boldsymbol{\theta})$ as a vector of simulated vertical tip displacements, whose components are

$$\mathbf{u}(\boldsymbol{\theta}) = \begin{pmatrix} u\left(\lambda = 0.25\,\mathrm{N/m^2}, \boldsymbol{\theta}\right) \\ u\left(\lambda = 0.5\,\mathrm{N/m^2}, \boldsymbol{\theta}\right) \\ u\left(\lambda = 0.75\,\mathrm{N/m^2}, \boldsymbol{\theta}\right) \\ u\left(\lambda = 1.0\,\mathrm{N/m^2}, \boldsymbol{\theta}\right) \end{pmatrix}, \tag{7.3}$$

where $\boldsymbol{\theta}$ refers to the current material parameter vector and λ refers to the magnitude of the applied traction.

Next, for the purposes of material parameter identification, a cost function, which will be minimized, is constructed as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\left[\left(\hat{\mathbf{u}} - \mathbf{u}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{u}}} \left(\hat{\mathbf{u}} - \mathbf{u}(\boldsymbol{\theta})\right)\right], \tag{7.4}$$

where the diagonal weight matrix, $\mathbf{W}^{\hat{\mathbf{u}}}$, has components given by

$$\mathrm{W}_{ij}^{\hat{\mathbf{V}}} = \begin{cases} \frac{1}{[\hat{u}_i]^2} \text{ for } i = j \\ 0 \text{ for } i \neq j. \end{cases} \tag{7.5}$$

This choice of the weight matrix, effectively normalizes the problem and transforms the cost function, into a sum over the number of data points, of the squared percentage error, between simulated and experimental vertical tip displacements. A similar choice for the weight matrix, was utilized by Dubuis et al. [20], in their attempts to calibrate soft tissues of the compressed leg. However, the effect of this weight matrix, is that cost function values may appear small, as compared to values that would be computed if the weight matrix is initialized to the identity matrix, $\mathbf{I}$ [23, 56]. Hence, a user may be mislead into believing that the cost function associated with a parameter set, is close to a minimum. To illustrate this point consider the exact solution to the tip displacement, for the cantilever beam problem, which is given by

$$u\left(F, \boldsymbol{\theta}\right) = \frac{FL^3}{3\mathrm{E}I}, \tag{7.6}$$

where I, refers to the moment of inertia and F is the resultant force, acting on the end surface, at the tip of the cantilever beam. If Young's modulus is assumed to be 1500 $\mathrm{N/m^2}$, and the resultant force, F is set to 1 N, then the theoretical tip displacement computed from Eq. (7.6) is 2.67 m. However, the true tip displacement, with Young's modulus equal to 2100 $\mathrm{N/m^2}$, is equal to 1.904 m. Assuming only one force-deflection pair is used for data comparison (i.e. at F=1 $\mathrm{N/m^2}$), then the cost function computed

using Eq. (7.4) is 0.081. Referring to this cost function value, it may be interpreted that the parameter for Young's modulus, is close to the true solution. However, if the cost function value is converted to an accumulated percentage error, over the number of data points, the resulting value is 40.2 percent. Hence, the accumulated percentage error, between the simulated and experimental data is around 40 percent, which implies that the cost function is large, and the initial parameter set is far from the optimal solution. Thus, when the weight matrix is defined by Eq. (7.5), care needs to be taken when interpreting whether an initial parameter set, is close to or far from the optimal solution.

The parameter search space was then constrained, by applying and upper and lower limit for parameters E and ν. The upper boundary of the parameter search space was established as

$$\boldsymbol{\theta}^{\max} = \begin{pmatrix} \mathrm{E}^{\max} \\ \nu^{\max} \end{pmatrix} = \begin{pmatrix} 3000\,\mathrm{N/m^2} \\ 0.4 \end{pmatrix}. \tag{7.7}$$

Also, the lower lower bound for the parameter search space was defined by

$$\boldsymbol{\theta}^{\min} = \begin{pmatrix} \mathrm{E}^{\min} \\ \nu^{\min} \end{pmatrix} = \begin{pmatrix} 1850\,\mathrm{N/m^2} \\ 0.1 \end{pmatrix}. \tag{7.8}$$

Next, the ability of the projected Levenberg-Marquardt algorithm to minimize Eq. (7.4) and arrive at a solution for the parameter vector, $\boldsymbol{\theta}$, that is close to the true parameter set, $\boldsymbol{\theta}^{\mathrm{true}}$, was tested. Several starting parameters, which are summarized in Tab. 7.1, were used to test for global convergence.

Parameter	Run 1	Run 2	Run 3
$\mathrm{E}\,(\mathrm{N/m^2})$	1850	3000	2000
ν	0.1	0.4	0.2

TABLE 7.1: Various initial parameter guesses used to test for global convergence of the projected Levenberg-Marquardt algorithm

Referring to Tab. 7.1, run 1 and run 2, represent minimization attempts, where initial parameter guesses correspond to the lower and upper boundary of the parameter search space, respectively. Run 3, corresponds to an initial parameter guess that is relatively close to the true parameter set.

The line search procedure utilized was the cubic interpolation method, presented in Sec. 6.4.4.1. The maximum number of line search or projected gradient iterations (i.e. maximum number of minor iterations), $t^{\max}$, was set to 10. The algorithmic constant,

μ_0 (Eq. (6.61)), utilized for the calibration was set to a value of unity. The stopping criteria for the algorithm were set to stringent values in order to test convergence. The stopping criteria for the algorithm are listed below.

- $J^2(\boldsymbol{\theta}^{k+1}) < 10^{-8}$.
- $||\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)|| < 10^{-8}$.
- $\frac{||\delta\boldsymbol{\theta}^k||}{||\boldsymbol{\theta}^k||} < 10^{-8}$.
- $k < 50$,

Note that, k, $J^2(\boldsymbol{\theta}^{k+1})$ and $||\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)||$, refer to the major iteration number, the cost function at the new parameter set and the norm of the cost function gradient vector respectively. $||\delta\boldsymbol{\theta}^k||$ and $||\boldsymbol{\theta}^k||$, refer to the norm of the parameter increment and that of the original parameter vector respectively.

7.2.2 Convergence Analysis

In Tab. 7.1, various initial parameter guesses for separate optimization attempts were presented. In the following sections, the convergence behaviour of the projected Leveneberg-Marquardt algorithm will be analysed, for each starting parameter guess in Tab. 7.1. In addition, an attempt will be made to explain the behaviour of the algorithm, based on the various initial parameters.

7.2.2.1 Results for Run 1

Run 1, uses initial starting parameters of 1850 N/m^2 and 0.1 for E and ν, respectively. The initial value of the cost function, computed via Eq. (7.4), is 0.035, which translates to an initial accumulated percentage difference between simulated and experimental data, of around 26 percent.

Figs. 7.4 (a) and (b), illustrate the evolution of Young's modulus and Poisson's ratio, respectively, over the duration of the optimization. Note, that iteration $k = 0$, refers to the end of the first major iteration of the projected Levenberg-Marquardt algorithm. Referring to Figs. 7.4 (a) and (b), it is clear that E and ν converge precisely, to the true parameter set given by Eq. (7.1). Also, Referring to Fig. 7.4 (b), at the end of the first iteration ($k = 0$), the new parameter computed for Poisson's ratio is 0.4, indicating that the solution has been projected onto the upper boundary for ν, via Eq. (6.40). Conversely, during the optimization, Young's modulus does not violate either its

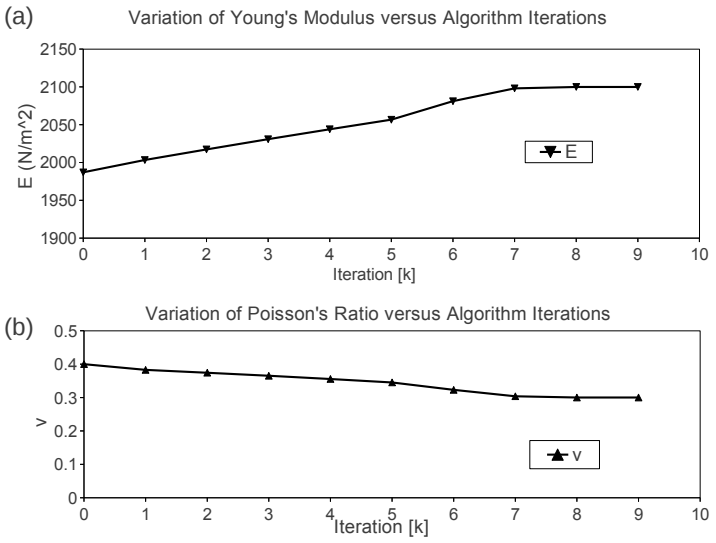

FIGURE 7.4: (a) The evolution of Young's modulus for Run 1. (b) The evolution of Poisson's ratio for Run 1.

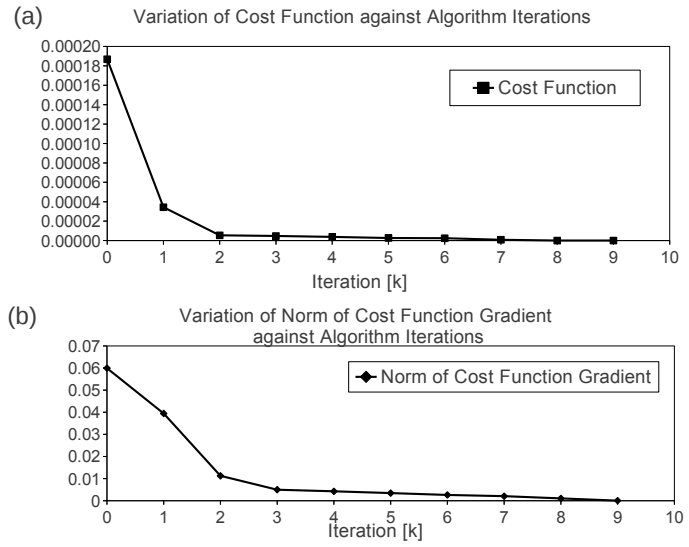

FIGURE 7.5: (a) The evolution of the cost function value for Run 1. (b) The evolution of the value of the norm of the cost function gradient vector for Run 1.

upper or lower boundary, hence the values for this parameter in Fig. 7.4 (a), represent the exact solution to the unconstrained minimization problem in Eq. (6.36).

The obtained final solutions for E and ν can be checked, by observing the convergence of the cost function, as well as that of the norm of the cost function gradient vector, over the duration of the optimization (Figs. 7.5 (a) and (b), respectively). Referring to Figs. 7.5 (a) and (b), it is clear that at the final parameter set obtained in the final algorithm iteration, the cost function and norm of the cost function gradient vector, are less that the desired thresholds of 10^{-8}. Consequently, the solution obtained for the

parameters, does in fact minimize the cost function, which is to say that they represent a local minimum in the cost function surface, which in this case, happens to coincide with the global minimum. Finally, the data fit produced by the converged parameter set, is illustrated in Fig. 7.6.

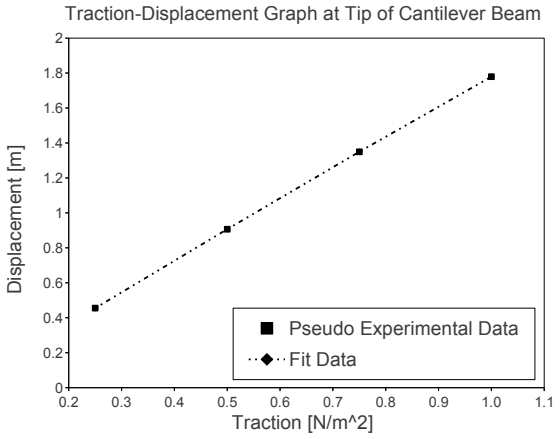

FIGURE 7.6: Data fit for the converged parameter set for Run 1.

7.2.2.2 Results for Run 2

Run 2, uses initial starting parameters of 3000 N/m^2 and 0.4 for E and ν, respectively. The initial value of the cost function, computed via Eq. (7.4), is 0.185, which translates to an initial accumulated percentage difference, between simulated and experimental data, of around 61 percent. Hence, this initial parameter guess, represents a starting point that is relatively far from the optimal solution given by $\boldsymbol{\theta}^{\text{true}}$.

Figs. 7.7 (a) and (b), illustrate the evolution of Young's modulus and Poisson's ratio, respectively, over the duration of the optimization. Referring to Figs. 7.7 (a) and (b), it is evident that E and ν, again converge precisely to their desired values, given by $\boldsymbol{\theta}^{\text{true}}$. Furthermore, Fig. 7.7 (a) illustrates that over the coarse of the optimization, E does not violate any on its constraint boundaries. However, based on Fig. 7.7 (b), ν is held at a constant value of 0.4 for iterations 1 to 3, indicating, that this parameter wants to violate its upper boundary, which is prevented by the Euclidean projection, utilized to handle boundary constraints.

The obtained final solutions for E and ν can be checked, by observing the convergence of the cost function, as well as that of the norm of the cost function gradient vector, over the duration of the optimization (Figs. 7.8 (a) and (b) respectively). Based on the final values of the cost function and norm of the cost function gradient vector, in

Figs. 7.8 (a) and (b) respectively, it is clear that the obtained solution for E and ν, is a local minimum, that minimizes the cost function. In addition, the number of algorithm iterations, required to minimize the cost function has not increased substantially from Run 1. This observation is attributed to the damping term, μ^k, that is present in Eq. (6.36), which is utilized to compute the solution to the unconstrained minimization problem, $\delta\boldsymbol{\theta}_u^k$. The damping term is computed using Eq. (6.61). Implicit to this damping update, is the value of the cost function at the initial parameters for an iteration, $J^2(\boldsymbol{\theta}^k)$, which in itself, serves as a gauge, as to how far a parameter set is from a desired minimum (characterized by $J^2(\boldsymbol{\theta}^k) \approx 0$). Run 2 starts at a parameter set that is further from the desired local minima, than that used in Run 1. The latter comment is clearly illustrated by the initial accumulated percentage differences, between simulated and experimental data, at the initial parameter sets, for Run 1 and Run 2 (26 percent and 61 percent respectively). Hence, the damping term at the start of the optimization for Run 2, is larger than that for Run 1, resulting in the algorithm taking smaller steps in an approximate steepest descent direction than the latter. This is necessary, because the Gauss-Newton method lacks the robustness of the steepest descent method, when parameters are far from the optimal solution. As the parameters in Run 2 approach the optimal solution, the reduced distance to the latter is reflected by a reduction in the cost function (see Fig. 7.8 (a)). As the cost function decreases, so too does the damping term, which causes the algorithm to take larger Gauss-Newton type steps, in the vicinity of the local minima, which allows the projected Levenberg-Marquardt algorithm, to benefit from final quadratic or super linear convergence. Finally, the data fit, produced by the converged parameter set, is illustrated in Fig. 7.9.

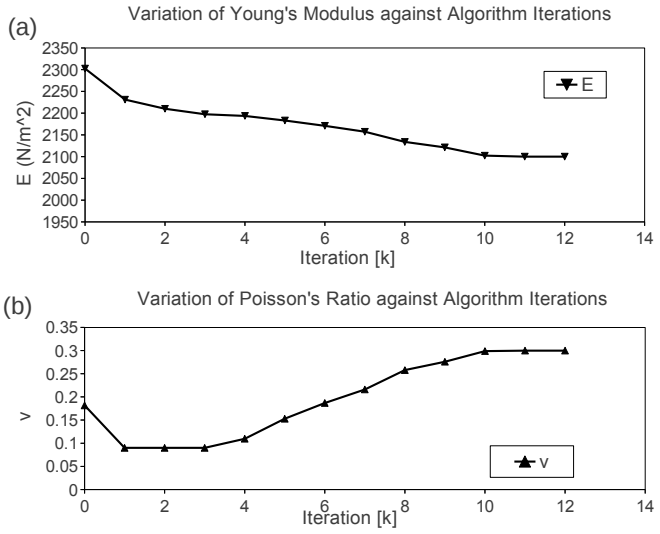

FIGURE 7.7: (a) The evolution of Young's modulus for Run 2. (b) The evolution of Poisson's ratio for Run 2.

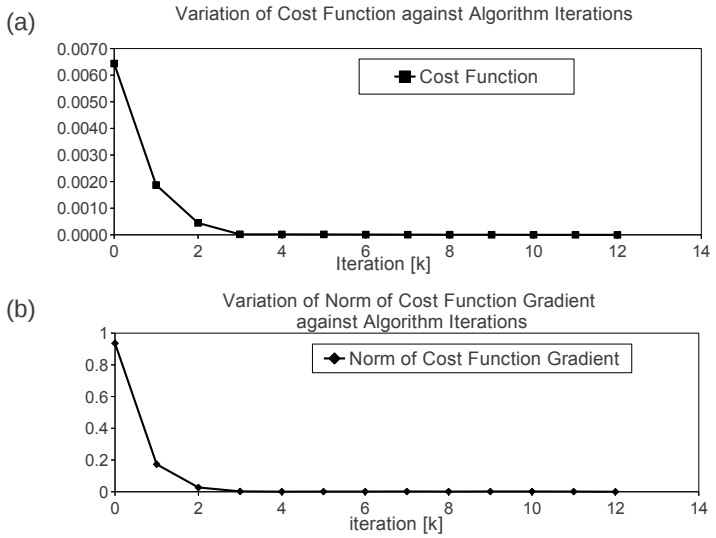

FIGURE 7.8: (a) The evolution of the cost function value for Run 2. (b) The evolution of the value of the norm of the cost function gradient vector for Run 2.

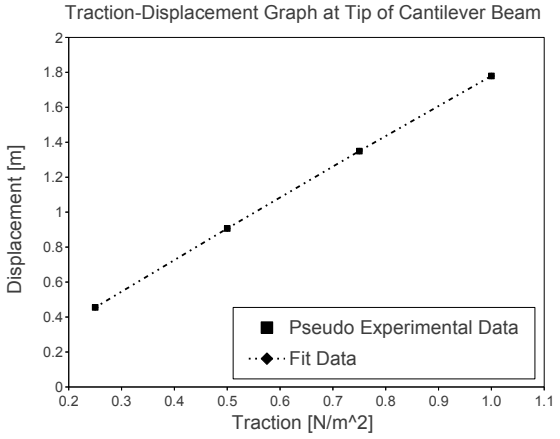

FIGURE 7.9: Data fit for the converged parameter set for Run 2.

7.2.2.3 Results for Run 3

Run 3, uses initial starting parameters of 2000 N/m^2 and 0.2, for E and ν respectively. The initial value of the cost function, computed via Eq. (7.4), is 0.0052, which translates to an initial accumulated percentage difference, between simulated and experimental data, of around 10.2 percent. Hence, this initial parameter guess, represents a starting point that is relatively close to the optimal solution, given by $\boldsymbol{\theta}^{\text{true}}$, as compared to Run 1 and Run 2.

Figs. 7.10 (a) and (b), illustrates the evolution of Young's modulus and Poisson's ratio, respectively, over the duration of the optimization. Referring to Figs. 7.10 (a) and

(b), it is clear that during the optimization, neither E nor ν, violates their respective parameter boundaries. Hence, the optimization for these parameters, is identical to an unconstrained minimization, which is to say that the computed search direction, $\delta\boldsymbol{\theta}^k$ (Eq. (6.41)), is always equal to $\delta\boldsymbol{\theta}_u^k$, at each algorithm iteration. The converged set of parameters at the end of the optimization are again precisely those given by $\boldsymbol{\theta}^{\text{true}}$.

As was the case for Run 1 and 2, the obtained final solutions for E and ν can be checked, by observing the convergence of the cost function value, as well as that of the norm of the cost function gradient vector, over the duration of the optimization (Figs. 7.11 (a) and (b) respectively). Referring to Figs. 7.11 (a) and (b), it is evident that the number of iterations, required to minimize the cost function, is less than what was required for Run 1 and 2. This observation is attributed to the fact that the initial parameter guess, is much closer to the optimal solution, than Run 1 and Run 2, which is illustrated by the accumulated percentage difference, between simulated and experimental data (10.2 percent for Run 3, as compared to 26 percent and 61 percent, for Run 1 and Run 2 respectively). Consequently, the damping term, μ^k, at the start of the optimization for Run 3 is relatively small, and the computed search direction resembles a Guass-Newton type step. As the optimization proceeds, the cost function is reduced and so too is the damping term. Consequently, the damping update method in Eq. (6.61), implicitly accounts for the approximate distance to the optimal solution, which in this case, enables the algorithm to benefit from the rapid final convergence, displayed by the Gauss-Newton method. Finally, the data fit, produced by the converged parameter set, is illustrated in Fig. 7.12.

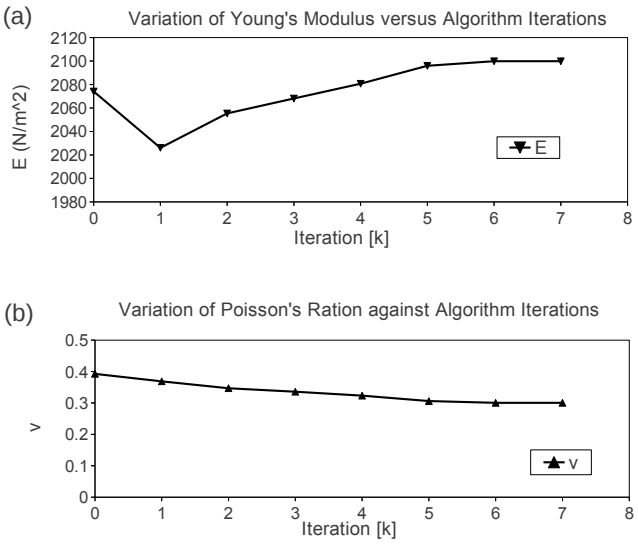

FIGURE 7.10: (a) The evolution of Young's modulus for Run 3. (b) The evolution of Poisson's ratio for Run 3.

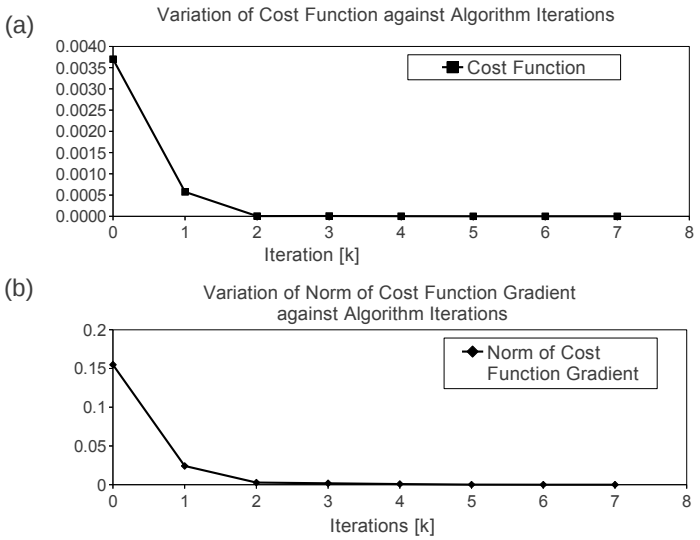

FIGURE 7.11: (a) The evolution of the cost function value for Run 3. (b) The evolution of the value of the norm of the cost function gradient vector for Run 3.

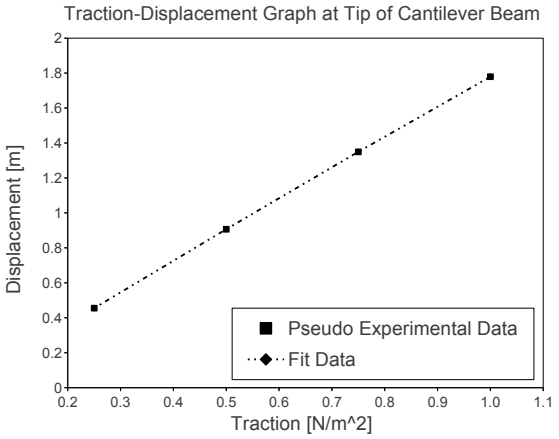

FIGURE 7.12: Data fit for the converged parameter set for Run 3.

7.2.2.4 Summary

Based on the convergence analysis of Run 1 to 3, the projected Levenberg-Marquardt algorithm, is capable of finding a local minimum of a cost function, starting from different initial parameter sets. In this example, the achieved local minimum, is in fact a global minimum, as the true parameter set is known prior to the optimization. However, the convergence behaviour illustrated in this example, does not imply that the algorithm is a global minimizer, that is independent of the initial parameter guess. In fact, for problems containing multiple minima, the algorithm will converge to different parameter sets, depending on the initial parameter guess. Hence, it is noted that for convergence, a good initial parameter guess is required.

Computational times for Run 1 to Run 3 were approximately, 1, 1.5 and 0.35 hours respectively, using 4 processors. Hence, in this example, computational times for all calibration attempts were reasonably small. However, for larger models that have many more degrees of freedom, then, in order to avoid large computational times associated with calibration, it is recommended that a starting point for optimization that is within ten percent of the optimal solution is chosen.

7.3 Left Ventricle Cardiac Mechanics Calibration

In Chap. 4, the idealized canine ellipsoidal model of the left ventricle was outlined. In this section, the results of the model calibration will be presented. The model was calibrated using the projected Levenberg-Marquardt algorithm. However, before these results are presented, the problem set-up for the cardiac mechanics problem will be defined.

7.3.1 Problem Setup

In the following sections, the set-up for the problem will be outlined, by considering the following aspects:

- Selection of an appropriate mesh size.
- Assignment of boundary Conditions for the model.

7.3.1.1 Selection of an Appropriate Mesh Size

Fig. 7.13 illustrates the canine ellipsoidal model, utilized for the purposes of calibration. The model was generated in the software GID and discretized into 224 hexahedral elements with 363 nodes, the latter corresponding to the number of particles used in the EFGM method implemented in SESKA. The relative coarseness of the particle distribution was chosen to minimize the computational time associated with calibration. To elaborate on the latter point, the average computational time associated with the systolic phase (isovolumetric contraction and ejection), was 8 hours and that for diastolic filling was approximately 2 hours, for the model consisting of 363 particles. Computational time for the model associated with the isovolumetric relaxation phase, was approximately 6 hours. Furthermore, in Chap. 6, it has been explained that the at each major iteration of the projected Levenberg-Marquardt algorithm, a Jacobian matrix is computed, representing the sensitivity of the simulated vectors, to changes in the material

parameters. Consequently, at each major iteration, if the Jacobian is approximated using forward differences (which is the default option), a simulation needs to be performed $m + 1$ times, where m refers to the number of material parameters that need to be identified. The result, was that for the calibration of the systolic phase, using the model consisting of 363 particles, each algorithm iteration required on average, $8(m+1)$ hours. Thus, the Jacobian evaluation at each major iteration, resulted in model calibration becoming unacceptable in terms of computational time, when using left ventricle models with highly refined particle distributions (i.e. models with more than 363 particles). The author acknowledges that for convergence of the EFGM method, refined particle distributions are required. However, for the purposes of model calibration, the model consisting of 363 particles represented the best compromise, between accuracy of the EFGM method and computational time.

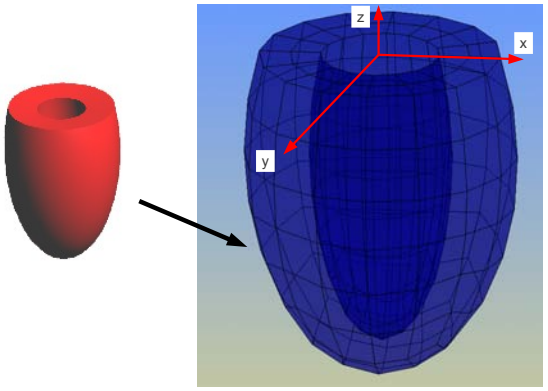

FIGURE 7.13: The ellipsoidal canine left ventricle model, discretized into 256 hexahedral elements with 363 nodes, the latter corresponding to the number of particles used in the EFGM method.

7.3.1.2 Boundary Conditions for the Model

In order to correctly simulate the mechanical response of the left ventricle, appropriate boundary conditions must be applied to the model. Two types of boundary conditions are required, namely, pressure and deformation boundary conditions [12, 17]. Deformation boundary conditions are required in order to prevent rigid body motion and also to constrain the base of the heart. In addition, pressure boundary conditions are required to simulate the blood filling pressure, acting on the endocardium surface of the left ventricle.

Deformation Boundary Conditions

The location of the base of the ellipsoidal model (refer to Fig. 4.1) corresponds to the

connection of the left ventricle, to the mitral aortic valves. This juncture, is relatively stiff, hence, the ring at the base of the ellipsoid model (i.e. the yellow ring in Fig. 7.14), is constrained to undergo no displacement in the x, y and z directions [17]. Additionally, all displacements in the z direction, are restrained on the base surface (the green surface in the Fig. 7.14).

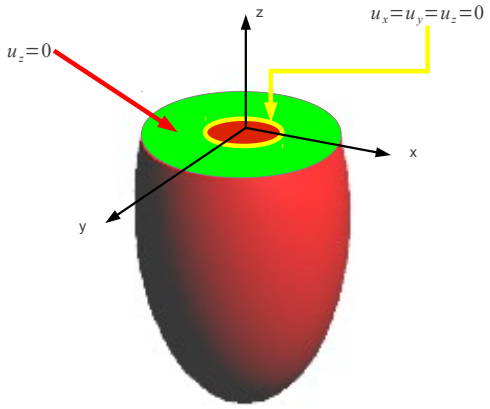

FIGURE 7.14: Deformation boundary conditions for the ellipsoidal left ventricle model

Pressure Boundary Conditions

The blood inside the heart exerts an outward pressure on the endocardium of the left ventricle, which suggests that a pressure boundary condition be applied to the inner wall of the ellipsoidal model. This pressure, is applied as an outward surface pressure, away from the cavity volume as illustrated in Fig. 7.15. Pressure boundary conditions, at the end of each of the four phases of the first beat of the cardiac cycle, for a canine left ventricle, have been found in the literature. These boundary conditions were used by Kerckhoffs et al. [17], whose left ventricle model specifications, are similar to those used in this work. The pressure boundary conditions are summarized in Tab. 7.2.

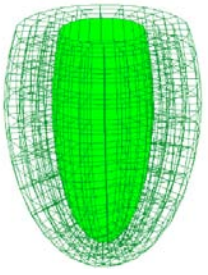

FIGURE 7.15: Illustration of the pressure boundary condition applied to the endocardium surface (green surface)

TABLE 7.2: Pressure boundary conditions for a canine ellipsoidal left ventricle model [17]

Phase of the cardiac cycle	final cavity pressure(kPa)
Diastole	1.0
Isovolumetric contraction	10.0
Ejection	13.3
Isovolumetric relaxation	0.0

7.3.2 Methodology for the Inverse Problem

In the context of this research the focus of inverse modelling, is geared primarily towards calibration of material parameter for myocardial tissue (passive and active tension parameters), as well as Windkessel parameters relating to the arterial system. Fig. 7.16 is an illustration of a set of idealized simulated (solid black lines) and experimentally observed (green lines) cardiac cycles.

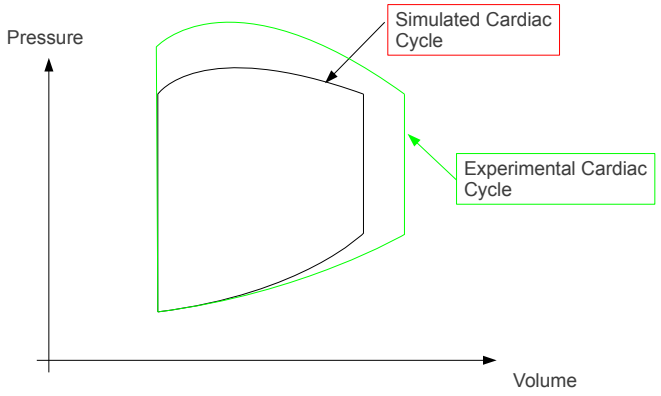

FIGURE 7.16: Illustration of a set of idealized simulated (solid black lines) and experimentally observed (green lines) cardiac cycles.

In this work, an inverse approach is used to calibrate material parameters, such that agreement is achieved between experimentally observed and simulated pressure-volume loops, for a single beat of a cardiac cycle. In the following sections, several aspects of the inverse problem will be elaborated on, which are summarized below:

- Methodology for passive material parameter identification.
- Methodology for active tension and Windkessel parameter identification.

7.3.2.1 Methodology for Passive Material Parameter Identification

Passive material parameters [28] of the myocardium, are identified during diastolic filling. During this phase of the cardiac cycle, the cavity pressure is increased incrementally upto an experimentally measured end-diastolic pressure (usually around 1 kPa [17]), making the simulation pressure driven. Due to the lack of detailed experimental displacement data (e.g. strain or displacement data from MRI images), for the left ventricle over the duration of diastolic filling, calibration was attempted using pressure-volume data.

Fig. 7.17 illustrates an assumed set of experimental and simulated pressure-volume data for the diastolic filling phase of the cardiac cycle. The dashed line in Fig. 7.17 illustrates a typical set of experimental data point and the solid line represents a typical set of simulated data point. From Fig. 7.17, it is important to notice that experimental and simulated cavity volumes, are compared at the same pressure point and it is for this reason, that the cost function used for passive material parameter identification, is of single objective form.

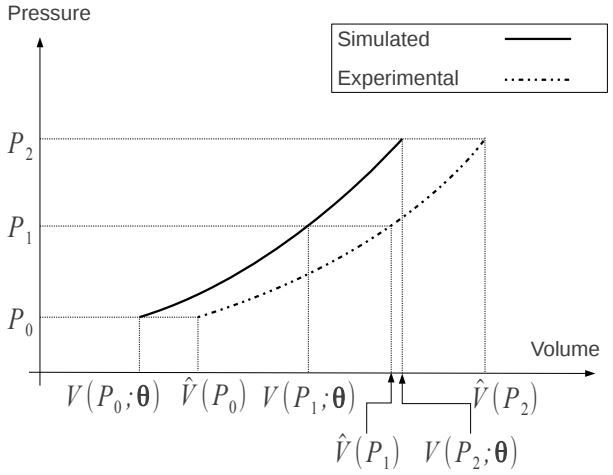

FIGURE 7.17: An idealized representation of experimental and simulated data for the diastolic filling phase of the cardiac cycle.

The cost function to be minimized can be formulated in terms of simulated and experimentally measured left ventricular cavity volumes. In this case, the first simulated vector is

$$\mathbf{f}(\boldsymbol{\theta}) = \mathbf{V}(\boldsymbol{\theta}), \tag{7.9}$$

and, the experimental vector is given by

$$\hat{\mathbf{f}} = \hat{\mathbf{V}}, \tag{7.10}$$

where, $\mathbf{V}(\boldsymbol{\theta})$ and $\hat{\mathbf{V}}$ are vectors containing simulated and experimental left ventricle cavity volumes ($V_i(P_i;\boldsymbol{\theta})$ and $\hat{V}_i(P_i)$, respectively), extracted at pressure steps, P_i and given a certain set of material parameters $\boldsymbol{\theta}$ (refer to Fig. 7.17). In addition, the parameter vector, $\boldsymbol{\theta}$, contains passive material parameters, which are obtained from the passive constitutive model of Usyk et al. [28], outlined in Chap. 3. The parameter vector has components given by

$$\boldsymbol{\theta}^{\mathrm{T}} = [A, A_{\mathrm{comp}}, a_1, a_2, a_3, a_4, a_5, a_6,]\,. \tag{7.11}$$

Initial calibration attempts involved identification of all the parameters given in Eq. (7.11). However, it was found that the projected Levenberg-Marquardt algorithm, would often converge to values for the parameters a_1 to a_6, that created numerical instability. Based on this observation, it was realized that the computed cavity volume used in calibration, is a direction-less quantity, that does not truly capture the anisotropic behaviour of the myocardium of the left ventricle, in the fibre, sheet and sheet-normal directions. Hence, parameters a_1 to a_6 were neglected from the optimization, which is reasonable since the variation of relative anisotropy, between specimens is nearly identical [26, 28]. Consequently, for the final calibration results, the parameter vector is reformulated as

$$\boldsymbol{\theta}^{\mathrm{T}} = [A, A_{\mathrm{comp}}]\,. \tag{7.12}$$

Next, the general single objective cost function in Eq. (6.2), is reformulated as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\left[\left(\hat{\mathbf{V}} - \mathbf{V}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{V}}} \left(\hat{\mathbf{V}} - \mathbf{V}(\boldsymbol{\theta})\right)\right]. \tag{7.13}$$

Alternatively, in discretized form, Eq. (7.13) is given by

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i=0}^{i=N-1}\left[\left(\frac{\hat{V}_i(P_i) - V_i(P_i;\boldsymbol{\theta})}{\mathrm{w}_i^{\hat{\mathbf{V}}}}\right)^2\right]. \tag{7.14}$$

Furthermore, the diagonal weight matrix, $\mathbf{W}^{\hat{\mathbf{V}}}$, in Eq. (7.13), has components given by

$$\mathrm{W}_{ij}^{\hat{\mathbf{V}}} = \begin{cases} \frac{1}{\left[\hat{V}_i(P_i)\right]^2} = \frac{1}{\left[\mathrm{w}_i^{\hat{\mathbf{V}}}\right]^2} \text{ for } i = j \\ 0 \text{ for } i \neq j. \end{cases} \tag{7.15}$$

7.3.2.2 Methodology for Active and Windkessel Parameter Identification

Active tension parameters were obtained from the constitutive model of Guccione et al. [29]. In addition, systemic parameters, relating to the arterial tree, are based on the three element Windkessel model. The parameter vector, $\boldsymbol{\theta}$, used for calibration, has components given by

$$\boldsymbol{\theta}^{\mathrm{T}} = [T_{\max}, B, m, C, R, R_0]\,. \tag{7.16}$$

Referring to Eq. (7.16), it is noted that that m, B and $T_{\max}$, are the only active tension parameters included in the vector, $\boldsymbol{\theta}$. The remaining active tension parameters, have been found from rigorous experimental testing on canine left ventricles [29] and are thus neglected from the material parameter identification. This assumption is validated, as the included active tension parameters, govern the gradient and peak of active tension development most pronouncedly [29].

Initial attempts at active tension and Windkessel parameter identification, involved the comparison of experimental and simulated cavity volumes and pressures, at specific time points, over the phases including systole and isovolumetric relaxation. Fig. 7.18 illustrates the typical set-up, with solid lines representing the simulated pressure-time and volume-time relationships, for systole and isovolumetric relaxation. In addition, the dotted lines on Fig. 7.18, represent the experimental pressure-time and volume-time relationships. What is not evident from Fig. 7.18, is that the duration of the phases

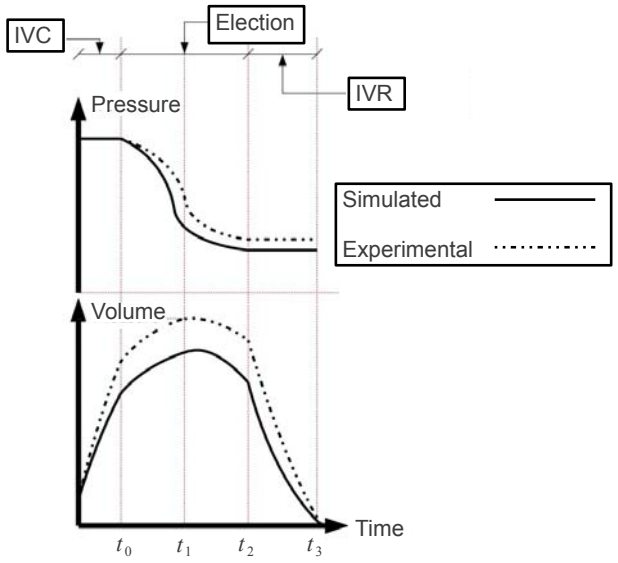

FIGURE 7.18: An idealized representation of experimental and simulated data for systole and isovolumetric relaxation.

consisting of systole and isovolumetric relaxation, are not necessarily equal for simulated and experimental data. To elaborate, changes in material parameters, result in changes in the simulated time duration required for completion of these phases. The simulated time duration, can be greater or less than the observed experimental duration. Hence, over the duration of a calibration attempt, the number of data comparison points, N, changes continuously. Thus, for initial calibration attempts, segmentation violations were encountered frequently, the latter being defined as attempts to access memory that the CPUs cannot address. These segmentation violations occurred because of the mismatch in size, between experimental and simulated vectors. A solution to this problem was found by comparing data over a minimum number of available data points. Hence, if the size of the simulated pressure and volume vectors, were less than their associated experimental vectors, then the number of data comparison points, N, was set equal to the size of the former. Conversely, if the size of the simulated pressure and volume vectors, was greater than their respective experimental vectors, then the number of data points was set equal to the size of the latter. This solution, prevented segmentation violations, which in turn allowed the projected Levenberg-Marquardt algorithm, to proceed through a calibration cycle. However, based on calibration attempts, the algorithm failed to produce meaningful parameters, with the termination criteria that stops the calibration, always being the maximum number of major iterations exceeded (i.e. $k > k_{max}$). Eventually, the reason behind this problem was found, by analysing the scale of the material parameters that were being identified. Tab. 7.3, illustrates a typical set of active tension and Windkessel parameters, obtained respectively from the works of Guccione et al. [29] and Kerckhoffs et al. [17].

TABLE 7.3: Typical parameter values, used for active and systemic parameter identification

Parameter	Typical Value
C $(\frac{\text{mm}^3}{\text{kPa}})$	1000
R $(\frac{\text{kPa·s}}{\text{mm}^3})$	1.4×10^{-5}
R_0 $(\frac{\text{kPa·s}}{\text{mm}^3})$	1.5×10^{-6}
B (mm)	4.75×10^{3}
T_{max} (kPa)	135.7
m $(\frac{\text{s}}{\text{mm}})$	1.0489×10^{3}

Referring to Tab. 7.3, it is noticed that the peripheral and flow resistances, R and R_0, are several orders of magnitude smaller than the arterial compliance and peak active tension, C and T_{max}. In addition, the peripheral and flow resistances, were found to be the most sensitive parameters, that affected the gradient of the pressure-time and

volume-time curves in ejection. Hence, these parameters play a significant role in determining the length of the simulated duration, for systole and isovolumetric relaxation. Referring to Sec. 6.4.7 of Chap. 6, the Jacobian matrices represent the sensitivity of the simulated vectors (e.g. simulated pressure and volume vectors), to perturbations in the material parameters. The material parameter perturbation used for the computation of the Jacobian matrix is given by Eq. (6.66). Since, the peripheral and flow resistances are small (of the order 10^{-5} and 10^{-6} respectively) and also represent the most sensitive parameters, the constant, ϵ_2, in Eq. (6.66), needed to be of the order of at least 10^{-4}, in order for the perturbed simulated pressure and volume vectors to not vary in size. Consequently, the parameter perturbation in the Jacobian calculation, was usually of the order of 10^{-10}. The small perturbation was desirable, because it resulted in the columns of the Jacobian matrix all being of the same size, which implies that all the simulations performed with perturbed parameters, for the Jacobian computation, had the same simulated time duration. However, the disadvantage of the small parameter perturbation, was that significant numerical rounding errors were encountered, in the forward or central difference computations of the elements of the Jacobian matrix. These rounding errors tended to accumulate, eventually leading the projected Levenberg-Marquardt algorithm on a diverging path. In light of these remarks, an alternate method of calibrating active tension and Windkessel parameters was developed. This method will be explored in the following sections.

Alternate Approach to Active Tension and Windkessel Parameter Identification

The systolic phase terminates, when the flow rate of blood from the contracting ventricle drops to zero ($I_{lv} = \frac{dV}{dt} \approx 0$), resulting in the aortic valves closing, thus signalling the beginning of isovolumetric relaxation [3, 17]. In addition, another key point during systole, is the pressure-volume relation corresponding to peak systolic pressure, which is achieved when the pressure-volume gradient is zero ($\frac{dP}{dV} \approx 0$). By default, a third point exists, which corresponds to the pressure-volume relationship at the end of isovolumetric contraction (end-IVC). However, this point is fixed, because the simulated pressure volume relationship, will always match the experimental. Nevertheless, material parameter identification involves minimizing a cost function that compares the experimental systolic pressure-volume relationships at these three points, to that extracted from a numerical simulation using SESKA. It should be noted, that during systole, both the cavity pressure and volume are obtained via an iterative procedure at each time step. Hence, the three data comparison points that were mentioned, were the only ones available for calibration purposes, when using the proposed approach. The proposed calibration method is illustrated schematically in Fig. 7.19.

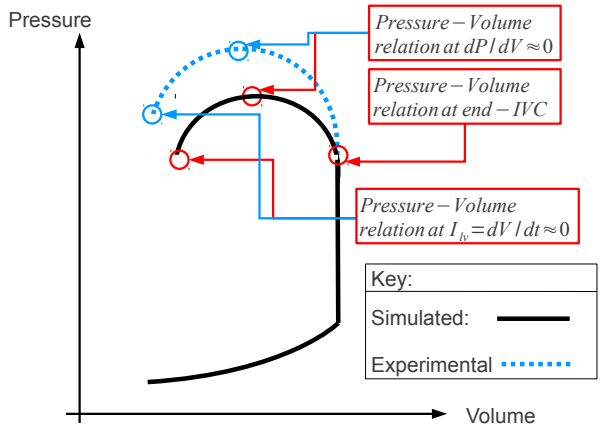

FIGURE 7.19: Schematic representation of the data comparison points used for material parameter identification during the systolic phase of the cardiac cycle.

Next, at each time step of a simulation in the systolic phase, both the cavity pressure and the volume are obtained via an iterative procedure and therefore both depend implicitly upon the material parameters. For this reason, the cost function for active tension and Windkessel parameter identification, is of bi-objective form. Using the weighted sum approach, a bi-objective cost function can be expressed in vector-matrix notation as

$$J^2(\boldsymbol{\theta}) = \frac{1}{2}\left[\omega_1 \left(\hat{\mathbf{V}} - \mathbf{V}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{V}}} \left(\hat{\mathbf{V}} - \mathbf{V}(\boldsymbol{\theta})\right) + \omega_2 \left(\hat{\mathbf{P}} - \mathbf{P}(\boldsymbol{\theta})\right)^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{P}}} \left(\hat{\mathbf{P}} - \mathbf{P}(\boldsymbol{\theta})\right)\right], \tag{7.17}$$

where the experimental and simulated cavity volume vectors have the following components:

$$\hat{\mathbf{V}}^{\mathrm{T}} = \left[\hat{V}_{\text{End-IVC}}, \hat{V}_{\frac{dP}{dt}=0}, \hat{V}_{\frac{dV}{dt}=0}\right] \ , \ \mathbf{V}(\boldsymbol{\theta})^{\mathrm{T}} = \left[V(\boldsymbol{\theta})_{\text{End-IVC}}, V(\boldsymbol{\theta})_{\frac{dP}{dt}=0}, V(\boldsymbol{\theta})_{\frac{dV}{dt}=0}\right]. \tag{7.18}$$

Also, the experimental and simulated cavity pressure vectors, have the following components:

$$\hat{\mathbf{P}}^{\mathrm{T}} = \left[\hat{P}_{\text{End-IVC}}, \hat{P}_{\frac{dP}{dt}=0}, \hat{P}_{\frac{dV}{dt}=0}\right] \ , \ \mathbf{P}(\boldsymbol{\theta})^{\mathrm{T}} = \left[P(\boldsymbol{\theta})_{\text{End-IVC}}, P(\boldsymbol{\theta})_{\frac{dP}{dt}=0}, P(\boldsymbol{\theta})_{\frac{dV}{dt}=0}\right]. \tag{7.19}$$

Furthermore, the diagonal weight matrix, $\mathbf{W}^{\hat{\mathbf{V}}}$, in Eq. (7.17), has components given by

$$\mathrm{W}_{ij}^{\hat{\mathbf{V}}} = \begin{cases} \frac{1}{\left[\hat{V}_i\right]^2} \text{ for } i = j \\ 0 \text{ for } i \neq j. \end{cases} \tag{7.20}$$

Similarly, the diagonal weight matrix, $\mathbf{W}^{\hat{\mathbf{P}}}$, in Eq. (7.17), has components given by

$$\mathrm{W}_{ij}^{\hat{\mathbf{P}}} = \begin{cases} \frac{1}{[\hat{P}_i]^2} \text{ for } i = j \\ 0 \text{ for } i \neq j. \end{cases} \tag{7.21}$$

It should be noted that the simulated cavity volume and pressure vectors are produced by simulating the systolic phase, using the calculation control mode outlined in Sec. 5.4.1 of Chap. 5. The proposed calibration approach has the advantage of being independent of the time duration, required for the simulation of systolic phase. Consequently, the method allowed larger parameter perturbations, for the computation of the Jacobian matrices, which avoids the problem of numerical rounding errors. The approach, however, has certain drawbacks, which will be explained in the following section.

Drawbacks of Proposed Calibration Method

It was found that the proposed calibration approach required a very good initial starting parameter guess, within 5 percent of the optimal solution. The reason for this observation, is explained by considering the number of data points used for comparison of simulated and experimental pressure-volume relations. In the proposed approach, only three data comparison points are available. These three data points do not capture the high degree of non-linearity in the ejection curve, especially towards the end of the phase, where the pressure-volume gradient ($\frac{\Delta P}{\Delta V}$) rapidly tends towards infinity, as the flow rate converges to zero (see Fig. 7.19). For a robust calibration approach, that is less dependant on the starting parameter guess, it is important that the simulated and experimental data points, are concentrated in areas of high curve non-linearity. This requirement shall be explained by firstly considering the Jacobian matrices, required for the parameter identification. In the context of active and systemic parameter identification, the computed Jacobian matrices are, $\mathbf{J}^{\mathbf{P}}$ and $\mathbf{J}^{\mathbf{V}}$, representing the sensitivity of the simulated pressure and volume vectors, respectively, to perturbations in the material parameters. Both these matrices are of size $n \times m$, where n and m refer to the number of comparison points and the size of the parameter vector, respectively. Next, consider the computation of the search direction, $\delta\boldsymbol{\theta}_u^k$, which is achieved via Eq. (6.36). Referring to Eq. (6.36), a Hessian matrix, $\mathbf{H}$, is required to compute the search direction. This Hessian matrix is of size $m \times m$ and its components are computed as

$$\mathbf{H} = \omega_1 (\mathbf{J}^{\mathbf{P}})^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{P}}} \mathbf{J}^{\mathbf{P}} + \omega_2 (\mathbf{J}^{\mathbf{V}})^{\mathrm{T}} \mathbf{W}^{\hat{\mathbf{V}}} \mathbf{J}^{\mathbf{V}}, \tag{7.22}$$

where ω_1 and ω_2 are weighting coefficients, and $\mathbf{W}^{\hat{\mathbf{P}}}$ and $\mathbf{W}^{\hat{\mathbf{V}}}$ are weight matrices. The Hessian matrix, will have the same size, irrespective of the number of data comparison points. However, fewer data comparison points, implies that each element of the Hessian matrix, will have fewer contributions from the columns of the Jacobian matrices and the

rows of their transpose. The resulting Hessian matrix is thus less dense and less refined, than if more data points are included in the calibration. The less dense nature of the Hessian matrix, directly affects the accuracy of the computed search direction. Consequently, for highly non-linear curve fitting applications, the robustness of the calibration approach, is directly affected by the number of included data comparison points.

7.3.3 Results of Cardiac Calibration

In this section the results of the cardiac calibration will be presented. This section will include the following topics:

- Results for passive material parameter identification.
- Results for active tension and Windkessel parameter identification.

As was the case for the cantilever calibration, the projected Levenberg-Marquardt algorithm was utilized as a means of parameter identification. For the passive parameter identification, the algorithm was used to minimize the cost functional given by Eq. (7.13). In addition, for active tension and Windkessel parameter identification, the algorithm was employed to minimize the cost function given by Eq. (7.17). For the inverse problem pertaining to the cardiac calibration, the line search procedure utilized, was the inexact backtracking Armijo type, which was explained in section 6.4.4.2 of Chap. 6. This procedure, was preferred over the cubic interpolation method, due to the requirement of fewer function evaluations, at each line search iteration, resulting in improved computational time. Furthermore, the stopping criteria for the algorithm were set to the following values,

- $J^2(\boldsymbol{\theta}^{k+1}) < 10^{-6}$,
- $||\boldsymbol{\nabla} J^2(\boldsymbol{\theta}^k)|| < 10^{-6}$,
- $\frac{||\delta\boldsymbol{\theta}^k||}{||\boldsymbol{\theta}^k||} < 10^{-5}$,
- $k < 50$.

Finally, the maximum number of line search iterations ($t^{\max}$) was set to 10. The algorithmic constant, μ_0 (Eq. (6.61)), utilized for the calibration was set to a value of 10^{-2}, which was found to be the optimal value that maximized the rate of convergence of the projected Levenberg-Marquardt algorithm (i.e. the number of iterations required by the algorithm to minimize the cost function).

7.3.3.1 Results for Passive Parameter Identification

The passive parameter vector to be identified was of the form,

$$\boldsymbol{\theta}^{\mathrm{T}} = [A, A_{\mathrm{comp}}]\,,$$

with the remaining passive parameters, a_1 to a_6, neglected from the optimization. However, values of the constants a_1 to a_6, for a canine left ventricle, were found previously by the research group that this work belongs to. The utilized parameter values for the constants, a_1 to a_6, are summarized in Tab. 7.4.

TABLE 7.4: Parameter values for the passive material constants, a_1 to a_6.

Parameter	Parameter Value
a_1	-6.21
a_2	-5.11
a_3	9.07
a_4	12
a_5	12
a_6	-5.33

In addition, for parameter identification, the upper bound of the parameter search space, $\boldsymbol{\theta}^{\max}$, was established as

$$\boldsymbol{\theta}^{\max} = \begin{pmatrix} A^{\max} \\ A^{\max}_{\mathrm{comp}} \end{pmatrix} = \begin{pmatrix} 1.0\,\mathrm{kPa} \\ 300\,\mathrm{kPa} \end{pmatrix}, \tag{7.23}$$

and similarly, the lower bound, $\boldsymbol{\theta}^{\min}$, was set to

$$\boldsymbol{\theta}^{\min} = \begin{pmatrix} A^{\min} \\ A^{\min}_{\mathrm{comp}} \end{pmatrix} = \begin{pmatrix} 0.1\,\mathrm{kPa} \\ 50\,\mathrm{kPa} \end{pmatrix}. \tag{7.24}$$

The lower bound value for the parameter $A^{\min}_{\mathrm{comp}}$, was set to a value of 50 kPa, which was small enough to still enforce near incompressibility of the myocardium of the left ventricle.

The inverse problem of passive parameter identification, was conducted using four different starting parameters, in order to improve the likelihood of the obtained solution, being a global minimum. The four separate starting parameter values used are summarized by Run 1 to Run 4 in Tab. 7.5.

TABLE 7.5: Various initial parameter guesses used to test for global convergence of the projected Levenberg-Marquardt algorithm

Parameter	Run 1	Run 2	Run 3	Run 4
A(kPa)	0.1	1.0	0.5	0.1
A_{comp}(kPa)	50	300	175	300

Run 1 and Run 2, in Tab. 7.5, represent starting parameter values, at the lower and upper boundaries of the parameter search space. Run 3 represents a set of starting parameters, in the center of the parameter search space. Lastly, Run 4 represents an initial parameter guess, where A and A_{comp}, are set to their lower and upper boundary values, respectively. The results of the inverse problem are summarized in Tab. 7.6.

TABLE 7.6: Initial guesses and final values, found for passive material parameters

Parameter	Run 1	Run 2	Run 3	Run 4
A^{initial} (kPa)	0.1	1.0	0.5	0.1
$A^{\text{initial}}_{\text{comp}}$ (kPa)	50	300	175	300
A^{final} (kPa)	0.289	0.294	0.301	0.292
$A^{\text{final}}_{\text{comp}}$ (kPa)	98.2	101.6	98.8	100.3
$J^2(\boldsymbol{\theta})$	0.00495	0.00468	0.00205	0.00378
$\|\|\boldsymbol{\nabla} J^2(\boldsymbol{\theta})\|\|$	0.0028	0.0023	0.0013	0.0016
$\frac{\|\|\delta\boldsymbol{\theta}k\|\|}{\|\|\boldsymbol{\theta}\|\|}$	3×10^{-6}	4×10^{-8}	1×10^{-6}	1.5×10^{-7}

Referring to Tab. 7.6, it is clear that the algorithm converges to a parameter value for A, of between 0.28 kPa and 0.31 kPa, for all four calibration attempts. Similarly, the parameter values for A_{comp}, collect between 98 kPa and 102 kPa. The termination criteria, that stops the calibration, is ratio of the norm of the parameter increment, to that of the parameter vector, which falls below the specified threshold of 10^{-5}, for each calibration attempt. Consequently, the algorithm always terminated, when the parameters stopped changing significantly. Also, in Tab. 7.6, it is observed that for each calibration attempt, the cost function at the final parameter set is not as small as what is expected. However, the norms of the cost function gradient vectors, tend to be of the order of 10^{-3}, indicating that the achieved parameters fall in the vicinity of a stationary point. The best estimate of the parameters that produce the local minimum, are those which produce the smallest cost function value. Hence, the selected parameter set, that minimizes the cost function is clearly the set produced from Run 3. Fig. 7.20 illustrates the diastolic filling curve, based on the parameter set from Run 3, against the reference data, extracted from Kerckhoffs et al. [17].

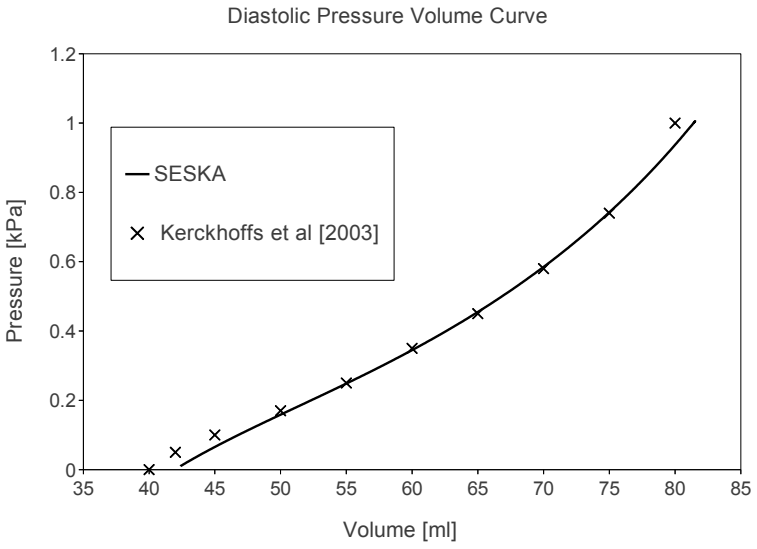

FIGURE 7.20: Illustration of the diastolic filling curve, obtained at the final set of passive parameters.

Referring to Fig. 7.20, the reasons for the relatively large cost function values in Tab. 7.6, can be explained. Firstly, at the start of diastolic filling, the cavity volume differs from that of the Kerckhoffs model by approximately 2.0 ml. In addition, this discrepancy, is independent of the material parameters. Consequently, in the low pressure region of the diastolic filling curve (0 kPa to 0.18 kPa), the projected Levenberg-Marquardt algorithm struggles to minimize the difference, between simulated and reference data. In addition, by observing the latter portion of the diastolic curve (0.74 kPa to 1.0 kPa), it is clear that the mismatch between experimental and simulated data, has not been sufficiently minimized, which in turn contributes to the higher final cost function value. The reason for the algorithm's inability to minimize this mismatch, arises from the exclusion of the exponential constants, a_1 to a_6, which tend to govern the gradient of the diastolic curve more pronouncedly in the high pressure regions. However, these exponential terms caused numerical instability when included in the optimization, which lead to their exclusion in the calibration attempts. Nevertheless, by observing Fig. 7.20, it is evident that the obtained parameters from run 3, generally produce a good fit for the diastolic filling curve, especially in the intermediate pressure range (0.18 kPa to 0.74 kPa). Consequently, the parameters from Run 3, were accepted as the final calibrated passive parameters for the model.

7.3.3.2 Results for Active and Windkessel Parameter Identification

Firstly, a value of unity was chosen for the weighting coefficients ω_1 and ω_2 in Eq. (7.17). Due to the formulation of the weight matrices, $\mathbf{W}^{\hat{\mathbf{P}}}$ and $\mathbf{W}^{\hat{\mathbf{V}}}$ (see Eq. (7.20) and Eq. (7.21) respectively), a degree of weighting is already assigned to the separate objectives of the bi-objective cost function in Eq. (7.17), which appears to be the reason why a value of unity for ω_1 and ω_2 was sufficient. The parameter vector to be identified was of the form

$$\boldsymbol{\theta}^{\mathrm{T}} = [T_{\max}, B, m, C, R, R_0]\,.$$

In addition, the upper bound of the parameter search space, $\boldsymbol{\theta}^{\max}$, was established as

$$\boldsymbol{\theta}^{\max} = \begin{pmatrix} T_{\max}^{\max} \\ B^{\max} \\ m^{\max} \\ C^{\max} \\ R^{\max} \\ R_0^{\max} \end{pmatrix} = \begin{pmatrix} 200\,\mathrm{kPa} \\ 4.75 \times 10^3\,\mathrm{mm} \\ 3.0{\times}10^3\,\mathrm{mm} \\ 4500\,\frac{\mathrm{mm}^3}{\mathrm{kPa}} \\ 1.0{\times}10^{-4}\,\frac{\mathrm{kPa{\cdot}s}}{\mathrm{mm}^3} \\ 9.0{\times}10^{-6}\,\frac{\mathrm{kPa{\cdot}s}}{\mathrm{mm}^3} \end{pmatrix}, \tag{7.25}$$

and similarly, the lower bound, $\boldsymbol{\theta}^{\min}$, was set to

$$\boldsymbol{\theta}^{\min} = \begin{pmatrix} T_{\max}^{\min} \\ B^{\min} \\ m^{\min} \\ C^{\min} \\ R^{\min} \\ R_0^{\min} \end{pmatrix} = \begin{pmatrix} 100\,\mathrm{kPa} \\ 1.00 \times 10^3\,\mathrm{mm} \\ 1.0{\times}10^3\,\mathrm{mm} \\ 1000\,\frac{\mathrm{mm}^3}{\mathrm{kPa}} \\ 1.0{\times}10^{-5}\,\frac{\mathrm{kPa{\cdot}s}}{\mathrm{mm}^3} \\ 1.0{\times}10^{-6}\,\frac{\mathrm{kPa{\cdot}s}}{\mathrm{mm}^3} \end{pmatrix}. \tag{7.26}$$

The parameter search space was established, such that all points contained within the hyperbox, possessed cost function values that were within five percent of that at the optimal solution. The optimal solution is unknown, however, its cost function value is required to be close to zero, i.e., $J^2(\boldsymbol{\theta}^{\mathrm{opt}}) \approx 0$. The reasons for the requirement of a narrowly constrained parameter search space have been discussed in Sec. 7.3.2.2. The parameter search space, was then navigated manually, to produce the best initial starting parameter guess. The initial parameter guess, for the inverse problem is given in Tab. 7.7.

TABLE 7.7: Initial guesses for active and Windkessel parameters, utilized for the systolic calibration

Parameter	Initial Value
C $(\frac{\text{mm}^3}{\text{kPa}})$	3000
R $(\frac{\text{kPa·s}}{\text{mm}^3})$	3.0×10^{-5}
R_0 $(\frac{\text{kPa·s}}{\text{mm}^3})$	1.5×10^{-6}
B (mm)	2.8×10^{3}
T_{max} (kPa)	135.7
m $(\frac{\text{s}}{\text{mm}})$	1.5×10^{3}

For the active tension and Windkessel parameter identification, only one initial parameter guess was utilized, due to the significant amount of associated computational time (refer to section 7.3.1.1), which did not warrant a full convergence analysis. Tab. 7.8 summarizes the results of the inverse parameter identification.

TABLE 7.8: Final active and Windkessel parameters identified using the projected Levenberg-Marquardt algorithm

Parameter	Final Value
C $(\frac{\text{mm}^3}{\text{kPa}})$	4476.2
R $(\frac{\text{kPa·s}}{\text{mm}^3})$	8.13×10^{-5}
R_0 $(\frac{\text{kPa·s}}{\text{mm}^3})$	6.43×10^{-6}
B (mm)	2.48×10^{3}
T_{max} (kPa)	136.9
m $(\frac{\text{s}}{\text{mm}})$	2.047×10^{3}
$J^2(\boldsymbol{\theta})$	0.0018
$\|\|\boldsymbol{\nabla} J^2(\boldsymbol{\theta})\|\|$	0.00034
$\frac{\|\|\delta\boldsymbol{\theta}\|\|}{\|\|\boldsymbol{\theta}\|\|}$	1.78×10^{-8}

Referring to Tab. 7.8, the stopping criteria that terminated the algorithm, was again the ratio of the norm of the parameter increment, to that of the original parameter vector. Consequently, the algorithm stopped, when parameter values stop changing significantly. However, the value of the norm of the cost function gradient vector, which is of the order of 10^{-4}, indicates that the obtained parameter set does in fact represent a stationary point (local minima). Fig. 7.21 illustrates the systolic pressure-volume curve, produced using the parameters in Tab. 7.8. Also, the reference pressure-volume data [17] on Fig. 7.21, includes more data points, than what was used in the calibration. More reference data points have been included in Fig. 7.21, to illustrate the quality of the fit, based on the results of the inverse problem.

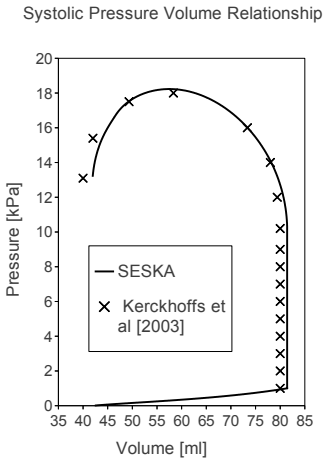

FIGURE 7.21: Illustration of the systolic pressure-volume curve, obtained at the final set of active and Windkessel parameters.

Referring to Fig. 7.21, a discrepancy between the SESKA model and that of Kerckhoffs et al. [17], is noticed in the isovolumetric contraction phase. This discrepancy occurs due to the difference between the end-diastolic volumes of the two models (see Fig. 7.20). In addition, it is clear that there is a mismatch of the pressure-volume relationship at the end of systole, between the SESKA model and that by Kerckhoffs et al. However, this mismatch was accepted, as it lead to a near closed loop pressure-volume curve, which is illustrated in Fig. 7.22.

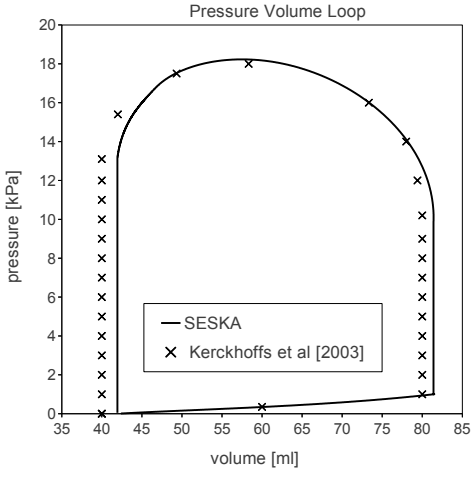

FIGURE 7.22: Illustration of the final pressure-volume loops, achieved using the identified set of active, passive and Windkessel parameters.

Finally, the obtained set of active tension and Windkessel parameters are validated, by observing the differences between the durations of ejection and isovolumetric contraction (Δt_{ej} and Δt_{ivc} respectively), for the SESKA model and that by Kerckhoffs et al.

These differences are summarized in Tab. 7.9. Tab. 7.9, also presents the durations of isovolumetric relaxation (Δt_{ivr}), for the SESKA model and that found by Kerckhoffs et al.

TABLE 7.9: Time durations for IVC, ejection and IVR for the SESKA model and that of Kerckhoffs et al. [17]

Parameter	SESKA	Kerckhoffs et al [2003]
Δt_{ivc} (ms)	52	44
Δt_{ej} (ms)	212	194
Δt_{ivr} (ms)	397	250

Referring to Tab. 7.9, it is clear that the simulated durations of isovolumetric contraction and ejection, are close to those found by Kerckhoffs et al. However, the duration of isovolumetric relaxation, for the SESKA model, is almost 147 ms greater than that found by Kerckhoffs et al. The reason for this discrepancy is due to the active tension parameter, m, which governs the duration of the relaxation period of the active tension-time distribution [29]. The majority of the relaxation period, of the active tension time distribution, occurs within the isovolumetric relaxation phase. Since the proposed approach to active tension and Windkessel parameter identification, does not include pressure-volume data from the isovolumetric relaxation phase, the estimate for the parameter, m, does not represent the value required to achieve the relaxation period, found in the reference data. However, the author has chosen to proceed with the identified parameter set, as they produce a closed loop pressure-volume curve, which is the primary objective of this work.

7.3.3.3 Summary

The projected Levenberg-Marquardt algorithm has been utilized to identify passive, active tension and Windkessel parameters, for a canine left ventricle, based on reference data extracted from the works of Kerckhoffs et al. [17]. The identified passive, active tension and Windkessel parameters are summarized in Tabs. 7.10, 7.11 and 7.12 respectively.

TABLE 7.10: Final identified passive material parameters.

Parameter	Value
A^{final} (kPa)	0.301
$A^{\text{final}}_{\text{comp}}$ (kPa)	98.8
a_1	-6.21
a_2	-5.11
a_3	9.07
a_4	12
a_5	12
a_6	-5.33

TABLE 7.11: Final identified active material parameters.

Parameter	Value
Ca_0 (μM)	4.35
$(Ca_0)_{\text{max}}$ (μM)	4.35
l_0 (mm)	1.58×10^{-3}
t_0 (ms)	100
B (mm)	2.48×10^3
T_{max} (kPa)	136.9
m ($\frac{\text{s}}{\text{mm}}$)	2.047×10^3

TABLE 7.12: Final identified three element Windkessel parameters.

Parameter	Value
C ($\frac{\text{mm}^3}{\text{kPa}}$)	4476.2
R ($\frac{\text{kPa}\cdot\text{s}}{\text{mm}^3}$)	8.13×10^{-5}
R_0 ($\frac{\text{kPa}\cdot\text{s}}{\text{mm}^3}$)	6.43×10^{-6}

7.3.4 Mechanical Analysis

The idealized canine ellipsoidal model was calibrated to reference pressure-volume data, extracted from the work of Kerckhoffs et al. [17]. Consequently, the obtained material parameters will be used to conduct a mechanical analysis of the behaviour of the left ventricle, over a single beat of a cardiac cycle. In the mechanical analysis, the primary focus will be left ventricular deformations, as these results highlight several key aspects

with regards to the manner in which left ventricular cardiac function is modelled. However, an analysis of the stresses and strains in the left ventricle shall also be provided in this section.

7.3.4.1 Selection of an Appropriate Mesh Size

The idealized canine left ventricle used for calibration, had a relatively coarse particle distribution consisting of 363 particles. In order to achieve convergence of the EFGM method, the particle distribution had to be further refined. To obtain an appropriate particle distribution for convergence, models with 1299, 2724 and 2884 particles were tested. Fig. 7.23 illustrates the models utilized to find an appropriate mesh size, for convergence of the EFGM method in SESKA.

Convergence was established by simulating the diastolic filling phase, for the 1299, 2724

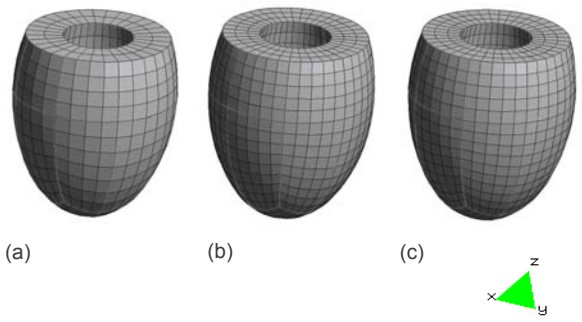

FIGURE 7.23: (a) 1299 particle model. (b) 2724 particle model (c) 2884 particle model

and 2884 particle models (Fig. 7.23 (a), (b) and (c) respectively). Boundary conditions for the models, were the same as those mentioned in Sec. 7.3.1.2. Furthermore, passive, active tension and Windkessel material parameters for the model, are summarized in Tabs. 7.10, 7.11 and 7.12. Fig. 7.24 illustrates the diastolic filling pressure-volume curves for each model. Referring to Fig. 7.24, it is clear that the cavity volume computed at each pressure step is the same for the 2724 and 2884 particle models. Hence, the 2884 particle model represents the particle distribution required for convergence of the EFGM method.

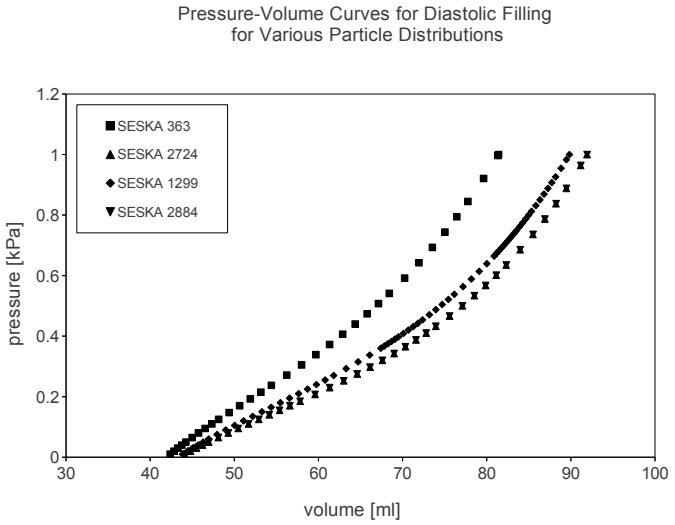

FIGURE 7.24: Diastolic pressure-volume curves for various mesh sizes.

7.3.4.2 Pressure-Volume Loop

The pressure-volume loops for the 363 and 2884 particle canine left ventricle models, are illustrated in Fig. 7.25. It should be noted that for the 2884 particle model, the phases consisting of isovolumetric contraction, ejection and isovolumetric relaxation, where simulated using the calculation control mode, outlined in Sec. 5.4.2 of Chap. 5.

Referring to Fig. 7.25, the end-diastolic volume for the 2884 particle model is ap-

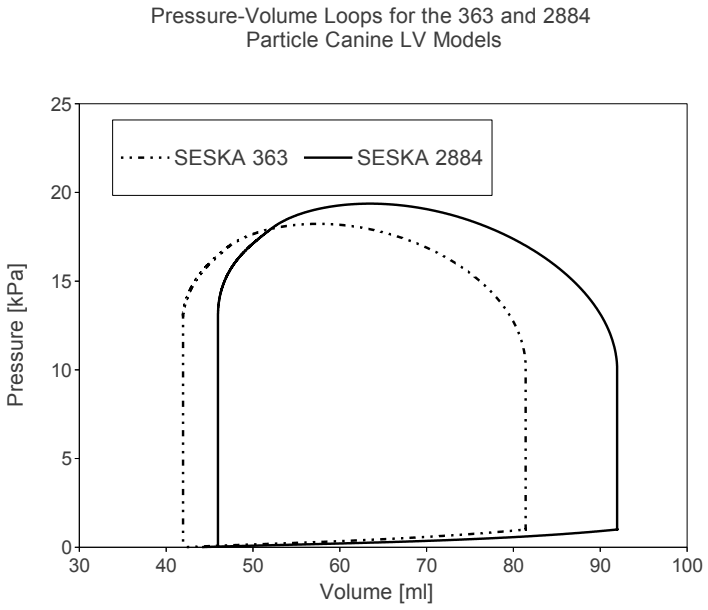

FIGURE 7.25: Final pressure-volume loop for the 2884 particle canine left ventricle mesh.

proximately 92 ml, as compared to that of the 363 particle model used for calibration, whose end-diastolic volume was around 81 ml. In addition, the peak and end-systolic pressures, have increased from 18 kPa and 13.1 kPa, to 19.3 kPa and 13.9 Kpa, for the 363 and 2884 particle models, respectively. Furthermore, the cavity volumes at which peak and end-systolic pressures are achieved, have increased from 58 ml and 42 ml, to 64 ml and 47 ml, for the 363 and 2884 particle models, respectively. Consequently, it appears that the particle distribution used for the calibration, was in fact not fine enough for convergence of the EFGM method. However, the computational time associated with the diastolic filling for the 2884 particle model, was approximately 15 hours, using 30 processors. In addition, the computational time for isovolumetric contraction, ejection and isovolumetric relaxation together, was approximately two and a half weeks, for the 2884 particle model, despite the fact that 30 processors were used for the simulation. The aspect of computational time, is once again the reason that the author has selected to conduct the inverse calibration problem, on a coarse 363 particle model. Using a refined particle distribution, such as the 2884 particle model, would yield unacceptable computational times, for calibration attempts that are not necessarily guaranteed to converge to a local or global minimum. However, based on Fig. 7.25, it is observed that a closed pressure-volume curve is achieved, for both the 363 and 2884 particle models. Since the objective of this work was to achieve this closed loop, the parameters were accepted for the following mechanical analysis.

7.3.4.3 Left Ventricular Twist Behaviour

Fig. 7.26 illustrates the displacements in the y direction of the left ventricle, at representative points in the cardiac cycle. Figs. 7.26 (a), (b), (c), (d), (e), and (f), represent the y displacements, at start-diastolic filling, end-diastolic filling, end-isovolumetric contraction, end-ejection, mid-isovolumetric relaxation and end-isovolumetric relaxation respectively.

Firstly, recall that diastolic filling is the final phase of diastole, with the initial phase being isovolumetric relaxation. In similar ellipsoid model studies of the left ventricle (e.g. Niederer and Smith [27] who modelled a rat left ventricle), the body exhibits global counter-clockwise twisting during systole, due to the initiation and rise of the time varying active tension, as well as the orthotropic material behaviour of the myocardium. In these studies, the ventricle then begins to untwist during isovolumetric relaxation, due to the decrease of active tension. Then, during the initial phases of diastolic filling, the left ventricle twists in a global clockwise manner, which causes further untwisting, which is atrributed to the orthotropic behaviour of the myocardium and the increasing blood

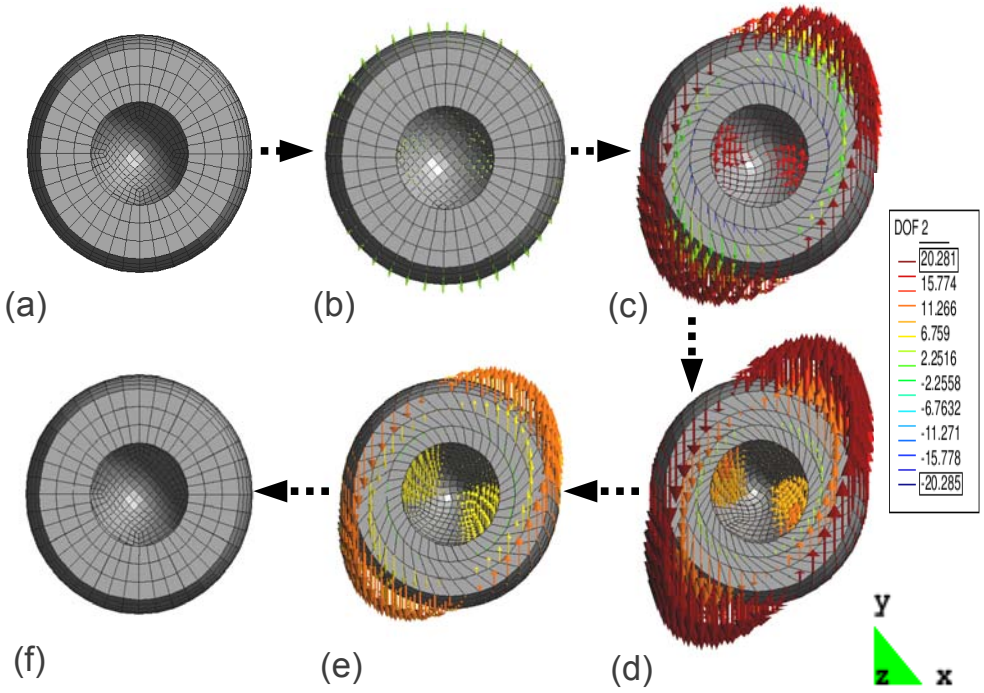

FIGURE 7.26: Displacements of the canine left ventricle, in the y direction, at representative points in the cardiac cycle.

pressure exerted on the endocarium wall, the latter due to suction of blood into the ventricle from the left atrium. In addition, towards the end of diastolic filling, atrial systole occurs, where the left atrium begins to contract and therefore pushes blood into the left ventricle, i.e., slow filling. During atrial systole, the left ventricle is observed to almost completely untwist and also expands in volume and size. However, referring to Figs. 7.26 (a) to (b), in our computational model of the left ventricle, no untwisting occurs over the duration of diastolic filling. The reason for the absence of untwisting in diastolic filling for our model, occurs as the phase commences from the reference configuration of the ventricle, which is in an undeformed and twist free state. Thus the increasing cavity pressure during diastolic filling appears to cause only small and almost negligible amounts of global clockwise twisting of the ventricle, and the observed governing mode of deformation in the left ventricle, is outward cavity expansion. However, Niederer and Smith [27] commence their diastolic filling phase from the deformed geometry and cavity pressure at the end of isovolumetric relaxation. At this point, Niederer and Smith [27] report that the left ventricle has not completely untwisted. Consequently, the authors observe global clockwise twisting of the ventricle, due to the blood filling pressure and orthotropic material behaviour of the myocardium, which almost completely untwists the body by the end of this phase. In this work, the approach utilized to simulate the diastolic filling phase is not incorrect and has been utilized by Kerckhoffs et al. [17],

as well as Bovendeerd et al. [12], to simulate a complete beat of a cardiac cycle for a canine left ventricle. Since the results of Kerckhoffs et al. [17] were used for calibration of material parameters, the author has chosen to simulate the cardiac cycle in the same manner as the aforementioned author.

Besides the phase of diastolic filling, the left ventricle behaviour for the remaining phases, in terms of global twist, appears to be close to that which has been observed in similar ellipsoid model studies [17, 27]. Firstly, a rapid increase in global counter-clockwise twisting is noted from end-diastolic filling to end-isovolumetric contraction (Figs. 7.26 (b) and (c) respectively), due to the initiation of the active contraction throughout the ventricle, as well as the orthotropic material behaviour of the myocardium.

The global counter-clockwise twisting, continues to increase from end-isovolumetric contraction to end-ejection (Figs. 7.26 (c) and (d) respectively), due to the active contraction, orthotropic material behaviour of myocardium and additionally, the ability of the cavity to now deform inwards or dilate. In addition, during ejection, the rate of increase of global counter-clockwise twist, increases initially and then decreases once the time to maximum active tension, t_0, is reached, which initiates the downhill portion of the active tension-time distribution.

From end-ejection to end-isovolumetric relaxation (Figs. 7.26 (d) to (f)), the active contraction in the ventricle decreases gradually with time, which in turn causes the left ventricle to almost completely untwist.

During isovolumetric contraction, relaxation and ejection (Figs. 7.26 (c) to (f)), the global counter-clockwise twisting of the ventricle generally increases longitudinally from the base to the mid-section, due to increased restraint nearer the former and additionally, particles nearer the latter possessing a larger lever arm. In addition, for these three phases, the global counter-clockwise twist generally decreases from the mid-section to the apex, due to particles nearer the latter possessing a smaller lever arm.

7.3.4.4 Physiologically Observed Left Ventricular Twist Behaviour

The twist behaviour illustrated by our ellipsoid model, although similar to other model studies, is still unphysiological. Buckberg et al. [16] described the left ventricle's twisting motion for a complete beat of a cardiac cycle, based on observations made from MRI images. Fig. 7.27 is an extract from the paper by Buckberg et al. [16], which illustrates the physiologically observed direction of twist at the base and apex of the left ventricle,

at representative points in the cardiac cycle.

Referring to Fig. 7.27, during isovolumetric contraction the left ventricle is observed

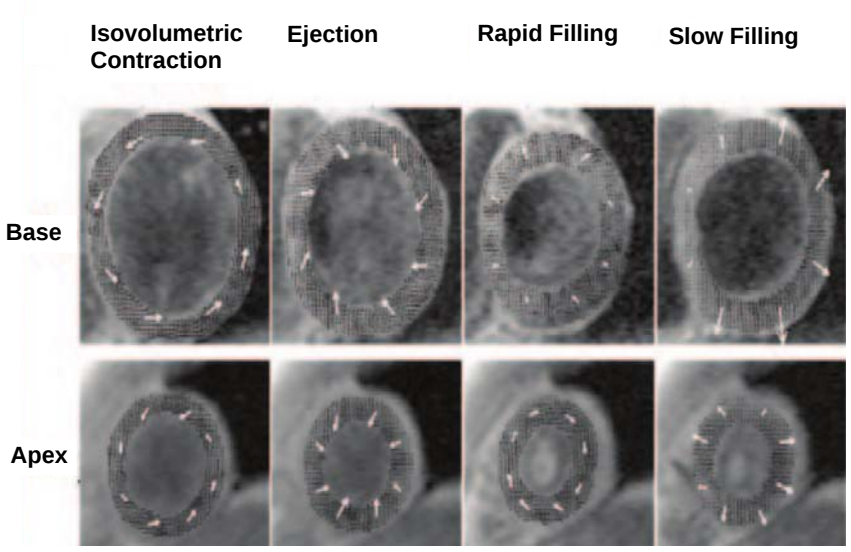

FIGURE 7.27: Physiologically observed direction of twist at the base and apex of the left ventricle, at representative points in the cardiac cycle [16].

to twist in a global counter-clockwise manner at all points in the body, which is consistent with our left ventricle model. However, during ejection, clockwise twisting is observed near the base and counter-clockwise twisting is observed near the apex. This latter observation differs from what is observed in our model, where the twist is globally counter-clockwise throughout the ventricle, in ejection. The difference between our model and the observed physiological behaviour during ejection, can be explained three-fold:

- Firstly, in the physiological case, the active contraction in the ventricle does not occur synchronously, which is an aspect that is not taken into account in our computational model.
- Secondly, the mechanics of the left ventricle are implicitly affected by that of the right ventricle, the latter providing increased restraint to the deformation and twist of the former. Obviously, in our current model of the heart, there is no representation of the right ventricle and no associated coupling of its mechanics to that of the left ventricle.
- Thirdly, the base is not fixed at the endocardium but only partly, i.e. elastically restrained by the major blood vessel connected to it.

Consequently, if the above considerations are accounted for, then, the physiological behaviour of the left ventricle in terms of twist may be reproduced. However, for such a model, which would preferably posses a realistic geometry based on MRI images, a realistic description of the fibre field is also required in order to simulate the left ventricle behaviour accurately.

In the physiological case, in the rapid and slow filling phases of diastolic filling, the left ventricle continues to untwist. However, in our model, the ventricle does not untwist and the governing mode of deformation is primarily outward cavity expansion. The reason for this difference has already been explained, however, in summary, the reason for the absence of untwisting in diastolic filling for our model, occurs as the simulation of the cardiac cycle begins from the reference configuration of the left ventricle, which is an undeformed and twist free state.

7.3.4.5 Changes in Left Ventricle Cavity Shape and Wall Thickness

Next, Figs. 7.28 (a), (b), (c), (d), (e), and (f) illustrate the shape and wall thickness of the left ventricle cavity, at the various time points in the cardiac cycle, namely, start-diastolic filling, end-diastolic filling, end-isovolumetric contraction, end-ejection, mid-isovolumetric relaxation and end-isovolumetric relaxation respectively.

During diastolic filling (Figs. 7.28 (a) to (b)), the left ventricle cavity expands due

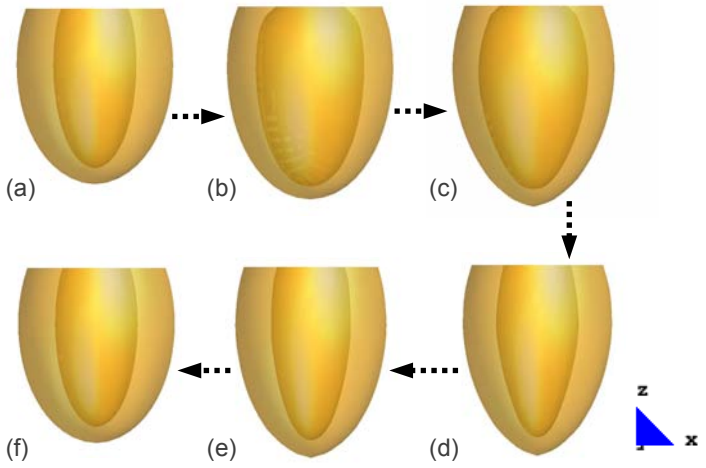

FIGURE 7.28: Shape and wall thickness of the left ventricle cavity, at representative time points in the cardiac cycle.

to blood filling pressure and a general decrease in wall thickness is observed. The cavity expansion in diastolic filling is restrained at the base, which leads to the increased

curvature of the cavity shape near this location. The observed changes in left ventricle cavity shape and wall thickness during diastolic filling, are consistent with those made by Niederer and Smith [27].

During isovolumetric contraction (Figs. 7.28 (b) to (c)), the cavity pressure must increase, such that no volume change occurs due to the initiation of the active contraction in the ventricle, which increases with time during this phase. Hence, a slight elongation of the cavity is observed, accompanied by a continued decrease of wall thickness from end-diastolic filling, both observations being consistent with those made by Niederer and Smith [27]. Overall, in this phase, it appears that generally the left ventricle cavity tends to want to become more spherical in nature.

During ejection (Figs. 7.28 (c) to (d)), the governing mode of deformation is inward cavity dilation, which is attributed to the active contraction and the fact that the cavity volume is un-restrained against volume change. Also, during ejection, the left ventricle cavity shortens longitudinally and a general increase in wall thickness is observed, both observations again corresponding to those made by Niederer and Smith [27].

During isovolumetric relaxation (Figs. 7.28 (d) to (f)), the left ventricle cavity is again restrained against volume change, however, its shape tends to that at the beginning of diastolic filling. In this phase, the cavity continues to shorten longitudinally and wall thickness generally increases and finally, the left ventricle arrives at a configuration which is almost identical to that at the beginning of diastolic filling. Provided that a closed loop pressure-volume curve is achieved, the left ventricle is expected to return to a configuration, that is close to its original un-deformed state at the beginning of diastolic filling and this is clearly observed in Fig. 7.28 (f).

7.3.4.6 Sarcomere Length Distribution

Figs. 7.29 (a), (b), (c), (d), (e), and (f) illustrate the sarcomere length distribution throughout the left ventricle, at the various time points in the cardiac cycle, namely, start-diastolic filling, end-diastolic filling, end-isovolumetric contraction, end-ejection, mid-isovolumetric relaxation and end-isovolumetric relaxation respectively.

From start to end-diastolic filling (Figs. 7.29 (a) to (b)), the sarcomere length increases at all locations in the ventricle, due to the increasing blood filling pressure, which expands the latter during this phase. The sarcomere lengths at all points in the ventricle at the end of diastolic filling represent the maximum values for entire cardiac

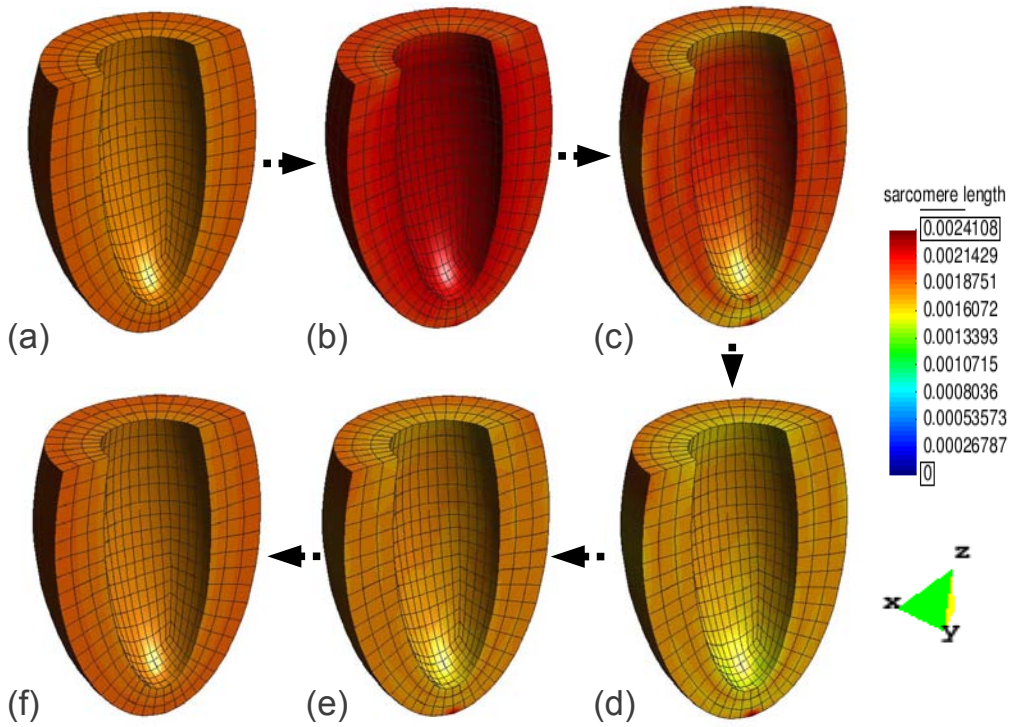

FIGURE 7.29: Sarcomere length distribution throughout the left ventricle, at representative time points in the cardiac cycle.

cycle.

During isovolumetric contraction (Figs. 7.29 (b) to (c)), an active contraction is initiated throughout the ventricle which increases with time, the result of which is a decrease of sarcomere length at all locations. However, during this phase, the sarcomere length decreases faster at locations nearer the apex and endocardium surface at the base.

During ejection (Figs. 7.29 (c) to (d)), the sarcomere lengths in the left ventricle continues to decrease. In addition, at the end of ejection it appears that the sarcomere lengths throughout the ventricle are similar.

During isovolumetric relaxation, the time varying active contraction is decreasing, hence, the sarcomere lengths in the left ventricle increase gradually and their distribution at the end of this phase is almost identical to that at the start of diastolic filling.

7.3.4.7 Left Ventricle Fibre Strains

Figs. 7.30 (a), (b), (c), (d), (e), and (f) illustrate the strain in the fibre direction throughout the left ventricle, at the various time points in the cardiac cycle, namely,

start-diastolic filling, end-diastolic filling, end-isovolumetric contraction, end-ejection, mid-isovolumetric relaxation and end-isovolumetric relaxation respectively.

During diastolic filling (Figs. 7.30 (a) to (b)), the left ventricle expands outwards

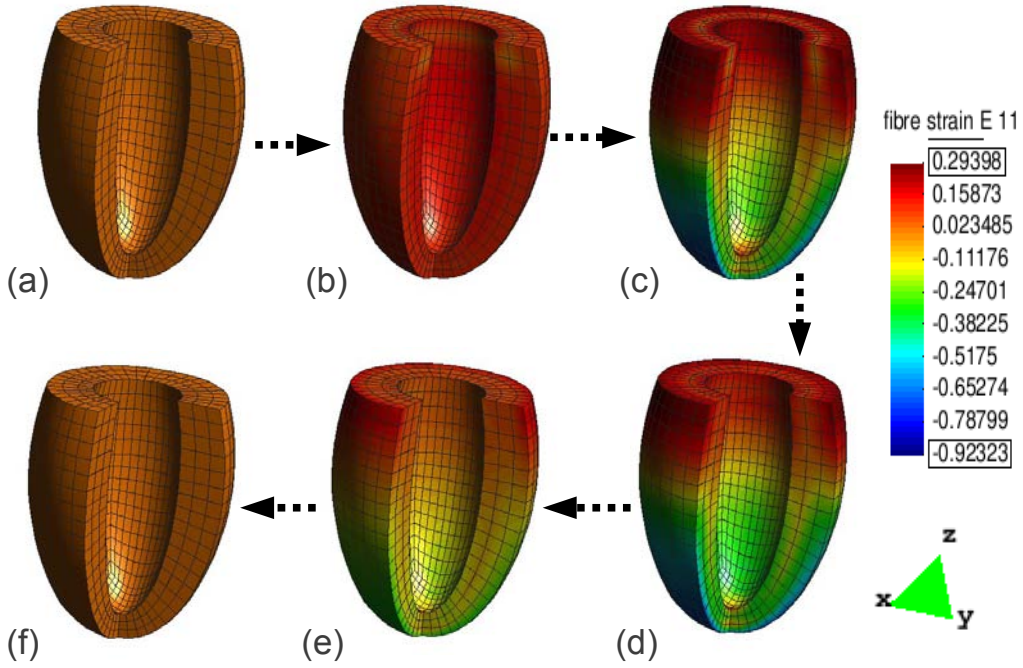

FIGURE 7.30: Strain in the fibre direction throughout the left ventricle, at representative time points in the cardiac cycle.

due to blood filling pressure. In this phase increasing blood filling pressure, causes sarcomere lengths to increase, the rate of which increases from start to end-diastolic filling. Hence, the associated increase of muscle fibre stretching over this phase, results in an increase of tensile strains in the fibre direction, throughout the ventricle. In addition, at the end of diastolic filling, a near homogeneous distributions of tensile strains in the fibre direction is observed, which is consistent with observations made by Kerckhoffs et al. [17]. Furthermore, at the start of diastolic filling (Fig. 7.30 (a)), the strains in the fibre direction throughout the ventricle are non zero, illustrating the presence of residual strains. The residual strains arise as the sarcomere rest length, l_R, is made to vary transmurally, with smaller values at the endocardium than at the epicardium.

During isovolumetric contraction (Figs. 7.30 (b) to (c)), the evolution of strain in the fibre direction is harder to interpret, due to the additional active stress component present in this phase. However, it is clear that the distribution of fibre strains in the left ventricle are very different at the start and end of isovolumetric contraction. To

elaborate, at the end of diastolic filling, fibre strains are primarily tensile throughout the ventricle, whereas at the end of isovolumetric contraction, some compressive fibre strains are noticed at the epicardium from the mid-section to the apex.

During ejection, it is expected that a more or less homogeneous distribution of the average fibre strain between the beginning and end of the phase, is observed (refer to Sec. 4.3). In our model, during ejection (Figs. 7.30 (c) to (d)), it is clear that the distribution of fibre strains throughout the ventricle, are similar at end-isovolumetric contraction and end-ejection. Consequently, in this phase, it appears that the constants p_1, p_2 and p_3, do in-fact yield a more or less homogeneous distribution of the average fibre strain between the beginning and end of ejection throughout the ventricle.

During isovolumetric relaxation (Figs. 7.30 (d) to (f)), the active tension in the ventricle is decreasing, causing the ventricle to relax and untwist. Thus, at the end of isovolumetric relaxation, the state of fibre strain in the left ventricle is almost identical to that at the beginning of diastolic filling, which is expected since a closed-loop pressure-volume curve has been achieved.

7.3.4.8 Left Ventricle Fibre Stresses

Figs. 7.31 (a), (b), (c), (d), (e), and (f) illustrate the stress in the fibre direction throughout the left ventricle, at the various time points in the cardiac cycle, namely, start-diastolic filling, end-diastolic filling, end-isovolumetric contraction, end-ejection, mid-isovolumetric relaxation and end-isovolumetric relaxation respectively.

Firstly, at the start of diastolic filling (Fig. 7.31 (a)), the fibre stresses throughout the left ventricle are zero, which is due to the absence of residual stresses. Furthermore, during diastolic filling (Figs. 7.31 (a) to (b)), the governing mode of deformation is outward cavity expansion. Hence, the associated stretching of muscle fibres contributes to an almost homogeneous state of tensile fibre stress throughout the ventricle, which very gradually increases from start to end-diastolic filling.

During the phases of isovolumetric contraction, relaxation and ejection, the development of fibre stress is a highly complex phenomenon, that is dependant on many aspects including torsion of the ventricle, fibre strain, active tension and passive stresses, along with many other contributors. In addition, it appears that many authors in the literature have conflicting opinions about why the left ventricle behaves the way it does, in terms of fibre stresses. Consequently, in this section, the author chooses to simply note

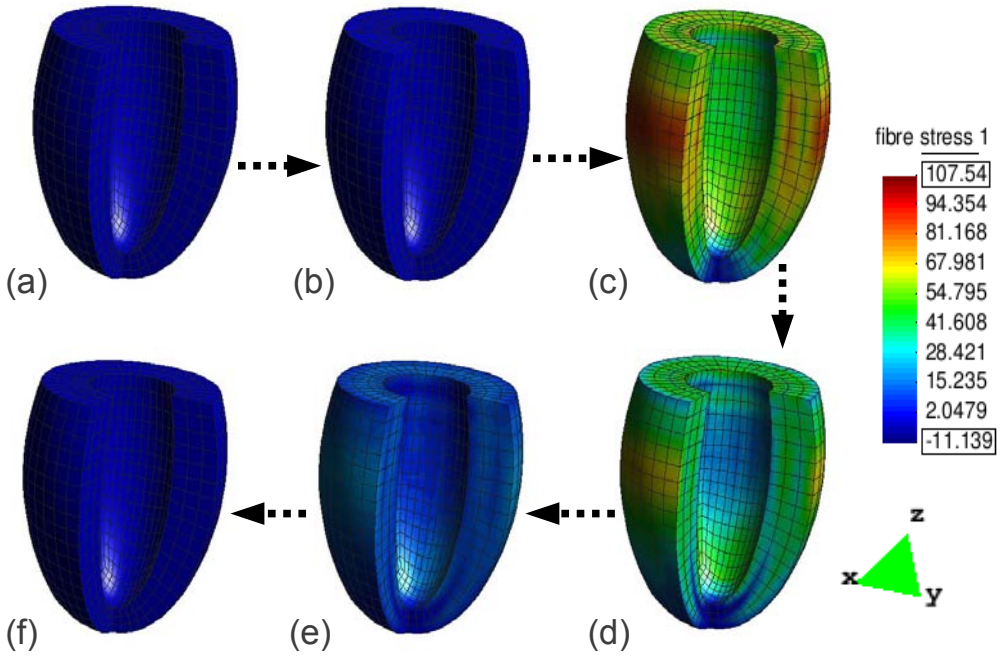

FIGURE 7.31: Stress in the fibre direction throughout the left ventricle, at representative time points in the cardiac cycle.

a few key aspects with regard to fibre stresses.

Over the coarse of the cardiac cycle, the left ventricle exhibits peak tensile fibre stresses at the epicardium, at the end of isovolumetric contraction. Furthermore, in the model study by Niederer and Smith [27], the left ventricle pressure was assumed to be unphysiologically constant, yielding a flat pressure-volume curve during this phase. The result of this assumption, was that high compressive fibre stresses were observed at the endocardium surface during ejection, due to the active stress in the ventricle. However, in our model, it appears that the coupling of the Windkessel model to the mechanics of the left ventricle during ejection, has the effect of increasing systolic-ejection pressures, which in turn reduces the compression on the endocardium surface, which would be much higher due to active tension forces, if the pressure were assumed constant [27]. Finally, at the end of the cardiac cycle, i.e., end-isovolumetric relaxation, the left ventricle has returned, almost completely, to the stress-free state at the beginning of diastolic filling, which is expected since a closed loop pressure-volume curve has been achieved.

Chapter 8

Conclusions and Recommendations

8.1 Conclusion

The conclusions section of this thesis will be split into two sections. Firstly, conclusions are made with respect to the inverse modelling applications presented in the results section. Next, conclusions are made with regards to the mechanical modelling of the left ventricular cardiac function.

8.1.1 Inverse Modelling

In this work, the projected Levenberg-Marquardt algorithm, was implemented successfully by the author, in the in-house modelling software SESKA. The algorithm was utilized to successfully calibrate active tension and passive material parameters for the myocardium of the left ventricle, as well as systemic parameters pertaining to the three element Windkessel model. The achieved set of parameters, produced a closed loop pressure-volume curve, which was consistent with the reference data adapted from Kerckhoffs et al. [17].

However, it was found that pressure-volume data during diastolic filling, did not truly capture the orthotropic behaviour of the myocardium of the left ventricle, in the fibre, sheet and sheet-normal directions. Consequently, the passive parameters in the constitutive model of Usyk et al. [28], a_1 to a_6, which govern the anisotropic behaviour of the myocardium, were neglected from parameter identification. Despite the exclusion of the parameters, a_1 to a_6, during diastolic filling, the algorithm was able to successfully fit

the pressure-volume curve of the reference data, and converged to a consistent parameter solution, based on four different initial parameter guesses. The latter point, illustrates that if the reference data utilized for calibration is appropriate to the parameters being identified, then, the algorithm is robust at obtaining a parameter solution, which produces a local minima in the cost function. Furthermore, it was illustrated that if the reference data is appropriate for the parameters that need to be identified, the damping term in the projected Levenberg-Marquardt algorithm, allows it to converge to a local minima regardless of whether the initial parameter guess is close too or far from the desired solution that minimizes the cost function.

Numerical issues associated with rounding errors in the computation of the Jacobian matrices via the finite difference scheme, prevented the author from using reference pressure-time and volume-time data to calibrate active tension and Windkessel parameters, during the phases of the cardiac cycle consisting of isovolumetric contraction, ejection and isovolumetric relaxation. However, it was illustrated that this problem can be overcome by calibrating these parameters using three specific points over the cardiac cycle, at which reference and experimental data are compared. These three points corresponded to the pressure-volume relations for simulated and reference data at end-isovolumetric contraction, peak-systolic pressure and end-ejection. The active tension and Windkessel parameters obtained using this method, produced a good fit of the systolic pressure-volume curve found in the reference data. In addition, the simulated durations of isovolumetric contraction and ejection, were close to those found in the reference data. However, due to the high degree of non-linearity in the ejection curve, it was observed that the projected Levenberg-Marquardt algorithm displayed a lack of robustness in calibrating active tension and Windkessel parameters, based on only three data points. For the algorithm to converge to a local minima, the initial parameter guess had to be very close to the optimal solution. In addition, at the obtained set of active tension, Windkessel and passive parameters, the simulated time duration of isovolumetric relaxation was greater than that reported in the reference data. The reason for this observation, was attributed to the fact that the active tension parameter, m, cannot be calibrated accurately without pressure-volume data pertaining to the isovolumetric relaxation phase.

The requirement of the Jacobian matrix computation at each major iteration of the projected Levenberg-Marquardt algorithm, resulted in calibration attempts on refined left ventricle models involving unacceptable computational times. Based on this reason, the calibration of passive, active tension and Windkessel parameters, was conducted on

a model with a relatively coarse particle distribution, which enabled more realistic calibration times, especially for the identification of active tension and Windkessel parameters, during systole. Despite the fact that cardiac output, in terms of pressure-volume relations over the cardiac cycle, changed significantly from the coarse to the refined model, a closed loop pressure-volume curve was achieved in both instances. Since, the achievement of this closed loop cardiac cycle is the objective of this work, the obtained parameter set was accepted.

8.1.2 Mechanical Modelling

Firstly, the three element Windkessel model was successfully coupled the left ventricle model, in order to describe the load faced by the latter, during ejection in pumping blood through the systemic arterial tree.

In order to conduct a mechanical analysis of the left ventricle behaviour, a refined particle distribution was required for the ellipsoid model, such that convergence of the EFGM method would be achieved. Via a convergence analysis, a refined 2884 particle model was found to be sufficient, to ensure convergence of the EFGM method. Using this refined model, and the material parameters obtained from the inverse modelling problem, a closed loop pressure-volume curve was still achieved for a single beat of a cardiac cycle. However, for this refined model, it was found that the stroke volume and peak-systolic pressures differed from those in the reference data used for calibration purposes [17].

The twist behaviour illustrated by the refined ellipsoid model, was similar to other ellipsoid model studies [17, 27]. However, this behaviour was found to be different to what is observed physiologically. In the physiological case [16], during isovolumetric contraction the left ventricle is observed to twist in a global counter-clockwise manner at all points in the body, which was observed by our left ventricle model. However, during ejection, clockwise twisting is observed near the base and counter-clockwise twisting is observed near the apex. This latter observation differs from what is observed in our model, where the twist was globally counter-clockwise throughout the ventricle, in ejection. The difference between our model and the observed physiological behaviour can be explained three-fold. Firstly, in the physiological case, the active contraction in the ventricle occurs asynchronously, which is an aspect that is not accounted for in our model. Secondly, the mechanics of the left ventricle are implicitly affected by that of the right ventricle, the latter providing increased restraint to the deformation and twist of the

former. However, in our model, the mechanics of the right ventricle have not been coupled to that of the left ventricle. Thirdly, the base is not fixed at the endocardium but only partly, i.e. elastically restrained by the major blood vessel connected to it. Next, in the physiological case, in the rapid and slow filling phases of diastolic filling, the left ventricle is reported to untwist. However, in our model, the ventricle does not untwist and instead the governing mode of deformation is outward cavity expansion, with small and almost negligible amounts of global clockwise twisting. The reason for the absence of untwisting in diastolic filling for our model, occurs as the simulation begins from the reference configuration of the left ventricle, which is in an undeformed and twist-free state.

In our left ventricle model, it was observed that the distribution of the average fibre strain between start and end-ejection throughout the ventricle, was more or less physiologically homogeneous. This observation is expected since the constants p_1 , p_2 and p_3, which are present in the computation of the helix and transverse fibre angles, have been calibrated by Rijcken et al. [13] in order to yield a near homogeneous distribution of the average fibre strain, between start and end-ejection.

8.2 Recommendations

Like the conclusions section, the recommendations made in this thesis will be split into two sections. Firstly, recommendations are made with respect to the inverse modelling applications presented in the results section. Next, recommendations will be made with regards to the mechanical modelling of the left ventricular cardiac function.

8.2.1 Inverse Modelling

Firstly, in order to calibrate the full set of parameters, given in the passive constitutive model of Usyk et al. [28], it is recommended that appropriate experimental data be utilized, that captures the anisotropic behaviour of myocardium in the fibre, sheet and sheet-normal directions. One option is to utilize displacement or strain data, based on shear testing of myocardial tissue. However, a more powerful method, that would implicitly account for cardiac output in terms of pressure-volume data, would be to calibrate passive parameters to displacement or strain data, taken from MRI images at various cross sections of the left ventricle, at representative time points during diastolic filling. Similar data, at representative time points over the remainder of the cardiac cycle, could also be utilized to calibrate for active tension and Windkessel parameters. Using this method, once passive, active tension and Windkessel parameters are identified, the

pressure-volume curves of the simulated and experimental data, should naturally coincide.

If it is chosen to calibrate the active tension and Windkessel parameters to pressure-time and volume-time data during systole and isovolumetric relaxation, it is recommended that alternate methods of computing the Jacobian matrices are considered. Such methods should be less reliant on the size of the parameter perturbations, required to capture accurate gradients of the simulated fields. Examples of such methods include automatic differentiation and the complex variable method [60], the latter being slightly more advanced in terms of numerical implementation, as the simulated fields need to be split into real and imaginary components. In addition, in order to eliminate the need to compute the Jacobian matrix at each major iteration of the algorithm, a Broyden rank 1 update scheme can be employed [58]. This method is a generalized secant procedure and requires no significant computational cost to compute and would therefore significantly reduce computational times associated with calibration [58]. However, the method is an approximate technique, which means that intermittently the Jacobian matrix will have to be evaluated using Finite differences or any other alternate procedure considered. Consequently, care needs to be taken in terms of implementation, as to when the secant procedure is applicable and when a full computation of the Jacobian matrices are required.

More heuristic, non-gradient based methods for model calibration could be considered, that completely bypass the need for accurate gradients of the simulated fields, which are clearly a drawback for gradient based methods, in terms of accuracy and computational times. Popular methods, appear to be evolutionary algorithms such as genetic search algorithms. These algorithms, although being associated with slower computational times, have the advantage of being less reliant on the starting parameter guess and are thus better means of global optimization. In addition, very noisy experimental data can have a detrimental effect on the convergence of gradient based algorithms. However, evolutionary algorithms, such as genetic search algorithms, are much more adept at handling noisy data, without significant loss of convergence.

8.2.2 Mechanical Modelling

Firstly, it is recommended that the asynchronous timing of active contraction is taken into account, either by using a bi-domain model [27] or the Eikonal diffusion equation [17]. In addition, the mechanics of the left and right ventricles should be coupled, especially since the latter provides restraint to deformation and twist of the former. In

this regard, MRI images can be used to obtain a realistic geometry of the left and right ventricles. Furthermore, for these realistic geometries, the fibre field, representing the fibre directions at all points in the body, needs to be obtained and implemented in the model [26]. Thirdly, the base at the endocardium may be modelled as being elastically restrained by the major blood vessel connected to it, rather than fixed. Consequently, taking the above considerations into account, the physiological behaviour of the left ventricle, in terms of deformation and twist maybe reproduced, especially during the systolic phase. Furthermore, in order for the model to display untwisting in diastolic filling, the following recommendations are made [27]; initially, the diastolic filling phase can be run upto a required final end-diastolic pressure. Hence, systole can be simulated, starting from the deformed geometry and pressure at the end of diastolic filling. In this case, the isovolumetric relaxation phase can be simulated and subsequently terminated at an experimentally observed final pressure, in which case it should be observed that the model is not completely untwisted. Then, diastolic filling can be repeated from the pressure and deformed configuration at end-isovolumetric relaxation, upto the same required final end-diastolic pressure. In this scenario, it should be observed that the model untwists during diastolic filling as the ventricle fills with blood. The remaining phases of the cardiac cycle can then be subsequently simulated and calibrated to achieve a closed-loop pressure-volume curve.

Bibliography

[1] ABCHealth. Heart disease, 2013. URL http://www.abc.net.au/health/library/stories/2003/02/27/1831271.htm.

[2] HumanHeartPictures. Learn basic anatomy with human heart pictures, September 2013. URL http://www.humanheartpictures.net.

[3] A.M. Katz. *Physiology of the Heart*. LIPPINCOTT WILLIAMS and WILLIAMS, 1977.

[4] faithanatomyg4. Anatomy of the muscle fiber, 2013. URL http://www.faithanatomyg4.wikispaces.com/Anatomy+of+the+Muscle+Fiber.

[5] F.B. Sachse. *Computational Cardiology*. Springer, 2004.

[6] STUDYBLUE. Managing pts with dysrhythmias, 2013. URL http://www.studyblue.com/notes/note/n/managing-pts-with-dysrhythmias-/deck/5356063.

[7] L. Robertson. The adult figure skater's guide to all things muscular, 2013. URL http://www.wikipedia.com.

[8] E.K. Rodriguez, J.H. Omens, Waldman L.K., and A.D. McCulloch. Effect of residual stress on transmural sarcomere length distribution in rat left ventricle. *Am. J. Physiol. Heart Circ. Phys.*, 264:H1048, 1993.

[9] G.A. Holzapfel and R.W. Ogden. Residual strain in rat left ventricle. *Phil. Trans. R. Soc.*, 367:1233–1250, 2012.

[10] Urgo Medical. The venous system within the cardiovascular system, 2013. URL http://www.urgomedical.com/Pathophysiologies/Compression/The-venous-system/In-the-cardiovascular-system.

[11] N. Westerhof, J. Lankhaar, and B.E. Westerhof. The arterial windkessel. *Med. Biol. Eng. Comput.*, 47:131–141, 2009.

[12] P.H.M. Bovendeerd, T. Arts, G.M. Huyghe, D.H. van Campen, and R.S. Reneman. Dependance of local ventricular wall mechanics on myocardial fibre orientation: a model study. *Journal of Biomechanics*, 25:1129–1140, 1992.

[13] J. Rijcken, P.H. Bovendeerd, A.J. Schoofs, D.H. van Campen, and T. Arts. Optimization of cardiac fibre orientation for homogeneous fibre strain during ejection. *Annual Biomedical Engineering*, 27:289–297, 1999.

[14] M.A. Branch and A. Grace. Matlab optimization toolbox user's guide. Technical report, MathWorks, Natick, MA, USA, 1996.

[15] M.K. Transtrum and J.P. Sethna. Improvements to the levenberg-marquardt algorithm for nonlinear least-squares minimization. *Preprint submitted to Journal of Computational Physics,*, January 30:1–31, 2012.

[16] G. Buckberg, J.I.E. Hoffman, A. Mahajan, S. Saleh, and C. Coghlan. Cardiac mechanics revisited : The relationship of cardiac architecture to ventricular function. *Circulation Journal of the American Heart Association*, 118:2571–2587, 2008.

[17] R. Kerckhoffs, O.P. Faris, H.M. Bovendeerd, F.W. Prinzen, K. Smits, E.R. McVeigh, and T. Arts. Timing of depolarization and contraction in the paced canine left ventricle: Model and experiment. *Journal of Cardiovasc Electrophysioly*, 14:195–198, 2003.

[18] T. Belytschko, Y.Y. Lu, and L. Gu. Element-free galerkin methods. *International Journal for Numerical Methods in Engineering*, 37:229–256, 1994.

[19] S. Avril, M. Bonnet, A.N. Bretelle, M. Grédiac, H. Hild, P. Ienny, L. Latourte, D. Lemosse, S. Pagano, E. Pagnacco, and F. Pierron. Overveiw of identification methods of mechanical parameters based on full-feild measurements. *Journal of Experimental Mechanics*, 48:402–438, 2007.

[20] L. Dubuis, S. Avril, J. Debayle, and P. Badel. Identification of material parameters of soft tissue in the compressed leg. *Computer Methods in Biomechanics and Biomedical Engineering*, 1:1–9, 2011.

[21] W. Sun, M.S. Sacks, T.L. Sellaro, W.S. Slaughter, and M.J. Scott. Biaxial mechanical response of bioprosthetic heart valve biomaterials to high in-plane hear. *Journal of Biomechanical Engineering*, 125:372–380, 2003.

[22] M.H. Kural, M. Cai, D. Tang, T. Gwyther, J. Zheng, and K.L. Billiar. Planar biaxial characterization of diseased human coronary and carotid arteries for computational modeling. *Journal of Biomechanics*, 45:790–798, 2012.

[23] H.B. Nielsen. Damping parameter in marquardt's method. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 1999.

[24] C. Kanzow, N. Yamashita, and M. Fukushima. Levenberg-marquardt methods for constrained nonlinear equations with strong local convergence properties. *Journal of Computational and Applied Mathematics*, 174:375–397, 2004.

[25] F. Guyon and R. Le Riche. Least squares parameter estimation and the levenberg-marquardt algorithm: deterministic analysis, sensitivities and numerical experiments. Technical report, Institut National des Sciences Appliquees, 2000.

[26] A.S. Niederer, G. Plank, P. Chinchapatnam, M. Ginks, P. Lamata, K.S. Rhode, A. Rinaldi, R. Razavi, and N.P. Smith. Length-dependent tension in the failing heart and the efficacy of cardiac resynchronization therapy. *Cardiovascular Research*, 89:336–343, 2011.

[27] A.S. Niederer and P.S. Smith. The role of the frank–starling law in the transduction of cellular work to whole organ pump function: A computational modeling analysis. *PLOS Computational Biology*, 5:1–18, 2009.

[28] T.P. Usyk, R. Mazhari, and A.D. McCulloch. Effect of laminar orthotropic myofiber architecture on regional stress and strain in the canine left ventricle. *Journal of Elasticity*, 61:143–164, 2000.

[29] J.M. Guccione, L.K.Waldman, and A.D. McCulloch. Mechanics of active contraction in cardiac muscle: Part i – constitutive relations for fiber stress that describe deactiveation. *Journal of Biomechanical Engineering*, 115:82–90, 1993.

[30] R.C.P. Kerckhoffs, J.C.S. Bovendeerd, F.W. Kotte, K.S. Prinzen, and T. Arts. Homogeneity of cardiac contraction despite physiological asynchrony of depolarization: A model study. *Annals of Biomedical Engineering*, 31:536–547, 2003.

[31] S. Göktepe, S.N.S. Acharya, J. Wong, and E. Kuhl. Computational modeling of passive myocardium. *International Journal for Numerical Methods in Biomedical Engineering*, 27:1–12, 2011.

[32] B.H. Dokos, S. Smail, Young A., and I.J. LeGrice. Identification of heterogeneous elastic properties in stenosed arteries: a numerical plane strain study. *Am. J. Physiol. Heart Circ. Phys.*, 283:H2650–H2659, 2002.

[33] D.R. Einstein, F. Del Pin, X. Jiao, A.P. Kuprat, J.P. Carson, K.S. Kunzelman, R.P. Cochran, J.M. Guccione, and M.B. Ratcliffe. Fluid–structure interactions of the mitral valve and left heart: comprehensive strategies, past, present and future.

International Journal for Numerical Methods in Biomedical Engineering, 26(3-4): 348–380, 2010.

[34] J.F. Wenk, C.S. Jhun, K. Sun, N. Stander, and J.M. Guccione. Determination of myocardial material properties by optimization. In *Computational Cardiovascular Mechanics*, pages 55–72. Springer, 2010.

[35] T.h. Pochet, P. Gérard, J.M. Marnette, V. D'orio, R. Marcelle, M. Fatemi, A. Fossion, and J. Juchmes. Identification of three-element windkessel model: comparison of time and frequency domain techniques. *Archives Of Physiology And Biochemistry*, 100(3):295–301, 1992.

[36] A. Cappello, G. Gnudi, and C. Lamberti. Identification of the three-element windkessel model incorporating a pressure-dependent compliance. *Annals of biomedical engineering*, 23(2):164–177, 1995.

[37] G. Mase, R. Smelser, and G. Mase. Continuum mechanics for engineers. CRC Press, 2009.

[38] V. Fuster, R. O'Rourke, R. Walsh, and P. Poole-Wilson. *Hurst's the heart*. McGraw-Hill Co., 2002.

[39] J. Vossoughi, R.N. Vaishnav, and D.J. Patel. *Advances in Bioengineering*. ed. Van C. Mow, 1980.

[40] J.H. Omens and Y.C. Fung. Residual strain in rat left ventricle. *Circ. Res.*, 66: 37–45, 2002.

[41] A. Rachev. Theoretical study of the effect of stress-dependent remodelling on arterial geometry under hypertensive conditions. *PJ. Biomech.*, 30:819–827, 1997.

[42] N. Westerhof, G. Elzinga, and P. Sipkema. An artificial system for pumping hearts. *Journal of Applied Physiology*, 31:776–781, 1971.

[43] M.F. O'Rourke. Pressure and flow waves isn systemic arteries and the anatomical design of the arterial system. *Journal of Applied Physiology*, 23:130–149, 1967.

[44] M.F. O'Rourke and A.P. Avolio. Pulsative flow and pressure in human sytemic arteries.studies in man and in a multibranched model of the human systemic arterial tree. *Circ. Research*, 46:363–372, 1980.

[45] O. Frank. Die grundform des arteriellen pulses. *Zeitung ur Biologie*, 37:483–586, 1899.

[46] N. Stergiopulos, B.E. Westerhof, and N. Westerhof. Total arterial inertance as the fourth element of the windkessel model. *American Journal of Physiology*, 276: H81–H88, 1999.

[47] M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables*, volume volume 55. Dover Publications, 1964.

[48] P.H.F. Nielsen, I.J. LeGrice, B.H. Smaill, and P.J. Hunter. Mathematical model geometry and fibrous structure of the heart. *American Journal of Physiology*, 260: 1365–1378, 1991.

[49] D.D. Streeter JR, I.J. LeGrice, B.H. Smaill, and P.J. Hunter. Handbook of physiology: a critical, comprehensive presentation of physiological knowledge and concepts. the cardiovascular system: Formerly section 2: Circulation. the heart the. 260:1–61, 1979.

[50] P. Wriggers. *Nonlinear Finite Elements*. Springer Verlag, 2008.

[51] J. Mbewu. Computational Modelling of Cardiac Function and Myocardial Infarction. 2011.

[52] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *The Quarterly of Applied Mathematics*, 9:164–168, 1944.

[53] L.A. Zadeh. Optimality and nonscalar-valued performance criteria. *IEEE Trans. Automat. Contr.*, AC-8:1–2, 1963.

[54] C. Hwang and A.S.M. Masud. Multi objective decission making, methods and application: A state of the art survey. Springer-Verlag. ISBN 978-0-387-09111-2, 2012.

[55] D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11:431–441, 1963.

[56] H. Gavin. The levenberg-marquardt method for nonlinear least squares curve fitting problems. Technical report, 2011.

[57] J.J. More. The levenberg-marquardt algorithm: implementation and theory. *Conference on Numerical Analysis,*, June 28:1–16, 1977.

[58] H.B. Nielsen. Methods for non-linear least squares problems. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 1999.

[59] A. Franquet, S. Avril, and R. Le Riche. Identification of heterogeneous elastic properties in stenosed arteries: a numerical plane strain study. *Computer Methods in Biomechanics and Biomedical Engineering*, 14:1–10, 2011.

[60] E.E. Nishida, J.T. Foster, and P.E. Briseno. Constant strain rate testing of a g10 laminate composite through optimized kolsky bar pulse-shaping techniques. *Journal of Composite Materials*, 47:2955–2963, 2013.