Effects of Inertia in Modeling of Left Ventricular Mechanics

Krister Stræte Karlsen
Master’s Thesis, Spring 2017
This master’s thesis is submitted under the master’s programme Computational Science and Engineering, with programme option Mechanics, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

The front page depicts a section of the root system of the exceptional Lie group $E_8$, projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.
Abstract

The mechanical behavior of the heart has been modeled for decades by using continuum mechanics and numerical methods. A widely used, but rarely discussed, assumption is that inertial effects can be neglected in these models. This M. Sc. project investigates the consequences of including inertial effects in modeling of left ventricular mechanics. A framework for simulating the complete cardiac cycle is implemented using the Python interface to FEniCS [10]. Numerical experiments are performed to study the role of inertia. The constitutive model for the myocardium suggested by Holzapfel and Ogden [21] will be used, along with a rule-based algorithm for assigning muscle fiber orientations to the ventricular geometry, by Bayer et al. [3]. Hozapfel and Ogden’s model assumes the myocardium to behave like an elastic material, despite strong evidence suggesting viscoelastic behavior. To see how inertial effects relate to rheology we propose a viscoelastic model for the myocardium, based on a generalization of the 1D Kelvin-Voigt material model.

Employing the elastic model [21], while including inertial effects, led to unphysiological oscillations in the ventricular walls caused by the rapidly increasing pressure load during the diastole. The wall oscillations further led to oscillations in intraventricular volume and pressure. If instead the viscoelastic model is used, the wall oscillations are damped and no oscillations in the pressure and volume curves are observed.
Acknowledgements

First and foremost I want to thank Dr. Joakim Sundnes, for invaluable constructive criticism and for always making room in his schedule for discussions throughout this M.Sc. project. Additionally, I want to thank Henrik Finsberg, Joakim’s PhD student, for insightful comments on a number of issues related to the project. I am grateful for the things I have learned from both of you. I would also like to acknowledge the excellent work environment provided by Simula, and the opportunity to attend the UCSD-UiO-Simula Summer School in biomechanics.

I want to thank, Knut Aasen, for proofreading my thesis and for being a good friend during my years at the University of Oslo. Even though his main field of interest is insurance mathematics, not mechanics, he is a good human being.

I would also like to thank my parents, Bjørn and Judith, for continuous support in everything I pursue, and my Uncle Kjell for encouraging my interest in science form the start.

Last but not least, I would like to express my sincere gratitude to the University of Oslo for five phenomenal years. Coming here, my view of higher education institutions was very old-fashioned, expecting a vast distance between the professor and the student in a strict academic hierarchy. This was, to my pleasant surprise, not the case at all. From the very beginning I felt that my opinions were heard, valued and appreciated.

Thank you.

Krister Strøte Karlsen
Oslo, May 2017
Notation

\( \mathbf{X} = (X_1, X_2, X_3) \) – reference coordinates
\( \mathbf{x} = (x_1, x_2, x_3) \) – current coordinates
\( \Omega_0 \) – reference domain
\( \Omega \) – current domain
\( \partial \Omega \) – domain boundary
\( \chi \) – motion mapping
\( \mathbf{U} \) – Lagrangian description of displacement
\( \mathbf{V} \) – Lagrangian description of velocity
\( \mathbf{A} \) – Lagrangian description of acceleration
\( \mathbf{I} \) – identity tensor
\( \mathbf{F} \) – deformation gradient
\( \mathbf{C} \) – right Cauchy–Green deformation tensor
\( \mathbf{b} \) – left Cauchy–Green deformation tensor
\( \mathbf{E} \) – Green-Lagrangian strain tensor
\( \mathbf{B} \) – volume force in the reference state
\( \mathbf{t} \) – stress vector in the current state
\( \mathbf{T} \) – stress vector in the reference state
\( \mathbf{P} \) – first Piola-Kirchhoff stress tensor
\( \mathbf{S} \) – second Piola-Kirchhoff stress tensor
\{f_0, s_0, n_0\} \quad \text{unit vectors in the fiber, sheet and normal direction in the reference configuration}

\{f, s, n\} \quad \text{unit vectors in the fiber, sheet and normal direction in the current configuration}

\rho \quad \text{density}

\eta \quad \text{weighting coefficient for transverse stress}

p \quad \text{pressure}

J \quad \text{volume ratio}

\lambda \quad \text{stretch}

\sigma \quad \text{Cauchy’s stress tensor}

\Psi \quad \text{strain-energy function}

\mathcal{T}_A \quad \text{active tension}
# Contents

1 Introduction ............................................. 11

2 Nonlinear Continuum Mechanics .................... 15
   2.1 Kinematics ........................................ 15
   2.2 Deformation ...................................... 17
       2.2.1 Change in Shape ............................. 17
       2.2.2 Change in Volume ............................ 19
   2.3 Stress ........................................... 20
   2.4 Constitutive Laws and Incompressibility ........... 21
   2.5 Lagrangian Formulation of the Equation of Motion ... 22
   2.6 A Glance at Rheology .............................. 24

3 Cardiac Mechanics ...................................... 27
   3.1 Architecture of the Left Ventricle ................ 27
   3.2 An Elastic Model for the Myocardium ............... 29
   3.3 A Viscoelastic Model for the Myocardium .......... 31
   3.4 Active Tension .................................... 32
   3.5 Modeling the Complete Cardiac Cycle ............... 33
       3.5.1 Phase 1: Ventricular Filling ................. 35
       3.5.2 Phase 2: Isovolumetric Contraction .......... 36
       3.5.3 Phase 3: Ejection ............................ 37
       3.5.4 Phase 4: Isovolumetric Relaxation .......... 38
       3.5.5 Implementation Considerations ............... 38

4 Numerical methods ..................................... 41
   4.1 The Finite Element Method ......................... 41
       4.1.1 Variational Formulation .................... 42
       4.1.2 Constructing a Computational Domain ......... 43
   4.2 Verification of Implementation .................... 45
Introduction

Cardiovascular disease is the leading cause of death in the industrialized world, responsible for more than half of overall mortality [12]. To develop better treatment methods and more precise diagnostic methods, a good understanding of the complex mechanisms of the heart is crucial. This serves as the primary motivation for pursuing cardiovascular modeling.

The heart is a muscular organ consisting of four chambers, two atria and two ventricles, see Figure 1.1. De-oxygenated blood enters the right atrium and is from there pumped into the right ventricle, and then to the lungs where carbon dioxide is released and oxygen is absorbed. The oxygenated blood then travels back to the left side of the heart into the left atrium, then into the left ventricle from where it is pumped into the aorta and arterial circulation. The left ventricle has a thick wall, made mainly of myocardium (muscle tissue), that allows it to generate high pressure during contraction. Like most studies of cardiac mechanics, the main focus of this thesis will be on the left ventricle, although in general, many of the important conclusions apply equally well to the right ventricle [6].

Ventricular pressure and volume curves have been used for a long time by clinicians as a measure of cardiac function. They give good insight into the overall pumping performance of the heart, but for pathological conditions where highly local changes may occur, such as infarction, knowledge of regional stress and strain distribution is needed to characterize ventricular function. Pressure can be measured in the subjects with a beating heart, strain and volume can be obtained from image data, stress on the other hand can not be measured in vivo and has to be computed. Thanks to rapid development of electronics and advances in numerical modeling, the most
common approach for estimating myocardial stress distributions is by using mathematical models based on the laws of continuum mechanics.

Figure 1.1: Overview of the heart anatomy. Abbreviations: RA - right atrium, LA - left atrium, RV - right ventricle and LV - left ventricle.

An important and absolutely essential aspect of any computational study of a biological system is to emphasize that "biological processes are an inseparable part of life", as Yamaguchi put it [42]. Therefore we must keep in mind that we are dealing with live, not dead, material, even when we are on a computer conducting computations. To properly model biological processes one must observe and analyze various mechanical causes and biological results, taking place in a wide range of temporal and spatial scales. This is a very complex task and even though computational modeling has been used to study the mechanics of the heart for over 20 years, some questions remain subject to debate:

- Representation of geometry and microstructure in computational models
- Rheology: Is the tissue elastic or viscoelastic?
- Constitutive relations
- Governing equations
- Boundary conditions
For these reasons we must be extremely careful in interpreting the computational results, no matter how correct they are mechanically or mathematically, to avoid making biologically pointless conclusions.

A commonly used assumption in cardiac mechanics is that the mechanical events are so slow that they can be analyzed without regard to inertial forces, or put in a more mathematical language: neglecting the acceleration term in the equation of motion. Such mechanical processes are called *quasistatic* [2]. Assuming quasistatic mechanical behavior of the heart allows for a series of static, or equilibrium, equations to be solved for time-dependent boundary conditions. If on the other hand, inertial effects are accounted for, what we here refer to as *dynamic models*, the equations require discretization in time. To the best of our knowledge, there are no reports on the consequences of including inertial effects when studying the mechanics of the heart. The primary objective of this thesis is to study the consequences of including these effects. We will also investigate how the effects of inertia relate to rheology.

In order to study the dynamic mechanical behavior of the heart we will first introduce the framework of nonlinear continuum mechanics needed to accurately model large deformations. Next the gross microstructure of the left ventricle will be presented along with the important aspects of constitutive relations, active muscle tension and the modeling of the cardiac cycle. Finally, before the results and conclusions are presented, the numerical methods used to solve the equations will be discussed and the implementation verified.

Since numerical experiments now are an integral part of the scientific method they should meet the same standards of reproducibility as natural science. To live up to these standards all scripts used to produce the results featured in the Results chapter, are made available to the reader at:

https://github.com/krikarls/master_public.git

All illustrations in this thesis are made by the author.
This chapter is a review of the general theory of nonlinear continuum mechanics and is based on the textbooks \cite{4} and \cite{22}.

One of the very first attempts to study regional stress distributions in the ventricular wall was done by Wong and Rautaharju \cite{41}. They formulated a model of the left ventricle made up of linearly elastic ellipsoidal shells of tissue and used infinitesimal strain theory to further simplify the analysis. A few years later Mirsky \cite{31} conducted a study on the effects of applying the theory valid for larger deformations. Not surprisingly, since the heart undergoes very large strains, the computed stresses were a lot higher than first predicted by Wong and Rautaharju simplified analysis. In this chapter the concepts of nonlinear continuum mechanics will be introduced so that a proper framework for modeling the mechanics of the heart behavior can be built.

2.1 Kinematics

The motion of a continuum body can be described either using Eulerian description or Lagrangian description. In Eulerian description the focus is on a point in space, i.e. how different quantities in that point change with time. In Lagrangian description we follow a material point as it moves through space and time. Since we want to study the evolution of the material
itself, and not some spatial point, a Lagrangian framework is the natural choice.

The initial state of the continuum body will be referred to either as the undeformed state or the reference configuration. In this state the body occupies the region $\Omega_0$. Later states will be referred to as deformed states or current configurations. The body is then in some region $\Omega$. The coordinate systems used for these two types of states respectively are referential coordinates, denoted by upper case letters $(X_1, X_2, X_3)$, and current coordinates for which lower case letters will be used.

Points in reference configuration are related to points in the current configuration through the motion mapping $\chi$. Given any point $X \in \Omega_0$ and some fixed time $t$,

$$x = \chi(X, t). \quad (2.1)$$

![Figure 2.1: Motion of a continuum body described using a Lagrangian framework. All quantities are expressed through referential coordinates.](image)

We will use a total Lagrangian description where all quantities are mapped back to the reference configuration, see Figure 2.1 for an illustration. From the undeformed state to some deformed state at time $t$ all points have moved

$$U(X, t) = x(X, t) - X. \quad (2.2)$$

This is the displacement field: a vector field assigning all points its new position in the deformed state.
The velocity $V(X,t)$ and acceleration $A(X,t)$ at which points move are simply the first and second derivative of the displacement with respect to time, i.e.

$$V(X,t) = \frac{\partial U(X,t)}{\partial t} \quad \text{and} \quad A(X,t) = \frac{\partial^2 U(X,t)}{\partial t^2}. \quad (2.3)$$

2.2 Deformation

Motion results in change of configuration, but not necessarily deformation. For instance, a rigid motion, i.e. a sum of an arbitrary translation and rotation, changes the body’s position, but not its shape or volume, which is how we define deformation.

2.2.1 Change in Shape

The deformation gradient

Essential to measure deformation of materials is how the points in the body move relative to each other. Consider two points initially in the undeformed state, $P, Q \in \Omega_0$, and at some later time the points are in the deformed state, $p, q \in \Omega$.

![Figure 2.2: Neighbor points under deformation.](image)

The distance between the points in the undeformed state can be expressed as

$$dX = X_Q - X_P.$$  

After the deformation the distance is

$$dx = x_q - x_p$$

$$= \chi(X_Q, t) - \chi(X_P, t)$$

$$= \chi(X_P + dX, t) - \chi(X_P, t).$$
Dividing both sides by $d\mathbf{X}$ we get the following relation

$$\frac{dx}{d\mathbf{X}} = \frac{\chi(\mathbf{X}_P + d\mathbf{X}, t) - \chi(\mathbf{X}_P, t)}{d\mathbf{X}} = \frac{\partial \chi(\mathbf{X}, t)}{\partial \mathbf{X}} = \mathbf{F}. $$

The distance between the two points is now given by $dx = \mathbf{F}d\mathbf{X}$, where $\mathbf{F}$ by definition is the deformation gradient: a tensor mapping line elements in the reference configuration into line elements in the current configuration.

By taking the gradient of (2.2) with respect to the current coordinates and reordering, the deformation gradient can be written in terms of the displacement field and the identity tensor

$$\mathbf{F} = \mathbf{I} + \nabla \mathbf{U}(\mathbf{X}, t). \quad (2.4)$$

**Stretch**

It can be useful to see how much a line element changes length during deformation, and this is indeed the case with for instance muscle fibers in the heart during contraction.

Suppose some line element has length $l_0$ in the undeformed state, and lies in direction of $\mathbf{n} \in \Omega_0$; $d\mathbf{X} = l_0 \mathbf{N}$. After deformation the length is $|dx|$, and it follows that

$$l = |dx| = |\mathbf{F}d\mathbf{X}| = l_0 |\mathbf{FN}|.$$

At a point $\mathbf{X} \in \Omega_0$ the stretch ratio, or just stretch, in the direction of $\mathbf{N}$ is defined as

$$\lambda = \frac{l}{l_0} = |\mathbf{FN}|.$$

A line element is unstretched for $\lambda = 1$, compressed for $\lambda < 1$ and extended for $\lambda > 1$.

**The Cauchy-Green deformation tensors**

Other useful measures of deformation include the Cauchy-Green deformation tensors. More more specifically they describe how the lengths and angles of
line elements change between configurations. To clarify, let’s consider the dot product of two line elements $dx_1$ and $dx_2 \in \Omega$

$$dx_1 \cdot dx_2 = FdX_1 \cdot FdX_2$$

$$= dX_1 \cdot (F^T F)dX_2$$

$$= dX_1 \cdot CdX_2.$$  

We find that $C = F^T F$, known as the right Cauchy-Green deformation tensor, relates the dot product in the deformed state to dot product in the reference configuration.

If we instead consider the dot product of the two line elements $dX_1$ and $dX_2 \in \Omega_0$

$$dX_1 \cdot dX_2 = F^{-1}dx_1 \cdot F^{-1}dx_2$$

$$= dx_1 \cdot (F^{-T}F^{-1})dx_2$$

$$= dx_1 \cdot b^{-1}dx_2$$

we obtain the left Cauchy-Green deformation tensor $b = FF^T$, relating the dot product in the reference configuration to the dot product in the current configuration.

**The Strain tensor**

Finally we will introduce the strain tensor used in nonlinear continuum mechanics. Consider the change in the squared length of a line element

$$dx^2 - dX^2 = dX \cdot CdX - dX^2$$

$$= dX \cdot (C - I)dX$$

$$= dX \cdot 2EdX.$$  

By definition $E = \frac{1}{2}(C - I)$ is the Green-Lagrange strain tensor. The factor $\frac{1}{2}$ is a normalization from linear elasticity.

**2.2.2 Change in Volume**

Before moving on to the cause of deformation, one last, but important way of describing deformation will be introduced: change in volume. We start by considering a volume element $dV$ in the undeformed state with edges given by $dX_i = dX_i i_i$, where $i_i$ are Cartesian unit vectors, $i = 1, 2, 3$. The original volume is

$$dV = dX_1 dX_2 dX_3.$$
From the following relation
\[ dx_i = F dX_i = \frac{\partial \chi}{\partial X_i} dX_i, \]
we find that the volume in the deformed state is given by
\[ dv = dx_1 \cdot (dx_2 \times dx_3) = \frac{\partial \chi}{\partial X_1} \cdot \left( \frac{\partial \chi}{\partial X_2} \times \frac{\partial \chi}{\partial X_3} \right) dX_1 dX_2 dX_3 = \text{det}(F) dX_1 dX_2 dX_3 = J \, dV. \]

Here we used that the triple product can be written as the determinant of \( F \) to find a simple expression for the volume change:
\[ J = \text{det}(F). \quad (2.5) \]

For deformation causing no change in volume \( J = 1 \).

### 2.3 Stress

Forces applied to a continuum body causes deformation. Consider an imaginary surface element within the interior of a continuum body in the deformed state, see Figure 2.3, and let the net force on that element with area \( a \) and normal \( n \) be \( f \). The stress on that element is then defined as:
\[ t = \lim_{a \to 0} \frac{f}{a}. \quad (2.6) \]

On any surface element the force acting over a unit area is known as the stress, or traction.

Having introduced stress the natural way of introducing the concept of stress tensors is through Cauchy’s stress theorem:

**Cauchy’s stress theorem.** There exists unique second-order tensor fields \( \sigma \) and \( P \) such that
\[ t = \sigma n, \quad (2.7) \]
\[ T = PN. \quad (2.8) \]
Here $\sigma$ is the Cauchy stress tensor, often called the true stress, $P$ is the first Piola-Kirchhoff stress tensor and $T$ is stress described in the reference configuration. The two stress tensors define the complete stress state in a point of body and are related through

$$\sigma = J^{-1}PF^T \quad \text{and} \quad P = J\sigma F^{-T}.$$  \hfill (2.9)

We introduce further a symmetric stress tensor $S$, the second Piola-Kirchhoff stress tensor

$$S = JF^{-1}\sigma F^{-T}.$$  \hfill (2.10)

This does not admit a physical interpretation in terms of surface tractions, like Cauchy’s stress tensor and the first Piola-Kirchhoff stress tensor. The two Piola-Kirchhoff tensors are related in the following way

$$P = FS.$$  \hfill (2.11)

### 2.4 Constitutive Laws and Incompressibility

Relating strain to stress requires constitutive laws, or models. Constitutive laws in continuum mechanics are mathematical models designed to approximate observed physical behavior of real materials. Constitutive models either state the stress-strain relationship directly through an equation, or more commonly in nonlinear continuum mechanics; through the strain-energy function.
The strain-energy function is a Helmholtz free-energy function $\Psi$ describing the stored energy (due to strain) in the material, defined per unit reference volume. An example is the St. Venant–Kirchhoff strain-energy function, based on the Green-Lagrange strain tensor

$$
\Psi = \frac{\lambda}{2} [tr(E)]^2 + \mu tr(E^2),
$$

where $\mu$ and $\lambda$ are the Lamé coefficients.

The stress-strain relationship can be obtained by differentiation of the strain-energy function

$$
S = \frac{\partial \Psi}{\partial E} = \lambda tr(E)I + 2\mu E.
$$

In the next chapter a strain-energy function suited to describe the behavior of the heart’s muscle tissue will be presented.

Most biological tissues, including the myocardium, can be regarded as an incompressible material. This has been confirmed by experiments done by Vossoughi and colleagues in [40]. A general constitutive relation for incompressible materials is obtained by adding the incompressibility constraint $J - 1 = 0$ to the the strain-energy function for the specific material $\Psi_m$

$$
\Psi = \Psi_m - p(J - 1).
$$

Here $p$ serves as an indeterminate Lagrange multiplier, which physically can be identified as the hydrostatic pressure.

### 2.5 Lagrangian Formulation of the Equation of Motion

Objects, whether it is a beam made of steel or cardiac muscle tissue, illustrated in Figure 2.4, must obey the laws of classical mechanics when subject to forces. Starting from a generalization of Newton’s second law we can now, by using the concepts introduced in this chapter, derive this thesis’ most important equation.

Newton’s second law for a continuum with volume $V$ can in Lagrangian description be stated

$$
\sum F = \frac{d}{dt} \int_V \rho \mathbf{V} \, dV.
$$
The sum of forces \( \sum \mathbf{F} \) can be divided into forces acting on the volume \( V = \Omega_0 \) and forces acting on the surface, i.e.

\[
\sum \mathbf{F} = \int_{\Omega_0} \mathbf{B} \, dV + \int_{\partial \Omega} \mathbf{T} \, dS. \tag{2.16}
\]

The surface forces can be rewritten using the stress tensor and the boundary normal

\[
\int_{\partial \Omega} \mathbf{T} \, dS = \int_{\partial \Omega} \mathbf{P} \mathbf{N} \, dS. \tag{2.17}
\]

We want the integrals to be on the same form, so by using the divergence theorem the surface integral can be converted to a volume integral

\[
\int_{\partial \Omega} \mathbf{P} \mathbf{N} \, dS = \int_{\Omega_0} \nabla \cdot \mathbf{P} \, dV. \tag{2.18}
\]

Equation (2.15) can now be expressed

\[
\int_{\Omega_0} \mathbf{B} \, dV + \int_{\Omega_0} \nabla \cdot \mathbf{P} \, dV = \frac{d}{dt} \int_{\Omega_0} \rho \mathbf{V} \, dV. \tag{2.19}
\]

Since \( \Omega_0 \) is independent of time and can be arbitrary, the integral can be omitted altogether. Finally, using relation (2.3) we arrive at Cauchy’s equation of motion in Lagrangian description

\[
\mathbf{B} + \nabla \cdot \mathbf{P} = \rho \frac{d^2 \mathbf{U}}{dt^2}. \tag{2.20}
\]

If indeed inertial effects are neglectable equation (2.20) becomes:

\[
\mathbf{B} + \nabla \cdot \mathbf{P} = 0. \tag{2.21}
\]

This is equation describes equilibrium of forces.
2.6 A Glance at Rheology

Rheology is the science of how matter flow and deform, describing the inter-relation between stress, strain and time. The term comes from Greek, *rheos* meaning *to flow*. Rheology is applicable to all materials, from gases to solids, but we will focus on viscous fluids and elastic solids, and the in-between; viscoelastic materials.

Viscoelastic materials exhibit both viscous and elastic behavior, in a sense acting a bit like a fluid and a bit like a solid. One of the principal features of elastic materials is the capacity to store mechanical energy when deformed by forces, and to release all that energy upon removal of the forces. The stresses in an elastic body are dependent on the strains in the material. On the contrary, in viscous flow mechanical energy is continuously dissipated and stresses are depending on the rate of straining. All actual materials store and dissipate energy to different degrees. This behavior is referred to as *viscoelastic*. In general, viscoelastic behavior may be imagined as a spectrum with elastic deformation as one limiting case and viscous flow the other extreme case, with varying combinations of the two spread over the range between. Therefore valid constitutive equations for viscoelastic materials embody purely elastic and viscous behavior as special cases. Such equations will in addition to stress and strain, involve the rates of change of stress and strain [30].

![Figure 2.5: Schematic representation of Kelvin–Voigt model.](image)

The traditional way of introducing viscoelastic models is by consider coupling of springs and dash-pots. The spring represents the elastic behavior; it undergoes instantaneous straining upon loading and immediately start contracting when the load is removed. The dash-pot is a piston inside a fluid-filled cylinder where the viscous response is achieved by pulling the piston through fluid.
One of the classical viscoelastic models is called *Kelvin–Voigt*, named after the British physicist and engineer Lord Kelvin and after German physicist Woldemar Voigt. It consists of a spring and dash-pot in parallel, see Figure 2.5, such that the total stress $\sigma$ is a sum of an elastic stress $\sigma_e$ and a viscous stress $\sigma_v$ [18]

$$\sigma = \sigma_e + \sigma_v.$$  

If a load is suddenly applied to a material following the Kelvin–Voigt model, it immediately responds like a viscous material, due to zero initial strain. After some straining has occurred, the elastic stress becomes non-zero and the material shows both viscous and elastic behavior.

Most models for the heart’s muscle tissue assume elastic material behavior, despite strong evidence from experimental data showing rate-dependent behavior, as mentioned by Cansiz and his colleagues [7]. In Section 3.3 we propose a viscoelastic model for the muscle tissue based on the Kelvin–Voigt model.
2.6. A GLANCE AT RHEOLOGY
Cardiac Mechanics

The field of cardiac mechanics has come a long way since the work of Wong and Rautaharju. Thanks to advances computational science, stresses and strains in the heart can now be studied in detail. With these advances understanding, and being able to represent in computational models, the microstructure of the cardiac muscle tissue has become increasingly important.

3.1 Architecture of the Left Ventricle

A healthy left ventricle is shaped like a truncated ellipsoid with long axis approximately twice the length of the short axis and wall thickness about 1 cm [38]. The ventricular wall consists of three layers, from the innermost to the outermost: endocardium, myocardium and epicardium [23], see Figure 1.1. Myocardium is the muscle tissue of the heart and forms the predominant part of the ventricle wall. It is composed of helical networks of muscle fibers made of myocytes. Myocyte is a type of muscle cell shaped like a cylinder with length 80 to 100 µm and radius 5 to 10 µm [34]. Locally the fibers are approximately parallel, thus defining a local mean fiber direction. The ventricular mechanics depends strongly on the directions of the fibers, thus being able to accurately represent how they are oriented in a computational model is important. Common approaches for assigning fiber orientations to computational models include image-based and rule-based methods. Even though image-based methods are generally considered to give more accurate representation of fiber alignment, the results are often negatively effected by
noisy image data. Since high quality image data are not always available, rule-based methods serve as a good alternative. Rule-based methods are algorithms assigning fiber orientations based on a set of rules. Examples of rules can be:

(i) The fibers in the ventricular walls are parallel to the endocardial and epicardial surfaces.

(ii) Orientation of the fiber rotate clockwise throughout the ventricular walls from the endocardium with an angle $+\alpha$ to the epicardium $-\alpha$.

(iii) The sheet normal is orthonormal to the longitudinal and transverse fiber directions.

We will use a rule-based algorithm suggested by Bayer et al [3] for assigning the fiber directions to our computational model. The rule examples listed above are taken from this algorithm, for the full set of rules see [3].

![Sketch of the left ventricle](image)

Figure 3.1: Sketch of the left ventricle(a) indicating the layering of sheets across the ventricular wall(b) and showing the local coordinate system on a small sheet element(c) in the reference configuration. $f_0$, $s_0$ and $n_0$ are mutually orthogonal vectors in fiber, sheet and normal direction.

Mathematically, the fiber-sheet structure can be characterized by a orthonormal set of basis vectors, $\{f_0, s_0, n_0\}$ associated with the reference coordinates $(X_1, X_2, X_3)$. The directions represent respectively the alignment of the fibers and two mutually orthogonal directions, one in the sheet plane and one normal to the sheet, Figure 3.1.
3.2 An Elastic Model for the Myocardium

Passive cardiac muscle tissue exhibits most of the mechanical characteristics of soft biological tissue in general [14], and several constitutive models for the myocardium have been proposed. The vast majority are varieties of elastic models, ranging from simple isotropic to more complex orthotropic ones. For a short review of models for the elasticity of the myocardium the reader is referred to [21], and for a more extensive one see [16].

One of the major challenges in developing good mechanical models for the myocardium is the lack of experimental data suitable for detailed parameter estimation. This is essential to accurately predict myocardial tissue response during the cardiac cycle. The most comprehensive study is based on multi- and biaxial tests of cardiac tissue mainly from dogs [19, 8]. The experiments showed that the passive myocardium exhibits significant anisotropy with substantially greater stiffness in the muscle fiber direction. This fibrous architecture of the myocardium has served as motivation for transversely isotropic models.

However, in more recent work done by Dokos et al. [9], on the basis of shear tests in different directions done on cube-shaped specimens, orthotropic behavior was observed. Based on these observations Holzapfel and Ogden proposed an orthotropic model for the mechanical response of the passive myocardium [21]. Despite the observed orthotropic behavior, we here employ a transversely isotropic version of their model. The model is based on one isotropic and one transversely isotropic invariant

$$I_1 = tr(C) \quad \text{and} \quad I_{4f} = f_0 \cdot (C_{f0}),$$

(3.1)
and is given by the strain-energy function

$$\Psi = \frac{a}{2b} [\exp(b(I_1 - 3)) - 1] + \frac{af}{2bf} [\exp(b_f(I_{4f} - 1)^2) - 1].$$  \hfill (3.2)
3.3 A Viscoelastic Model for the Myocardium

Despite the fact that the myocardial tissue is usually treated as an elastic material, there are studies reporting viscoelastic behavior. In recent experiments conducted on human myocardium, done by Sommer and colleagues [35], viscoelastic response was recovered in both biaxial and shear tests. Some suggest that a possible reason for viscoelastic behavior is the extracellular fluid filtrating through the myocardium [39, 43].

We start out by assuming that the viscoelastic behavior myocardium can be represented by a Kelvin-Voigt model, thus the total stress can be expressed as a sum of an elastic and a viscous contribution

\[ \sigma = \sigma_e + \sigma_v. \] (3.4)

Furthermore we assume that the elastic stress \( \sigma_e \) is given by Holzapfel and Ogden’s model (3.3), and that the viscous stress depends on time rates of \( \mathbf{C} \) and \( \mathbf{b} \). To obtain these time rates we need to start with the rate of change of the deformation gradient

\[ \dot{\mathbf{F}} = \nabla \mathbf{V}(\mathbf{X}, t). \]

Next, the velocity gradient in the current configuration can be expressed through \( \dot{\mathbf{F}} \)

\[ \nabla \mathbf{v}(\mathbf{x}, t) = \dot{\mathbf{F}} \mathbf{F}^{-1}. \]

Decomposition of \( \nabla \mathbf{v}(\mathbf{x}, t) \) into a symmetric part and a skew-symmetric part yields

\[ \nabla \mathbf{v}(\mathbf{x}, t) = \mathbf{d} + \mathbf{w} = \frac{1}{2}(\nabla \mathbf{v} + \nabla \mathbf{v}^T) + \frac{1}{2}(\nabla \mathbf{v} - \nabla \mathbf{v}^T), \]

where \( \mathbf{d} \) is the rate of deformation tensor and \( \mathbf{w} \) is the rate of rotation tensor. Now the time rates of the Cauchy-Green deformation tensors can be written as

\[ \dot{\mathbf{C}} = 2\mathbf{F}^T \mathbf{dF} \quad \text{and} \quad \dot{\mathbf{b}} = (\nabla \mathbf{v}) \mathbf{b} + \mathbf{b}(\nabla \mathbf{v})^T. \]

Finally, assuming that the viscous stress is isotropic and of a form similar to the elastic stress, we propose the following

\[ \sigma_v = \alpha \exp(\beta \dot{I}_1) \dot{\mathbf{b}}. \] (3.5)

Here \( \dot{I}_1 = tr(\dot{\mathbf{C}}) \), and \( \alpha \) and \( \beta \) are model parameters, see Appendix B.
3.4 Active Tension

Unlike most materials studied in solid mechanics biological tissue can deform without the presence of external loads, due to the activity of the muscle cells. This is often referred to as the *active* part of the deformation. There are two proposed ways to incorporate this in the framework of continuum mechanics. The most commonly used model is called *active stress*. In this model the total stress is assumed to be a sum of the active and the passive stress, i.e. $\sigma = \sigma_a + \sigma_p$. The other strategy is called *active strain* [37], and assumes the total deformation to be a product of a active and passive contribution, i.e. $F = F_p F_a$. In this thesis the active stress approach will be used. For a discussion on mathematical and biological properties of the two models we refer the reader to [1].

![Graph](image)

Figure 3.4: Development of active tension $T_A(t)$ over a cardiac cycle.

A model for the development of active tension in the muscle fibers, based on the concentration of free intracellular calcium $[Ca^{2+}]$ and stretching of the fibers $\lambda_f$, was suggested by Hunter, Nash and Sands [24] in 1997 and is still used today. Even more sophisticated models include the active stress to be modified by shortening or lengthening according to force–velocity relations, fiber tension is then reduced by increased shortening velocity. There are also fully history-dependent models that are more complex, based on cross-bridge theory [11]. Here we will take the liberty of assuming the active tension is only a function of time, where its shape is similar to the calcium transient, see Figure 3.4.
It is reasonable to expect that most of the active forces will be in the direction of the fibers, although results in [28] indicate a significant amount in the transverse directions. To incorporate this in a simple and flexible way we define the active part of the Cauchy stress as

$$\sigma_a = T_A(t)(f \otimes f) + \eta T_A(t)(s \otimes s + n \otimes n).$$

Here $\eta$ is a nondimensional weighting factor for the active tension in the transverse directions. The total First Piola-Kirchhoff stress thus becomes

$$P = J(\sigma_p + \sigma_a)F^{-T},$$

where $\sigma_p$ is given by (3.3). Research done on mice by Karlon et al. [25] shows fiber angle dispersion around 10-15 % about the mean fiber direction $f$. We assume the active stress in transverse directions to be 10 % of the stress in the fiber direction, i.e. $\eta = 0.1$.

### 3.5 Modeling the Complete Cardiac Cycle

Knowing the sequence of mechanical events during the complete cardiac cycle is important to understand how cardiac function is regulated. The cardiac cycle can be divided into two general categories: systole and diastole. Systole refers to events associated with ventricular contraction and ejection. Diastole refers to the rest of the cardiac cycle, including ventricular filling and relaxation. We will further divide the cardiac cycle into four phases: 1) ventricular filling, 2) isovolumetric contraction, 3) ejection and 4) isovolumetric relaxation. These events can be summarized in a cardiac cycle diagram showing intraventricular pressure and volume curves, see Figure 3.5.

The domain representing the left ventricle is denoted $\Omega$ with boundary $\partial \Omega$. The boundary $\partial \Omega$ is divided into three subdomains $\partial \Omega = \partial \Omega_{base} \cup \partial \Omega_{endo} \cup \partial \Omega_{epi}$, representing the base, endocardium and epicardium respectively.

We assume the base of the ventricle to be rigid and neglect body forces, as done in [36]. The outer surface is under a constant zero stress, and to mimic pressure from the blood a time-dependent load $\sigma(t) = p(t)I$ is applied to the interior wall. This is a quantity in the deformed state and must be mapped back to the reference coordinates. Using (2.9) the traction boundary condition becomes:

$$T = J(p(t)I)F^{-T}N = p(t)F^{-T}N.$$
3.5. MODELING THE COMPLETE CARDIAC CYCLE

Figure 3.5: Cardiac cycle diagram for the left ventricle showing the different phases (numbers) and their characterizing events (letters): mitral valve opening (a) and closing (b), aortic valve opening (c) and closing (d).

At the beginning of the cardiac cycle the ventricle is considered to be at rest in its equilibrium position, i.e. $U(t = 0) = 0$ and $\frac{\partial U}{\partial t}(t = 0) = 0$. The boundary value problems (BVP) for both models are stated below.

### Quasistatic BVP

\[
\nabla \cdot \mathbf{P} = 0 \quad \text{in} \quad \Omega_0 \quad (3.9) \\
\mathbf{U} = 0 \quad \text{on} \quad \partial \Omega_{\text{base}} \quad (3.10) \\
\mathbf{T} = p(t) \mathbf{F}^{-T} \mathbf{N} \quad \text{on} \quad \partial \Omega_{\text{endo}} \quad (3.11) \\
\mathbf{T} = 0 \quad \text{on} \quad \partial \Omega_{\text{epi}} \quad (3.12)
\]
CHAPTER 3. CARDIAC MECHANICS

Dynamic BVP

\[ \nabla \cdot \mathbf{P} = \rho \frac{\partial^2 \mathbf{U}}{\partial t^2} \quad \text{in} \quad \Omega_0 \quad (3.13) \]

\[ \mathbf{U} = 0 \quad \text{on} \quad \partial\Omega_{\text{base}} \quad (3.14) \]

\[ \mathbf{T} = p(t) \mathbf{F}^{-1} \mathbf{N} \quad \text{on} \quad \partial\Omega_{\text{endo}} \quad (3.15) \]

\[ \mathbf{T} = 0 \quad \text{on} \quad \partial\Omega_{\text{epi}} \quad (3.16) \]

Having established the mathematical problem, the different phases of the cardiac cycle can be modeled as follows.

3.5.1 Phase 1: Ventricular Filling

The filling phase is initiated by the mitral valve opening. At this point the atria are maximally filled so blood flow into the ventricle is rapid, and causes the intraventricular pressure to increase approximately linearly [26]. It takes about 400-500 ms before the left ventricle is filled and the mitral valve closes again. At the end of this phase the volume is at its maximum value around 100-120 ml. This is referred to as the end-diastolic volume \( V_{ED} \) and is associated with the end-diastolic pressure \( p_{ED} \), which is usually about 8-12 mmHg or around 1.0-1.5 kPa, see Figure 3.5.

![Figure 3.6: Intraventricular pressure \( p(t) \) used to simulate the first phase of the cardiac cycle.](image)

\[ \text{Figure 3.6: Intraventricular pressure } p(t) \text{ used to simulate the first phase of the cardiac cycle.} \]
Since $p(t)$ is a known function in this phase, see Figure 3.6, the boundary value problems introduced in the previous section can be solved directly using numerical methods. This is discussed in more detail in the next chapter.

### 3.5.2 Phase 2: Isovolumetric Contraction

This phase is initiated by depolarization increasing intracellular calcium, causing ventricular contraction, i.e. increasing active tension in the muscle fibers, see Figure 3.4 around 500 ms. Increasing active tension causes intraventricular pressure to rise rapidly, which closes the valves. With all the valves closed the ventricular volume stays constant, therefore the contraction is said to be isovolumetric. Even though the volume is constant in this phase, the ventricle undergoes significant change in geometry: from having an ellipsoidal shape to becoming more sphere-like, see Figure 3.7. This change is due to the helical alignment of fibers. As the contraction continues without ejection the pressure continues to increase rapidly throughout this phase until the aortic valve opens, defining the end of phase 2.

![Figure 3.7: Due to the helical alignment of muscle fibers ventricular geometry changes considerably during the isovolumetric contraction: from having an ellipsoidal shape(left) to becoming more sphere-like(right).](image)

From a viewpoint of mechanics, this phase is a struggle between the active tension and the intraventricular pressure to keep the volume $V(p(t), T_A(t))$ constant and equal to the end-diastolic volume $V_{ED}$. Since the active tension is a known function $T_A(t)$, the boundary condition $p(t)$ can be determined from the following equation

$$f(p) = V(p(t), T_A(t)) - V_{ED} = 0.$$  

(3.17)
This equation can be solved for the unknown pressure with Newton’s method:

\[ p_{k+1} = p_k - \frac{f(p_k)}{f'(p_k)}, \quad (3.18) \]

using a finite difference approximation for the derivative:

\[ f'(p_k) = \frac{dV}{dp} \approx \frac{V(p_k + \Delta p, T_A(t)) - V(p_k, T_A(t))}{\Delta p}. \quad (3.19) \]

For each iteration both \( V(p_k + \Delta p, T_A(t)) \) and \( V(p_k, T_A(t)) \) has to be computed, which requires the boundary value problem to be solved twice for each iteration, making it a computationally expensive procedure.

### 3.5.3 Phase 3: Ejection

When pressure in the ventricle exceeds the pressure in aorta \( p_a \), the aortic valve opens and blood is ejected out of the ventricle. Ejection can be modeled using the same idea as for the isovolumetric contraction. Instead of forcing the volume to be constant, we force it to change according to a model describing the outflow of blood as a function of pressure and volume:

\[ \frac{dV}{dt} = -C_{art} \frac{dp}{dt} - \frac{p - P_0}{R_{per}}. \quad (3.20) \]

This is a two-element Windkessel model and was originally formulated by Otto Frank [13]. The parameters \( C_{art}, R_{per} \) and \( P_0 \) can be found in Appendix B.

Discretization in time, and by introducing the notation \( V_n = V(p(t_n), T_A(t_n)) \), (3.20) becomes

\[ \frac{V_n - V_{n-1}}{\Delta t} = -C_{art} \frac{p_n - p_n-1}{\Delta t} - \frac{p_n - P_0}{R_{per}}. \quad (3.21) \]

Formulated as a function of the unknown pressure

\[ g(p_k) = \frac{V_n - V_{n-1}}{\Delta t} + C_{art} \frac{p_k - p_{n-1}}{\Delta t} + \frac{p_k - P_0}{R_{per}} = 0. \quad (3.22) \]

Since \( V_n \) is a function of \( p_k \) the derivative becomes

\[ g'(p_k) = \frac{1}{\Delta t} \frac{dV_n}{dp} + C_{art} \frac{1}{\Delta t} + \frac{1}{R_{per}} \quad (3.23) \]

where \( dV/dp \) is approximated by (3.19). The unknown pressure \( p_n \) can now be determined by inserting (3.22) and (3.23) into Newton’s method (3.18).
3.5.4 Phase 4: Isovolumetric Relaxation

As blood flow out of the ventricle slows down, and the active tension falls, at some point the pressure gradients across the valves reverse, causing the valves to close. Ventricular volume is therefore constant during this phase until the ventricular pressure once again is lower than the atrial pressure and the cardiac cycle starts all over. The blood volume that is left after the ejection is called the end-systolic volume \( V_{ES} \) and is usually about 40-50 ml. The difference between the end-diastolic volume and the end-systolic referred to as the stroke volume. The ejected volume of blood divided by \( V_{ED} \) is called the ejection fraction. These quantities are commonly used by clinicians as simple measures of cardiac function. Due to the relaxation of the muscle fibers in this phase, the geometry change featured in Figure 3.7 happens in the reversed order.

The isovolumetric relaxation is modeled exactly the same way as isovolumetric, except that the end-systolic volume is now the constraint

\[
f(p) = V(p(t), T_A(t)) - V_{ES} = 0. \tag{3.24}
\]

3.5.5 Implementation Considerations

Electrical activation causes the heart to contract, leading to pressure gradients across the valves. It is the pressure gradients that control the opening and closing of valves, thus the phase transitions. Since our model of the ventricle is not coupled to a complete model for to the circulatory system, the phase transitions need to be implemented in a more ad hoc manner. We need to choose some criteria that defines the end of a phase and the end of a cycle, such that multiple cardiac cycles can be simulated easily. One must keep in mind that the criteria we choose should in a good way reflect the actual events.

We start by defining a "phase variable" phase, keeping track of which phase is currently active so that the right set of boundary conditions is used. Next a "cycle counter" cycle and a maximum number of cycles cycles are defined. By the use of logical statements the phase and cycle variables are updated based on the decided criteria. The complete implementation of phase transitions can be summed up in Algorithm 1. The function \( \text{solve}(p(t), T_A(t)) \), solves the boundary value problems stated in the beginning of this section given the active tension and the intraventricular pressure as input values.
Computing the internal volume of the deformed ventricle is not entirely trivial as it lies outside of the computational domain. A walk-through of how it can be done is found in Appendix A, along with code examples of the algorithms for determining the unknown pressures in phase 2-4.
Algorithm 1: Phase switching

phase = 1

cycle = 1
cycles = n

while cycle ≤ cycles do

"determine phase"
if phase = 1 and $T_A(t) > \epsilon_1$ then
  phase = 2
if phase = 2 and $p > p_a$ then
  phase = 3
if phase = 3 and $V < V_{ES} + \epsilon_2$ then
  phase = 4

"solve BVP with right BCs"
if phase = 1 then
  solve($p(t), T_A(t)$)
if phase = 2 then
  determine $p(t)$ from (3.17)
solve($p(t), T_A(t)$)
if phase = 3 then
  determine $p(t)$ from (3.22)
solve($p(t), T_A(t)$)
if phase = 4 then
  determine $p(t)$ from (3.24)
solve($p(t), T_A(t)$)

"restart cycle"
if $t = 0.9$ then
  phase = 1
  cycle = cycle + 1
Numerical methods

Historically, numerical methods have been considered the third avenue of the sciences, but this approach is now one of the most important, paralleling experimental and theoretical methods. It has for a long time been regarded as a \textit{sine qua non} tool in engineering. Although its use in the fields of medicine has a shorter history, it is now spreading like wildfire to many areas of medical practice.

4.1 The Finite Element Method

\textit{The finite element method} (FEM) is a numerical method widely used to solve mechanics problems governed by partial differential equations. One of its attractive features is the ease of handling complex geometries, making it an excellent tool for studying the mechanics of the heart, and it has been used for that purpose for over 20 years. FEM is built on a rigorous mathematical framework based on functional analysis. We will not go into the mathematical details of FEM, but for a thorough introduction to the subject the reader is referred to [5], and for a more general review of FEM with applications see [33].

In this section, two of the main ingredients needed for solving partial differential equations using the finite element method will be introduced; variational formulation and constructing a computational domain.
4.1. THE FINITE ELEMENT METHOD

4.1.1 Variational Formulation

The primary motivation for using numerical methods is that analytical solutions to nonlinear initial boundary value problems, like the ones introduced in the previous chapter, only exist for some very special cases. Thus strategies for obtaining approximate solutions, like finite difference methods and variational principles, are needed. The finite element method belongs to the second category.

To obtain a variational formulation of the quasistatic boundary value problem (3.9)-(3.12), we start by multiplying the equation of motion by a vector-valued function weight function \( \delta u \) allowed to vary arbitrarily. The same is done with incompressibility constraint for a scalar weight function \( \delta p \). Integrating over the domain, using integration by parts, the divergence theorem and the boundary conditions, the solution to the quasistatic BVP is characterized by

\[
\left( U, p \right) \in \mathcal{U} \times \mathcal{P}:
\]

\[
\int_{\Omega} P : \nabla \delta u \, dV - \int_{\text{endo}} \mathbf{T} \cdot \delta u \, dS = 0 \quad \forall \, \delta u \in \mathcal{U},
\]

\[
\int_{\Omega} (J - 1) \delta p \, dV = 0 \quad \forall \, \delta p \in \mathcal{P}.
\]

(4.1)

(4.2)

The dynamic problem (3.13)-(3.16), can be restated as a system of two first-order equations. In a similar manner, as for the quasistatic problem, we multiply by weight functions \( \delta u \), \( \delta v \) and \( \delta p \) to obtain the variational formulation

\[
\left( U, V, p \right) \in \mathcal{U} \times \mathcal{V} \times \mathcal{P}:
\]

\[
\int_{\Omega} P : \nabla \delta u + \rho \frac{dV}{dt} \cdot \delta u \, dV - \int_{\text{endo}} \mathbf{T} \cdot \delta u \, dS = 0 \quad \forall \, \delta u \in \mathcal{U},
\]

\[
\int_{\Omega} \left( \mathbf{V} - \frac{\partial \mathbf{U}}{\partial t} \right) \cdot \delta \mathbf{v} \, dV = 0 \quad \forall \, \delta \mathbf{v} \in \mathcal{V},
\]

\[
\int_{\Omega} (J - 1) \delta p \, dV = 0 \quad \forall \, \delta p \in \mathcal{P}.
\]

(4.3)

(4.4)

(4.5)

Here \( \mathcal{U} \), \( \mathcal{V} \) and \( \mathcal{P} \) are appropriate function spaces satisfying \( \delta u = 0 \) for \( X \in \partial \Omega_{\text{base}} \).
All weight functions are defined on the reference configuration and have only spatial dependence. We could have used weighting functions in the time domain as well, but it is common in solid mechanics to use finite difference methods in time.

Letting the subscript \(n\) denote some quantity at time \(t_n\), the variational formulation of the quasistatic problem at time \(t_n\) reads

\[
(U, p) \in \mathcal{U} \times \mathcal{P}:
\int_{\Omega} \mathbf{P}_n : \nabla \delta \mathbf{u} \, dV - \int_{\text{endo}} \mathbf{T}_n \cdot \delta \mathbf{u} \, dS = 0 \quad \forall \ \delta \mathbf{u} \in \mathcal{U},
\int_{\Omega} (J - 1) \delta p \, dV = 0 \quad \forall \ \delta p \in \mathcal{P}.
\]

Using backward finite differences to approximate the time-derivatives, the variational formulation for the dynamic problem at \(t_n\) can be expressed

\[
(U, V, p) \in \mathcal{U} \times \mathcal{V} \times \mathcal{P}:
\int_{\Omega} \mathbf{P}_n : \nabla \delta \mathbf{u} + \rho \frac{\mathbf{V}_n - \mathbf{V}_{n-1}}{\Delta t} \cdot \delta \mathbf{u} \, dV - \int_{\text{endo}} \mathbf{T}_n \cdot \delta \mathbf{u} \, dS = 0 \quad \forall \ \delta \mathbf{u} \in \mathcal{U},
\int_{\Omega} \left( \mathbf{V}_n - \frac{\mathbf{U}_n - \mathbf{U}_{n-1}}{\Delta t} \right) \cdot \delta \mathbf{v} \, dV = 0 \quad \forall \ \delta \mathbf{v} \in \mathcal{V},
\int_{\Omega} (J - 1) \delta p \, dV = 0 \quad \forall \ \delta p \in \mathcal{P}.
\]

Since \(u'(t_n) = ((u(t_n) - u(t_{n-1}))/\Delta t + \mathcal{O}(\Delta t)\) we can expect an error of order \(\mathcal{O}(\Delta t)\) due to time discretization.

### 4.1.2 Constructing a Computational Domain

Continuum mechanics problems are continuous, and to be able to apply our numerical tools we need something discrete and finite. The domain \(\Omega\) is therefore partitioned into a finite number of smaller and simpler geometric entities, typically triangles in 2D and tetrahedra in 3D, called elements: \(\Omega^{(k)}, k = 1, 2, 3, ..., N < \infty\), see Figure 4.1. The set of all these elements is called the computational domain, or mesh. The domain can then be represented as

\[
\Omega = \Omega^{(1)} \cup \Omega^{(2)} \cup .. \cup \Omega^{(N)}.
\]
4.1. THE FINITE ELEMENT METHOD

In each of these elements, the solution is approximated by basis functions, often piecewise Lagrange polynomials, to obtain simple equations for each element. The equations obtained for each element are then assembled together with adjoining elements to form the global finite element equation for the whole problem domain.

![Mesh of the left ventricle made of tetrahedra. Dimensions: $a = 2.0 \text{ cm}$, $c = 3.0 \text{ cm}$, $d = 1.3 \text{ cm}$ and $e = 1.0 \text{ cm}$. Volume(internal volume of the small ellipsoid with axes $a$ and $c$): 50 ml. Number of elements: 6395.](image)

Figure 4.1: Mesh of the left ventricle made of tetrahedra. Dimensions: $a = 2.0 \text{ cm}$, $c = 3.0 \text{ cm}$, $d = 1.3 \text{ cm}$ and $e = 1.0 \text{ cm}$. Volume(internal volume of the small ellipsoid with axes $a$ and $c$): 50 ml. Number of elements: 6395.

There are numerous tools available for creating finite element meshes of the heart from image data, but for the purpose of comparing different mechanics-based models we think a more idealized geometry will be sufficiently accurate. More specifically, our model of the left ventricle will take the shape of a thick-walled, truncated, axisymmetric ellipsoid, and is created with gmsh [15], see Figure 4.1. The mesh consists of tetrahedra, and to approximate the solution piecewise Lagrange polynomials will be used.

The variational problems presented in abstract form in Section 4.1.1 are solved using FEniCS, which by default uses Newton’s method for nonlinear variational problems. FEniCS is a collection of free, open source, software components for solving for partial differential equations that can be used with high-level scripting in Python. It is very user-friendly with a syntax that closely mimics the mathematics. Therefore mathematical models for physical problems can be quickly translated into finite element code. FEniCS has support for parallel computing using MPI and can run on both laptops and high-performance computer clusters. For more information see [29] and [https://fenicsproject.org/](https://fenicsproject.org/)

To see the full implementation using the Python interface to FEniCS, we refer the reader to the github repository from the introduction.
4.2 Verification of Implementation

For computational modeling to contribute to advances in natural science, results from simulations should not be trusted blindly, but must be verified and validated before conclusions are drawn. Verification and validation is the process of answering the two questions: "are we solving the equations right?" and "are we solving the right equations?". We will concentrate on the first question, although they should be considered equally important. To verify our model implementation, numerical results will be compared with two exact solutions. One of physical relevance obtained under simplifying assumptions, and one manufactured solution with the sole purpose of verifying the code.

4.2.1 Analytical Solution: Uniform Deformation of a Cube of Tissue

We will now find an analytical solution by considering uniform deformation in a cube of tissue caused by active tension. Uniform deformation is defined by $x_i = \lambda_i X_i$. For a cube with fibers aligned along the $X_1$-axes and with the transversely isotropic model (3.2) the deformation can be expressed as

$$ x_1 = \lambda_f X_1, \quad x_2 = \lambda_t X_2 \quad \text{and} \quad x_3 = \lambda_t X_3, \quad (4.6) $$

where $\lambda_f$ is the stretch in the fiber direction and $\lambda_t$ is the stretch in the transverse direction.

Thus the deformation gradient becomes

$$ F = \begin{pmatrix} \lambda_f & 0 & 0 \\ 0 & \lambda_t & 0 \\ 0 & 0 & \lambda_t \end{pmatrix}. \quad (4.7) $$

Furthermore, since $F = F^T$, we have the Right and left Cauchy–Green tensors

$$ C = b = \begin{pmatrix} \lambda_f^2 & 0 & 0 \\ 0 & \lambda_t^2 & 0 \\ 0 & 0 & \lambda_t^2 \end{pmatrix} \quad (4.8) $$

and the invariants

$$ I_1 = \lambda_f^2 + 2\lambda_t^2 \quad \text{and} \quad I_{4f} = \lambda_f^2. \quad (4.9) $$
4.2. VERIFICATION OF IMPLEMENTATION

Inserted into the Cauchy stress (3.3) we get

\[
\sigma = a \exp(b(\lambda_f^2 + 2\lambda_t^2 - 3))b \\
+ 2a_f(\lambda_f^2 - 1)\exp(b_f(\lambda_f^2 - 1)^2)(f \otimes f) - pI,
\]

here \( f = Ff_0 = (\lambda_f, 0, 0)^T \).

Since the matrices are diagonal and the material incompressible \( J = 1 \), the mapping of the stress back to the reference configuration takes the simple form

\[
S_{ij} = F_{ij}^{-1}\sigma_{ij}F_{ij}^{-T}.
\] (4.10)

The complete stress state then reads

\[
S_{11} = a \exp(b(\lambda_f^2 + 2\lambda_t^2 - 3)) \\
+ 2a_f(\lambda_f^2 - 1)\exp(b_f(\lambda_f^2 - 1)^2) - \frac{p}{\lambda_f^2} 
\] (4.11)

\[
S_{22} = S_{33} = 2a_f(\lambda_f^2 - 1)\exp(b_f(\lambda_f^2 - 1)^2) - \frac{p}{\lambda_t^2} 
\] (4.12)

\[
S_{12} = S_{13} = S_{21} = S_{23} = S_{31} = S_{32} = 0 
\] (4.13)

In the absence of external forces the passive and the active stress must balance in all points and therefore the equation of motion may be reduced to

\[
S_{\text{passive}} + S_{\text{active}} = 0.
\]

Inserting the active stress (3.6) and equations (4.11)-(4.13) we get the following system of nonlinear equations

\[
S_{11}(\lambda_f, \lambda_t, p) + T_A(t) = 0 
\] (4.14)

\[
S_{22}(\lambda_f, \lambda_t, p) + \eta T_A(t) = 0. 
\] (4.15)

Using the incompressibility constraint \( \lambda_f\lambda_t^2 = 1 \) we have three equations and three unknowns to which exact solutions can be found easily using for instance the Python module \texttt{scipy.optimize}. For the active tension a Gaussian function is used

\[
T_A(t) = \gamma \exp(-\omega(t - t^*)^2) \quad t \in [0, 1],
\]

where \( \gamma = 5000 \) Pa, \( t^* = 0.5 \) s and \( \omega = 100 \) s\(^{-2}\).
To solve this as a boundary value problem using FEM, we need a proper set of boundary conditions. The cube has to be able to contract freely to get a uniform deformation, while ensuring uniqueness of the solution. Ideally we would like to have Neumann conditions on all the boundaries, but since that allows all rigid motions as solutions, we instead restrict three faces of the cube from moving in its normal direction. The boundary value problem can then be stated,

\[
\nabla \cdot \mathbf{F} \mathbf{s} = 0 \quad \text{in} \quad \Omega_0,
\]
\[
U_1 = 0 \quad \text{on} \quad \Gamma_1,
\]
\[
U_2 = 0 \quad \text{on} \quad \Gamma_2,
\]
\[
U_3 = 0 \quad \text{on} \quad \Gamma_3,
\]

where \( \mathbf{U} = U_1 \mathbf{i} + U_2 \mathbf{j} + U_3 \mathbf{k} \) and \( \Gamma_i \) is a plane defined by \( X_i = 0, \ i = 1, 2, 3. \)

Figure 4.2: Uniform deformation simulated using FEM.

Comparing the solution computed using FEM with the exact solutions to the equations (4.14)-(4.15), Figure 4.3, we see that the solution curves are virtually identical. The expected physical behavior is also observed; significant contraction in the fiber direction leading to extension of the tissue in the transverse directions, see Figure 4.2. To make differences between the approximate solutions obtained using FEM and the exact solutions more apparent, the absolute error is plotted and featured in Figure 4.4. The largest error over the time interval in the fiber stretch \( \lambda_f \) is less than \( 3.5 \cdot 10^{-11} \), for the transverse stretch \( \lambda_t \) less than \( 2.5 \cdot 10^{-11} \) and for the hydrostatic pressure \( p \) less than \( 3.0 \cdot 10^{-8} \).
Figure 4.3: Comparing the solution computed using FEM with the exact solution.
Figure 4.4: Error in stretch in fiber direction (top), transverse direction (middle) and in pressure (bottom).
4.2.2 Method of Manufactured Solutions

The previous test case covered many aspects of the physical problem, and many important elements of the implementation like active stress, the specific material model and the incompressibility constraint. This is all good and well, but the presented solution (4.6), had one serious flaw; inserting it into the equation of motion renders the acceleration term zero. To find an exact solution that exercises every term of the equation we employ the method of manufactured solutions (MMS).

The method of manufactured solutions is a general approach for creating exact solutions to mathematical models. The concept of MMS is to choose a solution, and then modify the original equations such that chosen solution satisfies the modified equations. For the purpose of code verification it is not required that the manufactured solution is related to the physical problem.

We consider a unit cube made of a St. Venant-Kirchhoff material and the analytical solution

\[ \mathbf{U}_e = \cos(X_2)\sin(t)\mathbf{i} + \cos(X_3)\sin(t)\mathbf{j} + \cos(X_1)\sin(t)\mathbf{k}. \]  
(4.16)

Sine and cosine functions are suitable choices for MMS due to smoothness, differentiability and their restricted magnitude of unity.

Inserting the exact solution (4.16) into \( \nabla \cdot \mathbf{P} = \frac{\partial^2 \mathbf{U}}{\partial t^2} \) there is a residual \( \mathbf{R} \), i.e.

\[ \mathbf{R} + \nabla \cdot \mathbf{P} = \frac{\partial^2 \mathbf{U}}{\partial t^2}. \]  
(4.17)

This is the modified equation to which we know the solution \( \mathbf{U}_e \). Computing \( \mathbf{R} \) is usually done with software supporting symbolic mathematics, but we have done it using the unified form language in FEniCS. Initial and boundary conditions are obtained from the exact solution by evaluation at \( t = 0 \) and on \( \mathbf{X} \in \partial \Omega_0 \).

Having a problem with an exact solution we perform order of accuracy tests. Order of accuracy tests studies the rate at which the discrete solution approaches the exact solution in the limit as the discretization parameters go to zero, and is argued to be the most rigorous approach for code verification [32].

First we investigate the convergence as result of spatial refinement by evaluating the exact and the computed solution over some time interval, while
refining the mesh between each simulation. Having refined the mesh the \textit{rate of convergence} can be computed according to

\[ r_i = \frac{\log(E_i/E_{i-1})}{\log(h_i/h_{i-1})}. \] (4.18)

Here \( i \) is some level of refinement with error \( E_i \) measured in \( L^2 \)-norm and cell diameter \( h_i \).

Convergence due to refinement in time is conducted similarly, except that \( h \) is held constant, and is replaced by \( \Delta t \) in (4.18).

We see from the space refinement test, Table 4.1, and the time refinement test, Table 4.2 that the expected convergence rates are recovered.

Table 4.1: Computed convergence rates \( r \) for Lagrange elements of different degrees \( P_m - P_n \), where \( m \) is the degree used for displacement and \( n \) is the degree used for velocity. \( N \) refers to the number of elements in each direction. \( N/A \) means that the quantity was not computed due to lack of memory.

<table>
<thead>
<tr>
<th>( N )</th>
<th>( P_2 - P_1 )</th>
<th>( P_2 - P_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E )</td>
<td>( r )</td>
<td>( E )</td>
</tr>
<tr>
<td>3</td>
<td>2.45174820e-08</td>
<td>-</td>
</tr>
<tr>
<td>6</td>
<td>7.38687871e-09</td>
<td>1.73077402</td>
</tr>
<tr>
<td>10</td>
<td>2.82931989e-09</td>
<td>1.87866244</td>
</tr>
<tr>
<td>16</td>
<td>1.14192378e-09</td>
<td>1.93045742</td>
</tr>
<tr>
<td>24</td>
<td>5.16508235e-10</td>
<td>1.95672413</td>
</tr>
</tbody>
</table>

Table 4.2: Computed convergence rates \( r \) for \( P_2 - P_1 \) Lagrange elements, where \( P_2 \) elements are used for the displacement and \( P_1 \) elements are used for the velocity. Here 16 elements in each direction are used.

<table>
<thead>
<tr>
<th>( \Delta t )</th>
<th>( E )</th>
<th>( r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.00122248223</td>
<td>-</td>
</tr>
<tr>
<td>0.05</td>
<td>0.000535837479</td>
<td>1.18994610</td>
</tr>
<tr>
<td>0.025</td>
<td>0.000254424389</td>
<td>1.07455851</td>
</tr>
<tr>
<td>0.0125</td>
<td>0.000124456388</td>
<td>1.03159668</td>
</tr>
</tbody>
</table>
Results

The results featured in this section can be reproduced by running the scripts `dynamic_elastic.py`, `dynamic_viscoelastic.py` and `quasistatic.py`. It is worth mentioning that the pressure curves in this chapter are the intra-ventricular pressure, i.e. the boundary condition on the endocardium, and not the hydrostatic pressure introduced through the incompressibility constraint.

All simulations were done on the mesh presented in Section 4.1.2 and with $\Delta t = 1$ ms.

5.1 Ventricular Geometry Change

The left ventricle undergoes severe geometry changes during the cardiac cycle, as described in Section 3.5. To visualize that our model captures the shape changes we include a time series of the computed geometry, see Figure 5.1. The results featured in Figure 5.1 is from the quasistatic model, but the major trends in geometry change are the same for all three models.

Video visualizations of the results are available to the reader at:

https://github.com/krikarls/master_public.git
5.2 Pressure and Volume Curves

Simulating one cardiac cycle for the three different models defined in Section 3.5, yields the results in Figure 5.2 and Figure 5.3.

By looking at the first 200 ms of volume curves one can see major differences between the models, see Figure 5.2. It should be noted that initial conditions in the dynamic models play an important role at this early stage of the simulation. The velocities are initially zero and a rapidly increasing pressure load is applied. At this point the strains are very low, thus the tissue very soft, see Figure 3.3, and only small amounts of stress are required to give significant deformations. In a quasistatic model the increasing pressure instantly changes the volume, such that equilibrium is maintained. When inertial effects are included the ventricular walls need time to accelerate, hence no immediate volume change. As a result a gap develops between the equilibrium volume, i.e. quasistatic volume, and the current volume. Closing this gap requires high wall velocities, and if there are no forms of damping or energy dissipation, the dynamic model overshoots the equilibrium. This leads to a periodic motion in the dynamic model around the quasistatic volume curves, see Figure 5.2 around 70 ms. The oscillations in the volume curve die out as the volume increases and the ventricle gets stiffer.

Figure 5.1: Visualization of the geometry changes of the ventricle over the cardiac cycle. This simulation is done using the quasistatic model.
Figure 5.2: Intraventricular volumes, pressures and average wall velocities as a function of time over the cardiac cycle for the different models.

The oscillations are obviously unphysiological as they indicate severe backflow into the left atrium. This is a consequence of assuming that the myocardium is a perfectly elastic material while including inertial effects. Using a viscoelastic model, continuously dissipating energy, causes a natural damping such that there are no volume oscillations. This is well illustrated by the decreasing velocity curve of the viscoelastic model in Figure 5.2.
5.2. PRESSURE AND VOLUME CURVES

The pressures are more or less the same for all the models during the isovolumetric contraction, but in the early ejection phase (around 600 ms) small oscillations in the pressure of the dynamic-elastic model is observed.

![Pressure-volume loop](image)

**Figure 5.3:** Intraventricular pressure against volume over one cardiac cycle, for all the three models.

A useful alternative to Figure 5.2 for displaying pressure and volume changes during the cardiac cycle is the *pressure-volume loop*, or simply *PV-loop*, shown in Figure 5.3. In the pressure-volume loop the vertical line segments are the isovolumetric phases of the cardiac cycle, the lower limb is ventricular filling, and the upper segment represents the ejection phase. The pressure-oscillations in the early ejection phase for the dynamic-elastic model become more apparent in the PV-loop.

The stroke-volume can be read from pressure-volume plots as the difference on the horizontal axis between the vertical line segments. For our three models the stroke-volumes are all the same. Not unexpected, since the end-diastolic volume is highly dependent on the fixed pressure in phase 1, and end-systolic volume is forced by an if-test in the implementation.
The oscillations during the early diastole in the dynamic-elastic model causes volume to change considerably without any significant change in pressure, this can be observed in the PV-loop as discontinuities in the lower limb of the PV-curve in Figure 5.3.

5.3 Stress and Strain Distributions

The stress and strain distributions featured in this section are taken at different time steps during the cardiac cycle, indicated by roman numerals in Figure 5.2. The plots in Figure 5.4 - 5.5 are created using ParaView [20].

5.3.1 Fiber Strain

The overall expected strain trends are captured by all the three models. We observe that muscle fibers are extended during the diastole and when the contraction starts the extension is reduced. The fibers are maximally contracted right after all the blood has been ejected out of the ventricle, while the active muscle tension is still high. The large gradients that can be seen near the base of the ventricle is due to the imposed displacement condition. Despite the fact that there are significant differences between the models in intraventricular volume in the mid-diastole, it is difficult to distinguish them in terms of fiber stretching, as seen in Figure 5.4.

5.3.2 Fiber Stress

The increasing pressure load inside the ventricle leads to steep gradients in the fiber stress near the endocardium in the late diastole and the isovolumetric contraction phases, see Figure 5.5. The stresses are at maximum values when the active tension, and pressure, is at its peak in the early ejection phase(IV). The stress distributions in the three models are virtually identical on a scale ranging from the smallest to the largest stresses.
Figure 5.4: Snapshots of strain in the direction of the fibers, taken from the volume-time curve in Figure 5.2, during the cardiac cycle employing a quasistatic (left), dynamic-elastic (middle) and dynamic-viscoelastic (right) model.
Figure 5.5: Snapshots of the stress in the direction of the fibers, taken from the volume-time curve in Figure 5.2, during the cardiac cycle employing a quasistatic (left), dynamic-elastic (middle) and dynamic-viscoelastic (right) model.
5.4 Multiple Heartbeats: Washing out the Initial Condition

To see how the dynamic models behave in the absence of effects from the initial condition, we simulate multiple cardiac cycles until stabilization. By stabilization we here mean that there are no changes from one cycle to the next. For the quasistatic model this is not needed, since the solution at $t_n$ depends only on $T_A(t_n)$ and $p(t_n)$, and is totally independent of previous states of the system.

Going into the second cycle, the velocities in the dynamic-elastic model are already very high, and stay high throughout the diastole, see Figure 5.7. In this phase the velocities are about three times higher than for the first cycle. Oscillations in internal volume are also greater and lasts until the isovolumetric contraction begins, as opposed to in the first cycle.

![Figure 5.6: Intraventricular pressure(top) and ventricular wall displacement(bottom) for the dynamic-elastic model in the second cycle. The displacement is measured in a point 2 cm from the base, at the endocardium, and is measured as vector length.](image)
Intraventricular pressure oscillations in the early ejection have increased significantly, see Figure 5.7 around 600 ms. This is even more apparent in the PV-plot in Figure 5.9. The pressure oscillations are most likely a result of the oscillating ventricular wall. To further support this statement we include a graph of the wall displacement for the time interval 585 - 620 ms. The periodic motion of the ventricular wall coincide almost perfectly with the internal pressure, as can be seen in Figure 5.6.

Towards the end of the second cycle the velocity is approximately the same as in the first cycle. The dynamic-elastic model has then stabilized and cycle number two and three are indistinguishable, see Figure 5.7 and 5.9.

Looking at the results in Figure 5.8, the dynamic-viscoelastic model is almost unaffected by the high velocity at the beginning of the second cycle. Kinetic energy is rapidly dissipated and the velocity is about halved in 100 ms, and entering the isovolumetric contraction phase the velocity is almost zero. This is expected behavior, because the filling of the ventricle is very slow at the end of the diastole due the stiffness of the tissue at high strains.

The pressure and volume curves are almost identical for all the three cycles using the dynamic-viscoelastic model, see Figure 5.8 and 5.10.
5.4. MULTIPLE HEARTBEATS: WASHING OUT THE INITIAL CONDITION

Figure 5.7: Intraventricular pressure and volume curves for multiple cardiac cycles, using a dynamic-elastic model.
Figure 5.8: Intraventricular pressure and volume curves for multiple cardiac cycles, using a dynamic-viscoelastic model.
5.4. MULTIPLE HEARTBEATS: WASHING OUT THE INITIAL CONDITION

Figure 5.9: Intraventricular pressure against volume over three cardiac cycles, using the dynamic-elastic model.
Figure 5.10: Intraventricular pressure against volume over three cardiac cycles, using the dynamic-viscoelastic model.
5.4. MULTIPLE HEARTBEATS: WASHING OUT THE INITIAL CONDITION
Summing up and Looking Ahead

We here summarize the work done in this M. Sc. project by listing its main findings, limitations and possible extensions.

6.1 Main Findings

The primary objective of this project was to investigate the effects of inertia in ventricular mechanics. As accounting for inertial forces proved to have serious consequences in terms unphysiological oscillations in internal volume and pressure, we started investigating the causes. This led us toward alternatives to elastic material models.

The main findings in thesis can be summed up in the following points:

- We discovered that assuming the tissue of the heart can be modeled as a perfectly elastic material, including inertial effects can lead to unphysiological oscillations in the ventricular walls, causing oscillations in intraventricular pressure and volume.

- The ventricular wall oscillations are most likely linked to elastic materials capacity to store mechanical energy. Using a viscoelastic model, continuously dissipating kinetic energy, had a damping effect on the oscillations.

- Despite differences in intraventricular pressures and volumes between the models, the stress and strain distributions were virtually identical.
6.2 Limitations and Extensions

We consider the most important limitations, and possible extensions, of this study to be:

- For more realistic boundary conditions and phase transitions the ventricular dynamics models should be coupled to a complete circulatory model.

- More sophisticated viscoelastic models for the myocardium, like the two recently suggested by Cansız et al. [7] and Gültekin et al. [17] should be studied in the context of inertial effects.

- Conduct similar numerical experiments on biventricular geometries.
Appendices
Implementation Technicalities

A.1 Computing Intraventricular Volume with FEniCS

The intraventricular volume $v$ lies outside the computational domain and can therefore not be computed directly in FEniCS. We have to use some mathematical manipulations to be able to do it. Let $V$ denote the volume of the inner ellipsoid in the reference configuration with boundary $S = S_1 \cup S_2$, and $v$ denote the volume in the current configuration with boundary $s = s_1 \cup s_2$.

![Diagram](image.png)

Figure A.1: Undeformed(left) and deformed(right) state of ellipsoid.

The volume we want to compute is given by

$$v = \int_v 1 \, dv.$$ 

By making use of the fact that $\frac{1}{3} \nabla \cdot x = 1$ and the divergence theorem we
Computing Intraventricular Volume with FEniCS

\[ v = \frac{1}{3} \int v \nabla \cdot \mathbf{x} \, dv = \frac{1}{3} \left[ \int_{s_1} \mathbf{x} \cdot \mathbf{n} \, ds + \int_{s_2} \mathbf{x} \cdot \mathbf{n} \, ds \right]. \]

The contribution of the integral over \( s_2 \) is zero

\[ \int_{s_2} \mathbf{x} \cdot \mathbf{n} \, ds = \int_{s_2} x_3 \, ds = 0. \]

Finally, using the definition of the displacement field \( \mathbf{x} = \mathbf{X} + \mathbf{U} \) and Nanson’s formula \([22] \): \( \mathbf{n} \, ds = \mathbf{F}^{-T} \mathbf{J} \mathbf{N} \, dS \), we can then express the volume as

\[ v = \frac{1}{3} \int_{s_1} (\mathbf{X} + \mathbf{U}) \cdot \mathbf{F}^{-T} \mathbf{J} \mathbf{N} \, dS. \]

An implementation in python with FEniCS is shown below. The minus sign appears because the mesh normal points into the volume we are computing, which is opposite of the convention in the divergence theorem.

```python
def Compute_Volume(u):
    X = SpatialCoordinate(mesh)
    vol = assemble((-1./3.)*dot(X + u, inv(F).T*n_mesh)*ds(endo))
    return vol*1e6  # scale volume to ml
```
A.2 Algorithm for the Unknown Pressure in Isovolumetric Phases

```python
def Isovolumetric_PressureUpdate(i, volume_constraint):
    dp = 10.
    iterations = 0
    err = 1.
    p_0 = pressure[i-1]

    while err > 0.01:
        iterations += 1
        Solve_System(active_tension[i], p_0)
        V_0 = Compute_Volume(u)
        Solve_System(active_tension[i], p_0+dp)
        V_1 = Compute_Volume(u)
        f = (Vf_0 - volume_constraint)
        dfdp = (Vf_1 - Vf_0) / dp
        p_1 = p_0 - f / dfdp
        err = abs(p_1 - p_0)
        p_0 = p_1

    print 'Error after ' , iterations , ' iterations = ', err
    print 'Converged in ' , iterations , ' iterations 
    print 'Pressure is now ', p_1 , ' 
    return p_1
```

A.3 Algorithm for the Unknown Pressure in Ejection Phase

```python
def PressureUpdatePhase3(i, V_ED):
    dp = 10  # Pa
    DT = (t[i] - t[i-1])*1000  # ms
    iterations = 0
    err = 1.
    p_0 = pressure[i-1]  # Pa

    while err > 0.01:
        iterations += 1
        Solve_System(active_tension[i], p_0)
        V_0 = Compute_Volume(u)
        Solve_System(active_tension[i], p_0 + dp)
        V_1 = Compute_Volume(u)
        dVdp = (V_1 - V_0)/dp
        g = (Vf_0 - volume[i-1]) + C_art*(p_0-pressure[i-1]) + DT*(p_0-P_o )/R_per
        dgdp = dVdp + C_art + DT/R_per
        p_1 = p_0 - g/dgdp
        err = abs(p_1 - p_0)
        p_0 = p_1
        print '
 Error after ', iterations , ' iterations = ', err
        print '
 Converged in ', iterations , ' iterations 
 '
print '\n Current outflow is', (Vf_0 - volume[i-1]), ' 
'
return p_1
```
Model Parameters

Table B.1: Material parameters for the elastic model for the myocardium.

<table>
<thead>
<tr>
<th></th>
<th>$a$ [Pa]</th>
<th>$a_f$ [Pa]</th>
<th>$b$</th>
<th>$b_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>original</td>
<td>2280</td>
<td>1685</td>
<td>9.726</td>
<td>15.779</td>
</tr>
<tr>
<td>scaled</td>
<td>228.0</td>
<td>116.8</td>
<td>7.780</td>
<td>11.834</td>
</tr>
</tbody>
</table>

Table B.2: Various model parameters

<table>
<thead>
<tr>
<th>symbol</th>
<th>numerical value</th>
<th>dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\rho$</td>
<td>1000</td>
<td>$[kg/m^3]$</td>
</tr>
<tr>
<td>$\eta$</td>
<td>0.1</td>
<td>[-]</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>22.6</td>
<td>$[Pa \cdot s]$</td>
</tr>
<tr>
<td>$\beta$</td>
<td>0.01</td>
<td>[s]</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>1</td>
<td>[ms]</td>
</tr>
</tbody>
</table>

Table B.3: Windkessel model parameters.

<table>
<thead>
<tr>
<th>$C_{art}$ [ml/Pa]</th>
<th>$R_{per}$ [Pa/ml]</th>
<th>$p_0$ [Pa]</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.001</td>
<td>20000</td>
<td>500</td>
</tr>
</tbody>
</table>
Bibliography


