

In Silico Modeling, Simulation and Optimization of Human Cardiac Motion

Zur Erlangung des akademischen Grades eines

DOKTOR-INGENIEURS

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Dipl.-Math. techn. Lukas Baron

geb. in Seeheim-Jugenheim

Tag der mündlichen Prüfung:	10. Februar 2022
Hauptreferent:	Prof. Dr. rer. nat. Olaf Dössel
Korreferent:	PD Dr. rer. nat. Mathias J. Krause

Erklärung

Ich versichere wahrheitsgemäß, die Arbeit selbstständig angefertigt, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde.

Karlsruhe, den 21.12.2021

Lukas Baron

Contents

Abstract	1
Zusammenfassung	2
Acknowledgments	3
I. Introduction and Fundamentals	5
1. Introduction	6
1.1. Motivation	6
1.2. Structure of the Thesis	8
2. Physiological Fundamentals	10
2.1. Cardiac Anatomy	10
2.2. Meso- and Microstructure of the Heart	12
2.3. Cardiac Physiology and Tension Development	14
2.4. Vascular System and the Cardiac Cycle	14
3. Mathematical Modeling	19
3.1. Continuum Mechanics and Description of Deformation and Stress	19
3.2. Governing Equations	23
3.3. Spatial Discretization	24
3.4. Temporal Discretization	26
3.5. Force Modeling Overview	28
3.6. Modeling of Passive Stress	29
3.7. Modeling of Active Stress	32
3.8. Modeling of the Pericardium	37
3.9. Modeling of Circulatory System	40
3.10. Stress-free State Estimation	42
3.11. The Heart Model by Fritz2014	45
4. Implementational Aspects	50
4.1. The Biomechanical Framework CardioMechanics	50
4.2. Adaptive Time Increments	52
4.3. Coupling of Multiple Systems	57
4.4. Parallelization	60
4.5. Geometric Model Generation	60

Contents

II. Approaches and Results	69
5. Validation of the FEM Solver	70
5.1. Necessary Mesh Resolution for Mesh Convergence	70
5.2. Scalability Study and Parallel Performance	78
6. Improving the Physiological Behavior	86
6.1. Application I – Estimation of Passive Material Parameters using Mathe- matical Optimization	86
6.2. Application II – Length-dependent Active Behavior	105
6.3. Application III – A Fully Coupled Circulatory System	123
6.4. Application IV – A zero-dimensional Pericardium Modeling	149
7. Summary and Outlook	164
List of Publications with Contributions of the Author	167
Bibliography	177

Abstract

Cardiac diseases are the number one reasons for death in the western world. Computational simulations provide the opportunity to conduct experiments and predictions that are not possible in humans due to ethical and other reasons. High performance computation allows the use of demanding coupled computational models of high complexity and a high level of detail, complying with a wide range of experimental data from the human heart. In this thesis, different aspects of computational heart modeling are covered: models describing passive tissue behavior, active contractile behavior, circulatory system modeling, influences of the pericardium and surrounding tissue on the heart as well as methods to obtain suitable parameters for these models. For each aspect, several modeling approaches are presented and compared. Finally, a scalability evaluation of the highly-parallelized implementation and an evaluation of the proper choice of mesh resolution for credible numerical results are covered. Concludingly, this thesis allows the reader to gain insights into the complexity of computational heart modeling and to make an appropriate choice of models and parameters suitable for specific applications.

Zusammenfassung

Herzkrankheiten sind die häufigste Todesursache in der westlichen Welt. Computersimulationen bieten die Möglichkeit, Experimente und Vorhersagen durchzuführen, die am Menschen aus ethischen und anderen Gründen nicht möglich sind. Hochleistungsrechner ermöglichen den Einsatz anspruchsvoller gekoppelter Rechenmodelle von hoher Komplexität und hohem Detaillierungsgrad, die mit einer Vielzahl von experimentellen Daten aus dem menschlichen Herzen übereinstimmen. In dieser Arbeit werden verschiedene Aspekte der rechnergestützten Herzmodellierung behandelt: Modelle zur Beschreibung des passiven Gewebeverhaltens, des aktiven kontraktiven Verhaltens, der Modellierung des Kreislaufsystems, der Einflüsse des Herzbeutels und des umgebenden Gewebes auf das Herz sowie Methoden zur Ermittlung geeigneter Parameter für diese Modelle. Für jeden Aspekt werden mehrere Modellierungsansätze vorgestellt und bewertet. Zusätzlich wird eine Bewertung der parallelen Skalierbarkeit und der richtigen Wahl einer passenden Gitterauflösung für verlässliche numerische Ergebnisse vorgenommen. Somit ermöglicht diese Arbeit dem Leser, einen Einblick in die Komplexität der rechnergestützten Herzmodellierung zu gewinnen und eine geeignete Auswahl von Modellen und Parametern für spezifische Anwendungen zu treffen.

Acknowledgments

This work was carried out at the Institute of Biomedical Engineering (IBT), Karlsruhe Institute of Technology (KIT). I would like to thank all persons who contributed to the success of the present work. First and foremost, I would like to express my sincere and heartfelt gratitude to Olaf Dössel for giving me the opportunity to work at his institute on a stimulating topic, his interest in my work and his review of the dissertation. My heartfelt thanks goes to all colleagues and students during my time at IBT for valuable feedback on my research activities and also their mental support on long working days. My special thanks go to my students, each of them had to deal with a difficult topic and did a great job, without exception. A big thank goes to Jochen Schmid and Steffen Schuler for their support during my doctoral studies. Both are amazing persons to talk to for any fruitful scientific discussion, no matter how absurdly impossible my ideas might have sounded. Furthermore, special mentioning deserve Gunnar Seemann, Bhawna Verma, Gustavo Lenis, Andreas Wachter and Yilun Su, they made me feel welcome in the group and always had an open ear for non-scientific aspects. Finally, I would like to thank my parents and family who have supported me throughout my studies and while working on this thesis. Without their support and patience, especially during difficult times, realization of this work would not have been possible. The biggest thank goes to Mathias Krause for accompanying my career as a scientist, his numerous advices in all parts of life and his helpful and always constructive contributions especially towards the end of my PhD studies.

Part I.

Introduction and Fundamentals

1. Introduction

1.1. Motivation

The motion of the human heart during contraction is the result of a complex process involving an equilibrium of a variety of forces acting at the same time. Activity of the human fetal heart can be detected as early as six weeks of pregnancy (Hertzberg et al., 1988), on the other hand cardiac diseases are the leading cause of death in the western world for people over 65 (Heron, 2019; Kotzeva, 2019). Its pumping function drives the circulatory system, metabolism and nutrition of the organs. Computational simulation of the human heart can be of great benefit for success prediction during therapy planning and allows experimental insights that are otherwise not possible due to ethical reasons. Moreover, model-enhanced measurements give access to otherwise inaccessible parameters valuable for diagnostics.

State of Research Challenges in modeling cardiac motion arise from a number of seemingly independent involved components: Passive tissue, active contraction, blood pressure (circulatory system) and the heart's environment all are areas with large uncertainties in almost all models' parameters. Designed for specific phenomena in an isolated context, a model does not necessarily work as expected in a combined context, the modeling choices need to fit together. During the cardiac cycle, chamber volumes change by more than 50 % and heart deformation is similarly considerable. Linearized approaches in modeling or the numerical solution process therefore can not be used.

Passive tissue models describe local forces due to local deformation and don't necessarily fulfill a global behavior. These models are naturally designed around measurements of local deformation (e. g. ex-vivo shear data by Dokos et al. (2002)). According parameters for the different models were determined by Schmid et al. (2008, 2009) or in recent years from tagged MRI (Hadjicharalambous et al., 2015). The global behavior in humans however is also known to follow a certain pressure-volume relationship (Klotz et al., 2006), independent from sex, age or health state. Additionally, different from electrical excitation modeling, anisotropic properties are mandatory as fiber orientation prescribes the direction of contraction, and incompressibility prolongation in the other ones. Models for active tension depend on a specific amount of local cell strain to develop sufficient tension (Sachse et al., 2003; Land et al., 2017). Another important role plays the time course of calcium development or the calcium transient, which needs to be at least periodic and even then does not necessarily work as expected. Since electrical excitation activation happens much quicker than the mechanical motion, it is often computed separately on a higher resolution structured grid and predefined in models of the whole heart (Fritz et al., 2014; Gurev et al., 2015). Circulatory system models interact with mechanical motion

1. Introduction

via exchange of pressure and volume changes. Local pressure information is normally not used unless the focus is specifically on intraventricular blood flow patterns, which then requires fluid structure interaction (Watanabe et al., 2004). Models for the surrounding pericardium are still in an early state and difficult to validate (Fritz et al., 2014; Kerckhoffs et al., 2007a). Data acquired during in-vivo measurements is exposed to a number of tertiary effects (passive, active forces, blood pressure), ex-vivo measurements are not ethically obtainable from healthy humans. Hearts donated to science often have altered elastic behavior, animal experiments on the other hand have pitfalls besides the obviously different dimensions. Pig hearts for example are mostly surrounded by very deformable lung tissue, whereas the lower half of human hearts is in the direct touch with liver tissue and diaphragm, with a more restricted motion in that area. Ideally, multiple heart beats are computed that reach a state where one beat behaves identical to the previous one (quasi-periodic static state), which requires periodically working models and easily increases simulation time from hours to days or weeks (Lassoued and Boubaker, 2020). Besides interdependencies between specific models and the computational challenge of multi model coupling itself, the number of unknowns thereby is very limited (mesh resolution) and numerical accuracy is difficult to obtain (e. g. by higher order elements, shell elements). Also, the creation of the geometric model is a laborious task and often only done for one or two chambers. Averaged cine MRI images are acquired over multiple heart beats and thus show smear. During segmentation, the typical resolution of cine MRI for humans is insufficient to show atrial and right ventricular walls, and even left ventricular myocardium is hard to distinguish from liver tissue. Wall thickness however is an essential factor for force development, ejection volume and driving of circulation and metabolism. A good quality of each of the deforming elements is essential to be pretained throughout the whole cardiac cycle. Problems during simulation of a fully-featured whole heart simulation are often caused by specific aspects of the geometric model, and require multiple iterations through the whole process of segmentation, mesh, fiber and model generation.

Currently, only few full heart simulation models are published that include more than one chamber and a circulatory system modeling. Sugiura et al. (2012) present a model involving biochemistry and fluid-structure interaction, which was used by Okada et al. (2017) with focus on the electrophysiological aspect to identify patients suitable to cardiac resynchronization therapy. Fritz et al. (2014) use a three-dimensional pericardium model to improve motion of the atrio-ventricular valve plane and the outer shape. Gurev et al. (2015) present a high resolution two-ventricle model exploring the limits of mesh resolution using advanced linear equation solvers. Land et al. (2017) worked on a ventricular tension model that specifically resolves the discrepant behavior of fiber-level models on organ level. Used as an atrial contraction model, Land and Niederer (2017) compared the results of different atrial model parameters on ventricular contraction, which showed surprisingly little effect with ejection fraction decreasing by less than 5%. In a similar study, Gerach et al. (2021) published a model which uses a very similar mesh, circulatory system modeling and patient-specific passive parameter estimation as developed in this thesis, but with a different tension development modeling to show effects of atrial ablation scars on hemodynamic parameters.

Aims and Solution Idea The main purpose of this thesis is to identify, describe and evaluate elastomechanic modeling approaches in each component suitable for application in fully-featured whole heart simulations. These models ideally provide periodic solutions (periodicity) and easy-to-determine parameters that are independent of a specific patient. A reproduction of certain component-specific aspects on a general level ensures a well-working interplay between different models.

Therefore, a tetrahedral mesh of the four-chamber geometry is generated for one specific patient outgoing from an MRI data set. New material parameters for passive tissue will be estimated by mathematical optimization techniques to obtain a better global behavior of the ventricle. Once the passive tissue behaves as expected, this allows to implement the active component by a length-dependent model of active tension and evaluate its results. As endocardial boundary condition, a new circulatory system model is developed and implemented that allows multiple heart beats until convergence. Due to lack of validation for the existing pericardium model, an alternative, reduced-order approach is implemented, which allows to evaluate both against each other and against measurement data. The computational demands make parallelization of the code with an efficient scalability necessary. Uncertainties in the required mesh resolution are resolved by conducting a study with different resolutions on a simple model of the left ventricle.

1.2. Structure of the Thesis

The remainder of the thesis is structured as follows. Part I presents the necessary background of physiological, mathematical, modeling and implementational fundamentals. Section 1 presents the general motivation behind numerical heart modeling as well as current challenges on the topic and the state-of-the art. Section 2 gives the description of the physiology of the human heart at multiple levels of detail as well as the vascular system. Section 3 introduces the mathematical fundamentals of the continuum mechanics approach and the discretization methods in general that are necessary to solve the problem using a computer. Additionally, the second half of Section 3 collects modeling approaches that are specifically suitable for describing passive and active properties of heart tissue as well as forces acting on the myocardial walls from the interior (circulatory system) and exterior (pericardium and surrounding tissue). Section 3 finally closes with a short overview of the Fritz heart model, created in an important preceding thesis, that serves the base of this thesis. Section 4 gives descriptions of implementing the previous sections realized in a software framework named “CardioMechanics”, the adaptive choice of temporal step size, strategies used to couple the different modeling components, software parallelization up to super computer scale, as well as the workflow used to generate a geometric mesh from a specific patient’s data.

Part II presents several projects that aim towards a modeling setup of the human heart where each isolated component is validated as well as the behavior of the coupled system as a whole. Besides modeling aspects, this includes aspects like mesh choice or validation of the FEM solver. Section 5.1 clarifies the necessary mesh resolution that is quantitatively needed when aiming to achieve a specific numerical accuracy, and Section

1. Introduction

5.2 measures the performance benefit of the software parallelization, using up to 256 cores. Section 6.1 treats the isolated behavior of a ventricle to estimate parameters for the passive material model using mathematical optimization. Section 6.2 investigates two different models for active contraction forces, introduces a length-dependent model and evaluates its influences on the level of a muscle fiber as well as of the heart as a whole. Section 6.3 describes the modeling of a closed-loop circulatory system and uses these models to add external forces onto the interior surface of the myocardial wall using strong iterative coupling. Section 6.4 investigates two different models of the heart-surrounding pericardium that is used to further add external forces onto the exterior surface, resulting in the proposition of a new zero-dimensional model. Finally, Section 7 summarizes the results obtained in this thesis and gives an outlook of future works.

2. Physiological Fundamentals

2.1. Cardiac Anatomy

The human heart is a muscular organ located in the center of the chest and has approximately the size of a fist. It is neighbored in the upper part by the lungs and skeleton, and in the lower part by the liver and diaphragm (Figure 2.3). In healthy persons, the heart is surrounded by the pericardium, a sac-like structure (Figure 2.1).

The purpose of the heart is to drive the cardiovascular circulatory system which itself ensures the supply of the organs with oxygen, hormones, minerals and nutrients, and the evacuation of waste products as well. The heart consists of four chambers, usually separated in two functional parts, consisting of one atrium and ventricle each. The right half consists of the right atrium and the right ventricle and drives the pulmonary circulation with the purpose to oxygenate deoxygenated blood in the lungs. The left half consists of the left atrium and left ventricle which have more muscle mass to develop a higher pressure needed in the systemic circulation. In order to ensure an unidirectional blood flow, the blood exchange between atrium, ventricles, pulmonary and the aorta is regulated via heart valves that work mostly passively. Between atrium and ventricle are the atrioventricular valves located, called mitral valve (left half) and tricuspid valve (right half). The ventricles are thereby bounded on both sides to regulate inward as well as outward flow (Figure 2.2). Moreover, the ventricles contain papillary muscles to prevent bulging of the valves towards the atria during contraction, which cover around 23% of the chamber volume (Chuang et al., 2012).

The heart contraction can be described by two phases. The *systolic phase* or *systole* denotes the contraction of a specified chamber, usually the ventricle if not denoted otherwise, and the heart ejects blood. During the *diastolic phase* or *diastole*, the heart chamber relaxes by a lowering contraction force and a release of previously stored passive energy, and refills with blood. Important temporal marks of the cardiac cycle are the end-diastole and end-systole as these specifically denote opening and closing events of certain valves as well as beginning and ending of active contraction. The wall thicknesses differ in the four chambers and depend on the pressures that need to be produced by them. Normal peak pressures in the atria are around 20 mmHg and the wall thickness fairly is thin. Ventricular walls are thicker, in the right ventricle the peak pressure is around 30 mmHg and the thickness up to 10 mm, in the left ventricle the pressure goes up to 140 mmHg with a thickness of 20 mm (Schmidt et al., 2007). These wall thicknesses describe the end-diastolic state and increase considerably during contraction.

2. Physiological Fundamentals

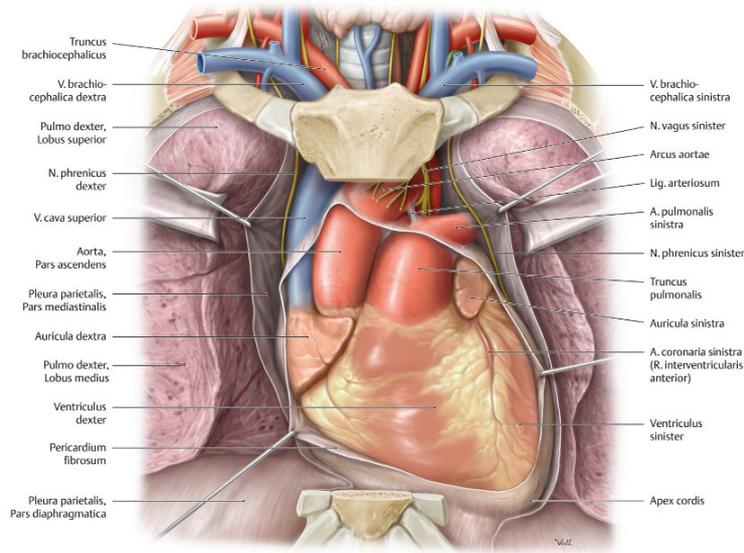


Figure 2.1.: The heart enclosed by the pericardium, from Schünke et al. (2009).

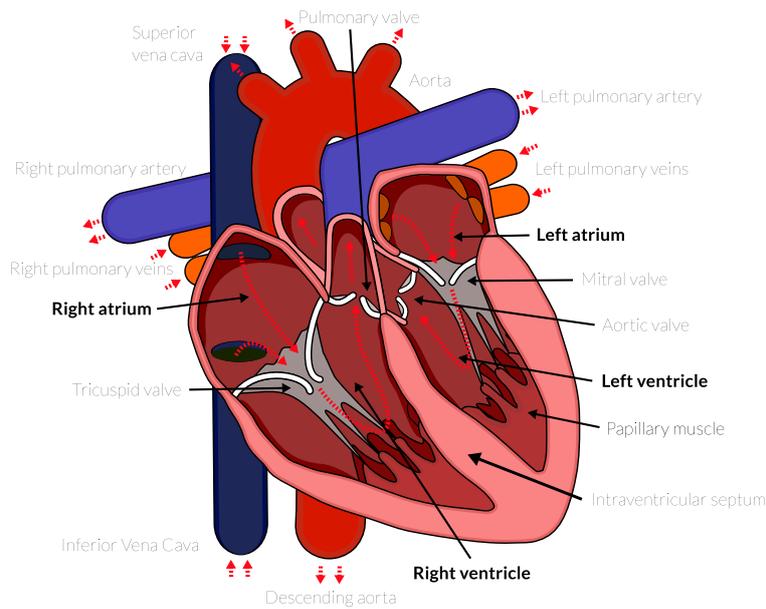


Figure 2.2.: Schematic four-chamber view of a human heart. Left and right half are separated by a septum and the ventricles are bounded by valves on both ends, allowing to direct the blood flow during the different phases of the cardiac cycle, from Fritz (2015).

2. Physiological Fundamentals

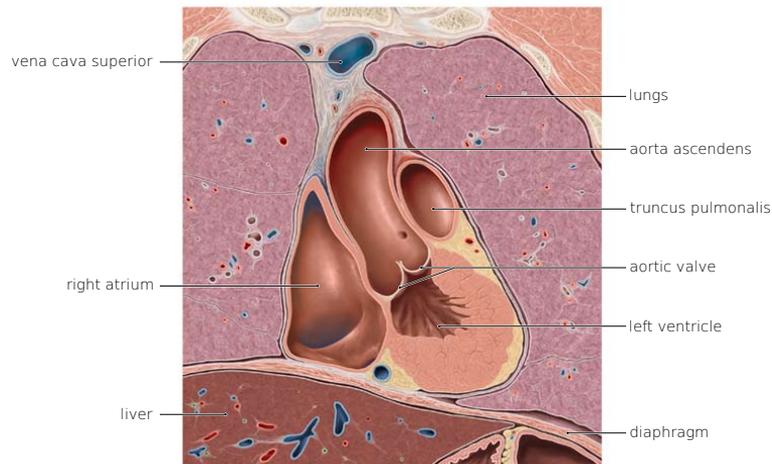


Figure 2.3.: Cross sectional view of the human heart in a torso surrounded by lungs and the diaphragm, picture adapted from Schünke et al. (2009).

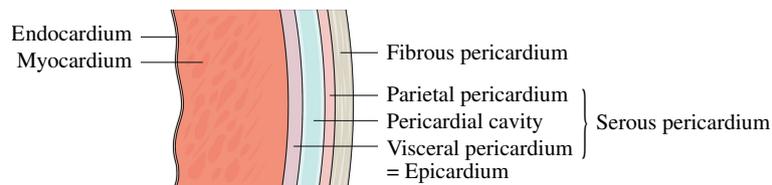


Figure 2.4.: Close view of the myocardium and the structure of the adjacent pericardial layers (adapted from OpenStax (2017), Chapter 19, Figure 5).

2.2. Meso- and Microstructure of the Heart

The myocardial wall consists of several layers, called endocardium, mid-wall and epicardium in the order from inside to outside. Myocardium is enclosed by the pericardium, a multi-layer structure, which itself again is built up of two main layers (Figure 2.4). The purpose of the outer, fibrous pericardium is to fix the position of the heart within the thorax. The serous pericardium is constructed of two subsequent layers. While the visceral pericardium is attached to the epicardium, the parietal pericardium is connected with the fibrous pericardium. The pericardial cavity located in-between the visceral and parietal layers is filled with lubricating fluid to reduce friction during the heart contraction.

In the myocardium, 70% of the volume consists of layers of fibrous muscle fibers. These are arranged in layers combining multiple fibers to sheets, which can be seen on microscope (Figure 2.5). These layers are connected to each other by perimysial collagen. Each myofiber is build up of several sarcomeres, which form the contractile element.

2. Physiological Fundamentals

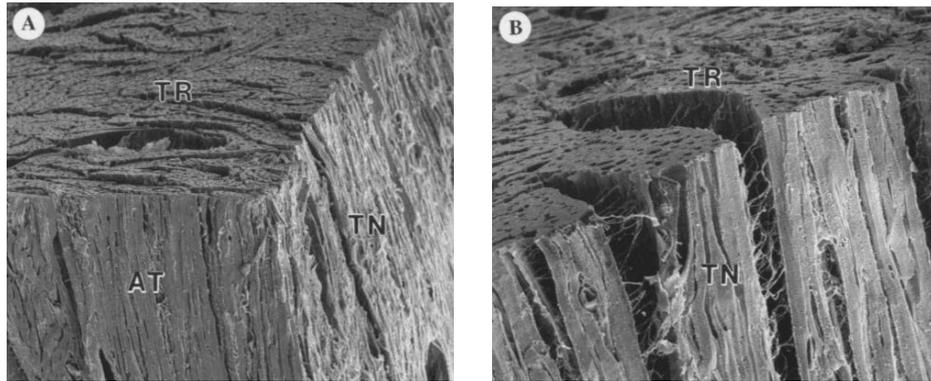


Figure 2.5.: Close-up view of myocardial tissue in a dog. Picture taken from LeGrice et al. (1995).

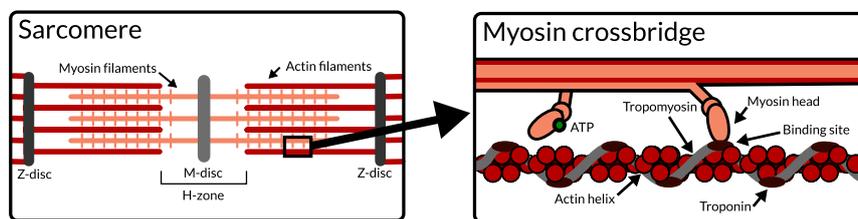


Figure 2.6.: Schematic view of a myocyte (left) and the action of the crossbridge actin heads (right), from Fritz (2015).

Each sarcomere has a length of approximately $2\ \mu\text{m}$ and consists of actin filaments attached to Z-discs, as well as myosin filaments attached to M-discs (Figure 2.6). Each myosin filament has a number of myosin heads that are responsible for the interaction between actin and myosin filaments. At rest, binding sites on the actin filaments are blocked by tropomyosin proteins. An increase in calcium concentration however causes tropomyosin to uncover the binding sites, and thereby allows the myosin heads to attach to them, creating a motion of the heads along the actin filament. In total, the overlap between actin and myosin filaments will change and a force develops between the Z-discs, which is the same as the fiber direction.

Physiologically, the length of a myosin filament is constant at around $1.6\ \mu\text{m}$, while the sarcomeres have a length of $1.9\ \mu\text{m}$ in the unloaded state, and $2.1\ \mu\text{m}$ at the end-diastole. Therefore, the length of a sarcomere is lower bound by the length of the myosin filament. During contraction, the shortening of a sarcomere is limited to at most $1.6/1.9 \approx 84\%$ of the unloaded length (or $1.6/2.1 \approx 76\%$ of the end-diastolic length).

2.3. Cardiac Physiology and Tension Development

Contraction of muscle cells in the heart is triggered by electrical stimulation. In normal healthy sinus rhythm, electrical excitation starts at the sinus node located in the right atrium. An electrical excitation wave spreads over the atria, reaches the atrioventricular (AV) node, located between atria and the ventricles. The AV node adds a delay of around 100 ms to give the atria time to contract and the ventricles time to fill with blood. Subsequently, the activation distributes over the bundle of His to the apex and from there via Purkinje fibers further over the ventricles. The electrical activation of the ventricles is inhomogeneous, the delay between earliest (apex) and latest activation (near the valve plane) corresponds to the length of the QRS complex in the ECG diagram and lasts between 80 ms and 120 ms.

On the cell level, a single heart muscle cell gets triggered by a stimulus current through the membrane. When the cell gets triggered, ion channels in the membrane open and close, and cause ion concentrations (Ca^{2+} , K^+) to change, building up the transmembrane potential, an electrochemical gradient between intracellular and extracellular space. The transmembrane voltage is usually at -70 mV at resting state and increases (depolarizes) up to 20 mV during activity, reaches a plateau phase and decreases (repolarizes) to the resting potential. After that, the normal, healthy cell is in a refractive phase and cannot be activated again for a short period of time to lead the electrical activation into a single direction (different from physical waves, two waves cannot ‘cross’) and suppress heart frequencies too high to eject blood (above 180 bpm). Ions can be exchanged directly from cell to cell through gap junctions, eventually causing a depolarization of neighboring cells.

During depolarization, the Ca^{2+} flow into the cell and the change of concentration causes an increased binding of Ca^{2+} to troponin-C proteins. These cause tropomyosin proteins to change their structure and finally a motion of myosin heads along actin filaments, also known as contraction.

2.4. Vascular System and the Cardiac Cycle

The blood flow – used to transport nutrients and waste products – is driven by contraction of the heart connected to a closed network of vessels. The circulatory system is separated into two non-connected parts, the systemic circulation and the pulmonary circulation (Figure 2.7). The *pulmonary system* transports blood from the right half of the heart to the lungs, where it gets oxygenated, and back to the left half. The *systemic circulation* transports blood from left half over the arterial system (aorta, arteries and arterioles) to the organs, and back to the right half using the venous system (venules and veins), where the circulation is closed. Larger vessels near the heart (aorta and pulmonary arteries) are elastic, allowing them to smooth out pressure spikes from the blood ejection and yielding a continuous blood flow (Windkessel effect). With distance from the heart, the diameter of the vessels decreases and the number of branches increases. This effectively reduces the velocity of blood flow at the organs, allowing for more time to exchange nutrients

2. Physiological Fundamentals

and waste products. Only the small vessels have the ability to actively influence their diameter and thereby resistance, giving them the possibility to control arterial blood pressure and distribution of blood to the different organs. Most of the blood is stored in the venous system, the larger veins hold about 70% of total blood volume (Klabunde, 2011).

The *cardiac cycle* is separated into *systole* or contraction phase and *diastole* or relaxation phase. The mechanical systole is defined as the interval between closure of AV valve and closure of semilunar (aortic/pulmonary) valve. It can be further subdivided. During *isovolumetric contraction*, where all valves are closed, the volume stays constant and the pressure rises. At that time, the AV valve deforms due to the large ventricular pressure towards atrium, which can be seen as “c” wave in the atrial pressure. The *ejection phase* follows the isovolumetric contraction. When a pressure threshold at the semilunar valve gets exceeded, the valve opens and blood is ejected from the ventricle. The ventricular volume decreases and the AV valve moves towards the apex. Thereby, the atrial volume increases and atrial pressure decreases (“x” descent). The ejection is called *rapid* before reaching the ventricular peak pressure and *reduced* afterwards. An additional “v” wave can be observed in atrial pressure during refilling with the closed AV valve. The total volume change during systole is called *stroke volume* (SV), and denotes the blood ejected during one cardiac cycle.

The *mechanical diastole* is defined as the interval between closure of the semilunar and closure of the AV valve. It can be subdivided into four phases: During *isovolumetric relaxation* AV valve is still closed, the ventricles relax due to decreasing contraction forces and decreasing overall stiffness, and the ventricular pressure decreases abruptly. It is followed by the phase of *early* or *rapid filling*. The AV valve opens and ventricular volume increases. First the ventricular pressure is below the atrial pressure and the ventricle sucks blood from atrium, then the atrial pressure follows the ventricular pressure (“y” descend). The third phase is the *diastasis* or *slow filling phase*, where the ventricles are almost filled with blood and experience only a slight volume increase. In the last phase, the atria contract and *late filling* occurs. The atrial pressure increases (“a” wave), the valve plane gets pulled towards the atria and ventricular volume increases since the atria actively pump blood into the ventricle. While for humans at rest the filling of the ventricles is mostly passive and atrial contraction only contributes 10-20% of the stroke volume, the passive filling phase shortens and the atria contribute up to 40% during exercise (Mitchell and Wang, 2014).

Pressure-volume loops (pV loops) are often used to visualize pressure against volume within a chamber in order to get a quick overview of diagnostic information. Figure 2.9 shows typical pV loops of the left ventricle including stroke volume, valve action events and systolic phases. For different loading conditions, a family of pV loops can be plotted. During the filling phase, it includes information about the passive properties of the chamber via the non-linear end-diastolic pressure volume relation (EDPVR). The linear end-systolic pressure volume relation (ESPVR) on the other hand tracks the pressure at the state of maximal contraction for a given volume. The slope elastance E_{ES} is an indicator of the stiffness in contracted state and thereby contains information about the active properties. Concerning the atrium, Figure 2.8 shows a pV loop including the

2. *Physiological Fundamentals*

complex interaction between atrial and ventricular contraction expressing in a typical double loop structure. During atrial contraction, volume decreases and pressure increases causing the “a” loop. During ventricular contraction, atrial volume increases and pressure increases due to passive filling. Ventricular relaxation finally causes the release of blood into the ventricle, seen by a decrease in atrial pressure and volume and expressed by the “v” loop.

2. Physiological Fundamentals

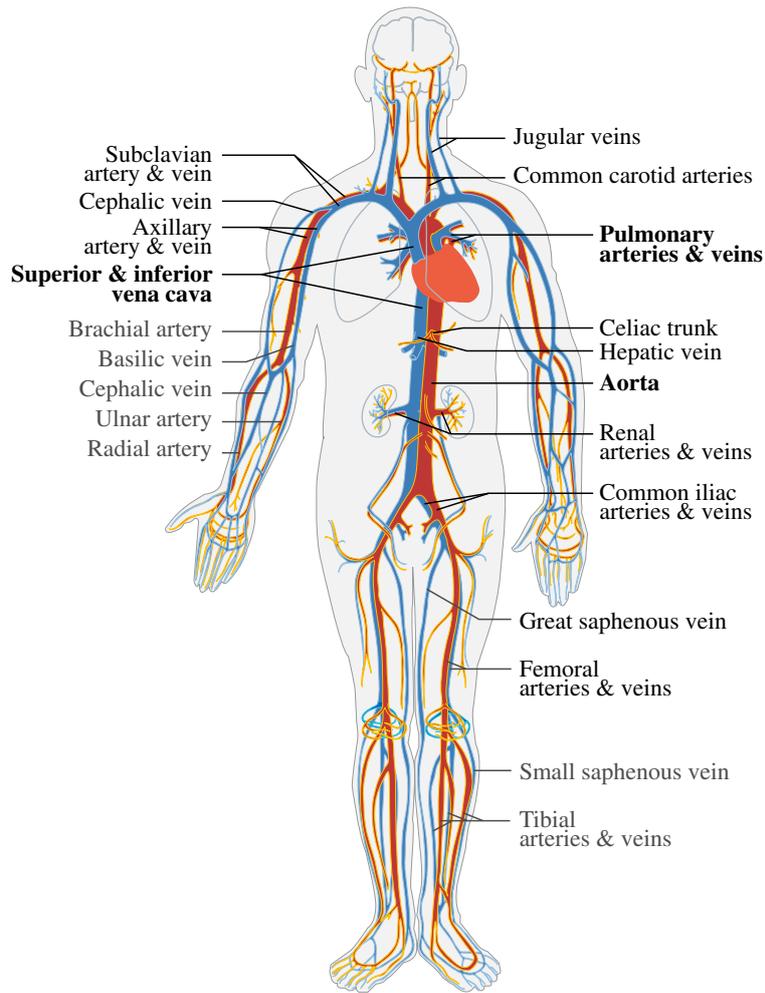


Figure 2.7.: Visualization of the human vascular system with arteries in red and veins in blue. Adapted from Wikimedia Commons (2016).

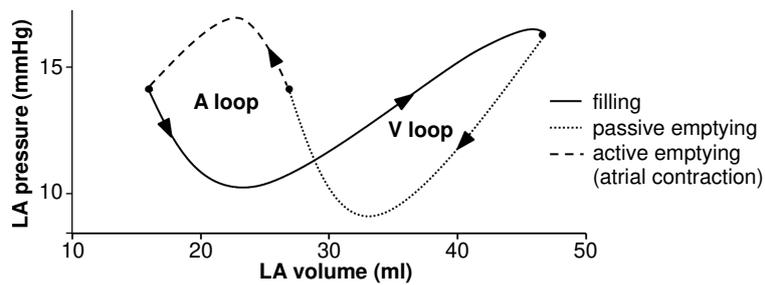


Figure 2.8.: A normal pV loop of the left atrium showing the characteristic double loop structure. Adapted from Weimar et al. (2012).

2. Physiological Fundamentals

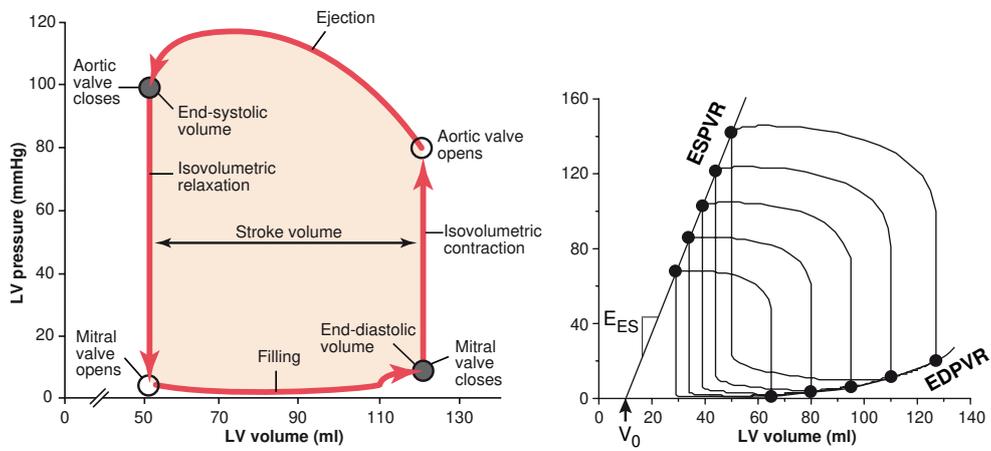


Figure 2.9.: A normal pV loop of the left ventricle with characteristic events of the heart cycle (left) and diagnostic values (right). Adapted from (Hall, 2015, fig. 9-10) and Burkhoff et al. (2005).

3. Mathematical Modeling

Modeling of cardiac motion is divided in two parts. Mathematical modeling describes the concepts of non-linear continuum mechanics that peak into the space-time-dependent governing equations. These can be solved using the method of finite elements for spatial discretization, and a generalized alpha method for temporal discretization. Various models for the multitude of present internal and external forces form the second part. An important aspect to the concept of continuum mechanics is the tissue-specific relationships between stress and strain, denoted as material law. Additionally, time-varying active forces are required to reflect the specific, non-static cardiac motion. The internal forces are complemented by mandatory boundary conditions that ensure mathematical uniqueness, and a number of optional boundary conditions, external surface forces that technically are not required to obtain a solution, but – if done correctly – vastly improve quality and validity of the solution. These are endocardial forces due to blood pressure from the circulatory system, and epicardial forces due to surrounding tissue and the rather restrictive pericardial sac the heart is embedded in. Contributing to the complexity, all of these forces are influenced by the cardiac deformation itself. Third, this chapter describes available methods to solve the discrepancy between a simulation-driven requirement to start with a stress-free geometry and the inherently not stress-free geometry obtained from in-vivo MRI imaging. Finally it completes with a description of the work by Fritz (2015), which lead to the systematic investigations of modeling approaches in this thesis.

3.1. Continuum Mechanics and Description of Deformation and Stress

Continuum mechanics is the theory of mathematically describing deformation of arbitrary objects due to internal and external forces. This chapter introduces the basic terms, definitions and concepts needed to describe the behavior of the soft tissue of the heart in the context of a computational elasticity simulation.

3.1.1. Stress vs Strain

An important aspect of modeling with continuum mechanics are the concepts of stress and strain. *Strain* denotes the part of deformation that ‘is visible’ from outside, by displacement of points on the surface (and within the material) in relation to a reference deformation. This is purely the actual deformation, without any respect of the inner forces. *Stress* denotes the inner forces that are either caused by a deformation (passive stress) or that causes a deformation (active stress). The relation between stress and

3. Mathematical Modeling

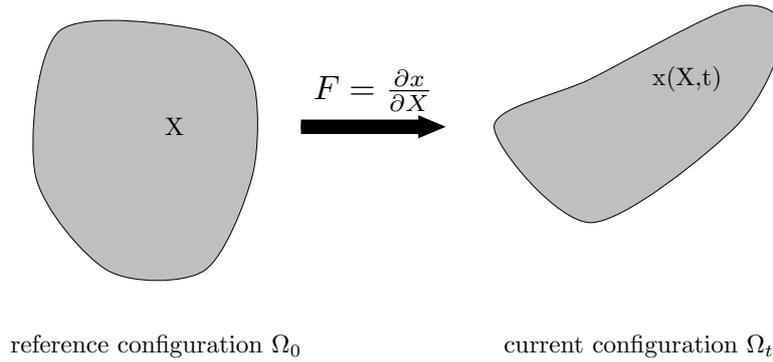


Figure 3.1.: The relationship between reference configuration X and the time-dependent current configuration x is given by the deformation tensor F .

strain is given for a certain material by an empirical relationship, called *material law* or *constitutive law*, which describes the reaction forces of a specified material to an imposed deformation. Two patches showing the same strain, can have very different inner forces depending on their material. A rather stiff piece of concrete or metal will produce a high stress already at a deformation of a few percent, a piece of soft tissue in contrast behaves fairly soft and will react with a lot lower forces by the material, or – when applying the same force – undergo a larger deformation.

In the human heart, the myocardial tissue is of a rather soft type and deformations are large. The enclosed ventricle volume changes during a normal heart beat roughly by 50 %, and the myocardial strain is accordingly high.

3.1.2. Measures of Deformation

For describing the current state of an object, a so-called reference configuration is defined. The *reference configuration* describes the undeformed state, mostly chosen at $t = 0$. The *current configuration* denotes the state after deformation at time t . The domain Ω is called Ω_0 in the reference configuration and Ω_t in the current configuration. The bijective movement function $\varphi : \Omega_0 \rightarrow \Omega_t, (X, t) \rightarrow \varphi(X, t)$ transforms a point's coordinates X in the reference configuration, denoted with a capital letter, to the deformed coordinates $x = \varphi(X, t)$ at time t . The *deformation tensor* F is a matrix describing the deformation of an infinitesimal small line element in the reference configuration dX in the current configuration, given by its components

$$\mathbf{F} = \frac{dx}{dX},$$

$$F_{i,j} = \frac{\partial F_i}{\partial X_j}, \quad i, j, k = 1, 2, 3.$$

The deformation of area and volume elements can be derived analogously.

3. Mathematical Modeling

Since the deformation tensor contains rotational components, it is rather unsuited to use it directly. A better measure that depends on deformation should not change under simple rotation or translation. For these applications, additional deformation measures exist and are introduced in the following. The left *Cauchy-Green Tensor* \mathbf{C} is given by

$$\mathbf{C} = \mathbf{F}^T \mathbf{F},$$

and the *Green-Lagrange strain tensor* \mathbf{E}

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - \mathbf{I})$$

with identity matrix \mathbf{I} . The Cauchy-Green tensor is rotational invariant, i.e. independent from the rotational component \mathbf{R} of the deformation tensor, which is an advantage over the deformation tensor. This can be seen by applying a polar decomposition $\mathbf{F} = \mathbf{R}\mathbf{U}$ and $\mathbf{C} = \mathbf{U}^T \mathbf{R}^T \mathbf{R} \mathbf{U} = \mathbf{U}^T \mathbf{U}$. The Green-Lagrange strain tensor has the additional property that it reflects an undeformed configuration by only zeros in it's components.

One important application of the deformation tensor in cardiac simulations is determining the change of fiber length. In its original coordinate system, the fiber is aligned with the local x-axis and has the orientation $f_0 = [1, 0, 0]$. In its deformed state, the fiber has a new orientation, given by $f_t = \mathbf{F}f_0$. The relative change of fiber length can be computed by

$$\lambda_f = \frac{\|f_t\|_2}{\|f_0\|_2} = \frac{\sqrt{(\mathbf{F}f_0)^T (\mathbf{F}f_0)}}{\sqrt{f_0^T f_0}} = \sqrt{f_0^T \mathbf{C} f_0}$$

and is also called the *stretch ratio*. This has the properties $\lambda_f = 1$ in the undeformed state, $\lambda_f > 1$ in positive stretched states and $\lambda_f < 1$ in shortened states. A similar value is the *engineering strain*

$$\varepsilon_f = \lambda_f - 1,$$

which is located around 0.

The second important application covers the computation of deformation energies. For formulation of isotropic material laws, the following characteristic ‘invariants’ of the Cauchy-Green tensor \mathbf{C} can be used, these do not depend on the choice of coordinate system:

$$\begin{aligned} I_1 &= \text{tr}(\mathbf{C}) &&= (\lambda_1)^2 + (\lambda_2)^2 + (\lambda_3)^2, \\ I_2 &= \frac{1}{2}((\text{tr}(\mathbf{C}))^2 - \text{tr}(\mathbf{C}^2)) &&= (\lambda_1 \lambda_2)^2 + (\lambda_2 \lambda_3)^2 + (\lambda_1 \lambda_3)^2, \\ I_3 &= \det(\mathbf{C}) &&= (\lambda_1 \lambda_2 \lambda_3)^2, \end{aligned}$$

3. Mathematical Modeling

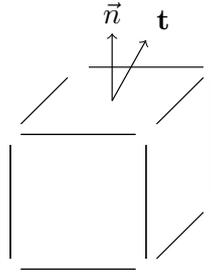


Figure 3.2.: The Cauchy stress tensor illustrated by a small cube. On the intersection area with normal \vec{n} acts the traction force \mathbf{t} .

whereby here the eigenvalues of the deformation tensor \mathbf{F} are denoted by $\lambda_{1,2,3}$. Anisotropic material laws however are designed to depend on the coordinate system, these need to use the entries of the Green-Lagrange tensor directly.

The third application is the measure of volume change, which can be incorporated by a penalty formulation. The volume change $\frac{dv}{dV}$ can be derived by $dV = dX_1 dX_2 dX_3$ and the corresponding deformations $dx_{1,2,3} = \mathbf{F}dX_{1,2,3}$ and is given by

$$\frac{dv}{dV} = \det \mathbf{F} = J = \sqrt{I_3}.$$

A simple possible formulation penalizing volume change is $K(J - 1)^2$ with a given penalty factor K .

3.1.3. Measures of Stress

Stress describes the inner forces, that occur inside a material. It is defined as the forces on the faces of an infinitesimal small unit cube. These forces do not necessarily need to be orthogonal to surface. Thereby the unit of stress is N/m or Pa.

The *Cauchy stress tensor* σ gives the stress in relation to the current configuration and is thereby also known as real or physical stress:

$$\sigma = (\mathbf{t}_1 \mathbf{t}_2 \mathbf{t}_3)$$

where $\mathbf{t}_{1,2,3}$ are the traction vectors corresponding to the unit surfaces. The relationship between Cauchy stress and force on an arbitrarily oriented surface with normal \mathbf{n} is linear:

$$\mathbf{t}_n = \sigma \mathbf{n}$$

In general, the stress vector on a surface with orientation \mathbf{n} is defined as $\mathbf{t}(\mathbf{n}) = \lim_{\Delta A \rightarrow 0} \frac{\Delta \mathbf{f}}{\Delta A} = \frac{d\mathbf{f}}{dA}$. Due to conservation of angular momentum, the stress tensor is symmetric ($\sigma = \sigma^T$).

3. Mathematical Modeling

Table 3.1.: Conversion formula between the different stress tensors.

	Cauchy	PK1	PK2
$\sigma =$	–	$J^{-1}\mathbf{P}\mathbf{F}$	$J^{-1}\mathbf{F}\mathbf{S}\mathbf{F}^T$
$\mathbf{P} =$	$J\sigma\mathbf{F}^{-1}$	–	$\mathbf{S}\mathbf{F}^T$
$\mathbf{S} =$	$J\mathbf{F}^{-1}\sigma\mathbf{F}^{-T}$	$\mathbf{P}\mathbf{F}^{-T}$	–

For the implementation of a finite element code, it is useful to measure stress related to the undeformed state. Therefore additional stress measures exists, which can be converted from one to another using the formula given in Table 3.1.

The unsymmetric *first Piola-Kirchhoff tensor* (PK1) \mathbf{P} measures the forces in the current configuration and the surface in the reference configuration:

$$\mathbf{P} = \frac{\mathbf{f}}{A_0}$$

The symmetric *second Piola-Kirchhoff stress* (PK2) \mathbf{S} measures force as well as area in the reference configuration:

$$\mathbf{S} = \frac{\mathbf{f}_0}{A_0}$$

3.2. Governing Equations

The behavior of non-linear continuum mechanics is dominated by the three conservation laws for mass, linear momentum and angular momentum.

1. Conservation of mass:

$$\rho_0 = J\rho, \quad J = \det(\mathbf{F}) \quad (3.1)$$

2. Balance of linear momentum:

$$\rho \mathbf{a} - \operatorname{div} \sigma - \rho \mathbf{b} = 0 \quad (3.2)$$

with acceleration forces \mathbf{a} , body forces \mathbf{b} and a hyperelastic material law that relates deformation \mathbf{E} to an energy potential $W(\mathbf{E})$, its derivative gives path-independent traction forces $\mathbf{S} = dW(\mathbf{E})/d\mathbf{E}$.

3. Conservation of angular momentum, which imposes symmetry on the second Piola-Kirchhoff tensor \mathbf{S} :

$$\mathbf{S} = \mathbf{S}^T \quad (3.3)$$

3. Mathematical Modeling

These equations together are called the *equilibrium equations* since they describe a state at which all forces at all points are in balance for the whole domain.

Additionally, a set of boundary conditions defining the behavior on Γ is needed to obtain uniqueness. These are usually one of or a combination of

- *displacement boundary condition*, where displacements u are prescribed, or
- *surface traction boundary condition*, where surface forces \mathbf{t} are defined

on parts of the domain. Examples are node fixations for the former and surface pressures for the latter.

3.3. Spatial Discretization

Outgoing from the equilibrium equations in section 3.2, a linearized matrix formulation, solvable with a computer, is needed. For spatial discretization, the equilibrium equations are discretized with the “principle of virtual work”, by multiplication with test functions w_I or “virtual displacements”. The remainder gives the total change of energy in the system δW . The strong formulation of the linear momentum conservation equation (3.2) gets first discretized with ansatz functions N_I and from there a weak formulation is derived by multiplication with arbitrary test functions w (this known as principle of virtual work in the context of engineering mechanics) and further discretization of w . In a Galerkin approach, the same space is used for ansatz and test functions. Tetrahedra of first and second order were used, the latter show less volumetric locking. These are denoted according to their number of nodes as T4 or T10 elements and shown in Figure 3.3.

Starting with Newton’s second law of motion in its integral representation, the equation

$$\int_{\mathbf{x} \in \Omega} \rho \mathbf{a}(\mathbf{x}, t) d\Omega = \int_{\mathbf{x} \in \partial\Omega} \mathbf{t}(\mathbf{n}, \mathbf{x}, t) d\Gamma + \int_{\mathbf{x} \in \Omega} \rho \mathbf{b}(\mathbf{x}, t) d\Omega$$

needs to be fulfilled by the acceleration forces $\rho \mathbf{a}(\mathbf{x}, t)$ in any deformed domain Ω , with surface tractions $\mathbf{t}(\mathbf{n}, \mathbf{x}, t)$ and body force density $\mathbf{b}(\mathbf{x}, t)$.

With the approximations $u \approx \sum u_{iI} N_I$, $w \approx w_{iI} N_I$ and transformation to the reference domain Ω_0 (this introduces the first Piola Kirchhoff stress tensor P), this leads to the *discrete equations of the weak form of the total Lagrangian formulation*:

$$\delta W = w_{iI} \int_{\Omega_0} \rho_0 N_I N_J \ddot{u}_{jJ} d\Omega_0 + w_{iI} f_{iI}^{int} + w_{iI} f_{iI}^{ext} = 0$$

3. Mathematical Modeling

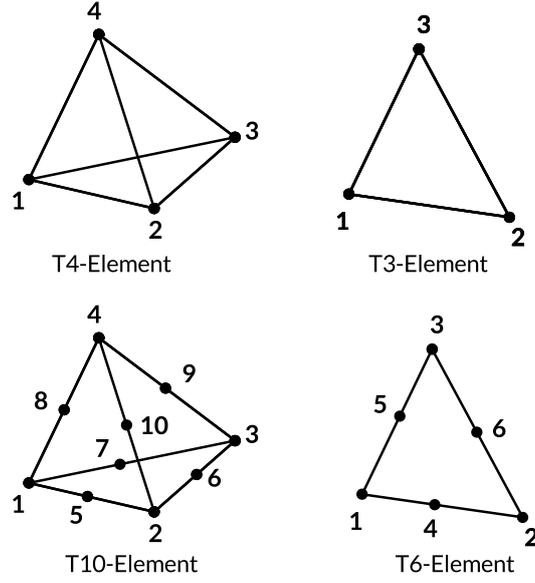


Figure 3.3.: Node numbering of a linear surface element (T3), a quadratic surface element (T6), a linear tetrahedron (T4) and a quadratic tetrahedron (T10). From Fritz (2015).

where $f_{iI}^{int} = \int_{\Omega_0} \frac{\partial N_I}{\partial X_j} P_{ji} d\Omega_0$ and $f_{iI}^{ext} = \int_{\Omega_0} N_I \rho_0 b_i d\Omega_0 + \int_{\Gamma_0} N_I t_i^0 d\Gamma_0$. This holds true for arbitrary w_i , so it follows:

$$\underbrace{\int_{\Omega_0} \frac{\partial N_I}{\partial X_j} P_{ji} d\Omega_0}_{\mathbf{f}^{int}} - \underbrace{\left(\int_{\Omega_0} N_I \rho_0 b_i d\Omega_0 + \int_{\Gamma_0^0} N_I t_i^0 d\Gamma_0 \right)}_{\mathbf{f}^{ext}} + \underbrace{\int_{\Omega_0} \rho_0 N_I N_J d\Omega_0}_{\mathbf{M}} \ddot{u}_{jJ} = 0$$

With internal forces \mathbf{f}^{int} , often called *stiffness matrix*, external forces \mathbf{f}^{ext} , and mass matrix \mathbf{M} this can be written in the abbreviated form:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{f}^{int}(\mathbf{u}, t) - \mathbf{f}^{ext}(\mathbf{u}, t) = 0 \quad (3.4)$$

External forces are influenced by boundary conditions, e.g. forces on the myocardial surface due to blood pressure on endocardium or penalty forces to ensure frictionless contact on epicardium. Since the nodal forces f^{int} (c.f. Section 3.6) and f^{ext} (c.f. Section 3.7) depend non-linearly on \mathbf{u} , the arising system is solved iteratively using a Newton method.

In the implementation, this equilibrium of forces is extended by a damping matrix \mathbf{C} and writes in matrix notation as

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{f}^{int}(\mathbf{u}, t) - \mathbf{f}^{ext}(\mathbf{u}, t) = 0 \quad (3.5)$$

3. Mathematical Modeling

with \mathbf{u} containing all coordinates of all points. The dependency of \mathbf{f}^{int} and \mathbf{f}^{ext} on \mathbf{u} makes the system non-linear. For hyperelastic materials, the passive forces in \mathbf{f}^{int} are path-independent and “memoryless”, they depend only on the current deformation state \mathbf{u} but not on past ones. A temporally changing dependency between \mathbf{f}^{int} and \mathbf{u} is introduced by active contractile forces.

One possible choice for the damping matrix is *Rayleigh damping*, which introduces damping depending on mass and passive stiffness (internal nodal forces):

$$\mathbf{C} = \alpha_1 \mathbf{M} + \alpha_2 \mathbf{K}$$

with the linearization of the passive forces

$$\mathbf{f}^{int}(\mathbf{u}) \approx \mathbf{K} \cdot \mathbf{u}, \quad \mathbf{K} := \nabla_{\mathbf{u}} \mathbf{f}^{int}(\mathbf{u})$$

using the stiffness matrix \mathbf{K} .

Since the internal forces \mathbf{f}^{int} depend non-linearly on u and are influenced by active contractile forces as well, Raleigh damping has the nice property of being an “adaptive” damping during the cardiac cycle, however it requires re-computation of \mathbf{K} in each step. During systole, the increasing active contraction forces lead to an increasing amount of damping. In that situation, the eigenfrequency of the tissue is large (active and stored passive stress is high) and the corresponding damping matrix as well ($\nabla_{\mathbf{u}} \mathbf{f}^{int}$). Later on, quickly decreasing active forces however might become problematic at the beginning of relaxation: A high amount of stored passive energy and quickly decreasing active forces cause a high acceleration and requires a comparatively large amount of permanent damping via α_1 to remain computable. It is difficult to choose the parameter α_2 to have sufficiently enough adaptive damping during the relaxation, but not too much during contraction. For some parameter combinations it could be observed that the system begins to oscillate heavily and becomes unstable. Lowering β helped in some of these cases. Typical values used in the simulations are $\alpha = 1000$ and $\beta = 0.01$, or $\beta = 0.005$ in those problematic cases¹.

Rajagopal et al. (2007) give a good brief overview of the general discretization approach, a more thorough description can be found in Belytschko et al. (2014) for the theory and Fritz (2015) for the implementation of the equations.

3.4. Temporal Discretization

In order to implement equation (3.5), the temporal derivatives $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ need to be discretized as well. Given the current state of all node positions, this gives us the next state of all node positions and thereby secondary values like velocity, acceleration, deformation as well. Throughout this section, displacement increments \tilde{d} are used (which is change of node coordinates relative to previous time step), instead of the displacements

¹This was observed in a simulation with only left ventricle and left atrium, fixed at apex and atrial orifices, only passive and prescribed active forces, but neither blood pressure nor pericardium modeling. The same setup with a four chamber geometry worked fine though.

3. Mathematical Modeling

\mathbf{u} itself (which is relative to the undeformed state). Computation-wise, this has the advantage of better numerical accuracy. Two different solvers are used in this work.

The first solver is the called the *static* or *equilibrium* solver. It omits all time derivatives in the differential equations and yields a steady-state solution under the respective boundary conditions for the current time step (active tension, blood pressure) without depending on previous solutions:

$$\mathbf{r}(\mathbf{d}) := \mathbf{f}^{int}(\mathbf{d}) - \mathbf{f}^{ext}(\mathbf{d}) = \mathbf{0}$$

In that case, time integration is not necessary and the nonlinear system of equation is solved using Newton's method by linearizing the deformation- and potentially time-dependent terms $\mathbf{f}^{int}(\mathbf{d})$ and $\mathbf{f}^{ext}(\mathbf{d})$ in each Newton step. In theory, the solution \mathbf{d} does not depend on previous steps. In the implementation however, the node displacements could be found faster and more reliable by reusing the displacement increments from the previous time step as initial guess of Newton's method for the current time. In case of a changing time step size, it needs to be adapted accordingly. This effectively corresponds to a constant-velocity extrapolation from the previous time step as initial guess and improves convergence behavior especially for larger time step sizes.

The *Newmark-beta* solver is the second solver and respects all time derivatives in the momentum equation. It additionally tracks velocity \mathbf{v} and acceleration \mathbf{a} to improve the initial guess for the displacement \mathbf{d} in next time step (Belytschko et al., 2014). This method is designed for elasticity problems and is a special case of the generalized alpha method introduced by Chung and Hulbert (1993). Depending on the choice of the algorithm parameters β and γ , the Newmark-beta method represents an explicit or an implicit method and can introduce additional numerical damping. In contrast to the equilibrium solver, including time derivatives now reflects effects due to mass inertia and damping over the matrices \mathbf{M} and \mathbf{C} :

$$\mathbf{r}(\mathbf{d}) := \mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{f}^{int}(\mathbf{d}, t) - \mathbf{f}^{ext}(\mathbf{d}, t) = \mathbf{0}$$

At t^{n+1} , the following system of nonlinear equations is solved iteratively for \mathbf{d}^{n+1} using Newton's method:

$$\mathbf{r}(\mathbf{d}^{n+1}) = \mathbf{M}\mathbf{a}^{n+1} + \mathbf{C}\mathbf{v}^{n+1} + \mathbf{f}^{int}(\mathbf{d}^{n+1}, t^{n+1}) + \mathbf{f}^{ext}(\mathbf{d}^{n+1}, t^{n+1}) \stackrel{!}{=} \mathbf{0} \quad (3.6)$$

The update formula are given by (Belytschko et al. (2014), sec 6.3.3, p 339):

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2} \left((1 - 2\beta) \mathbf{a}^n + 2\beta \mathbf{a}^{n+1} \right), \quad (3.7)$$

and with the definitions

$$\tilde{\mathbf{d}}^{n+1} := \mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2} (1 - 2\beta) \mathbf{a}^n, \quad (3.8)$$

$$\tilde{\mathbf{v}}^{n+1} := \mathbf{v}^n + (1 - \gamma) \Delta t \mathbf{a}^n \quad (3.9)$$

3. Mathematical Modeling

for acceleration and velocity by

$$\mathbf{a}^{n+1} = \frac{1}{\beta\Delta t^2}(\mathbf{d}^{n+1} - \tilde{\mathbf{d}}^{n+1}) \quad (3.10)$$

$$= \frac{1}{\beta\Delta t^2} \left(\mathbf{d}^{n+1} - (\mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2}(1 - 2\beta)\mathbf{a}^n) \right), \quad (3.11)$$

$$\mathbf{v}^{n+1} = \tilde{\mathbf{v}}^{n+1} + \gamma\Delta t \mathbf{a}^{n+1} \quad (3.12)$$

$$= (\mathbf{v}^n + (1 - \gamma)\Delta t \mathbf{a}^n) + \gamma\Delta t \mathbf{a}^{n+1} \quad (3.13)$$

$$= \mathbf{v}^n + \Delta t \left((1 - \gamma) \mathbf{a}^n + \gamma \mathbf{a}^{n+1} \right). \quad (3.14)$$

Inserting these in equation (3.6) leads to the following system of nonlinear equations that depends only on \mathbf{d}^{n+1} , but not on \mathbf{a}^{n+1} and \mathbf{v}^{n+1} anymore:

$$\begin{aligned} \mathbf{r}(\mathbf{d}^{n+1}) = & \mathbf{M} \left(\frac{1}{\beta\Delta t^2} \left(\mathbf{d}^{n+1} - (\mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2}(1 - 2\beta)\mathbf{a}^n) \right) \right) \\ & + \mathbf{C} \left((\mathbf{v}^n + (1 - \gamma)\Delta t \mathbf{a}^n) + \gamma\Delta t \frac{1}{\beta\Delta t^2} \left(\mathbf{d}^{n+1} - (\mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{2}(1 - 2\beta)\mathbf{a}^n) \right) \right) \\ & + \mathbf{f}^{int}(\mathbf{d}^{n+1}, t^{n+1}) - \mathbf{f}^{ext}(\mathbf{d}^{n+1}, t^{n+1}) \\ = & \mathbf{0} \end{aligned}$$

The Jacobian of \mathbf{r} is given by

$$\mathbf{A} = \frac{1}{(\Delta t)^2\beta} \mathbf{M} + \frac{\gamma}{(\Delta t)\beta} \mathbf{C} + \mathbf{K}$$

whereby internal and external forces \mathbf{K} are linearized in each Newton iteration

$$\mathbf{K}\mathbf{d}^{n+1} \approx \mathbf{f}^{int}(\mathbf{d}^{n+1}, t^{n+1}) - \mathbf{f}^{ext}(\mathbf{d}^{n+1}, t^{n+1}).$$

A good initial guess for Newton's method is to use the approximated displacement increments $\tilde{\mathbf{d}}^{n+1}$ from (3.8) as proposed by Hughes et al. (1979). Finally, \mathbf{d}^{n+1} is known and \mathbf{a}^{n+1} , \mathbf{v}^{n+1} can be updated using (3.11) and (3.14).

Initial values for both solvers are chosen such that $v_0 = a_0 = 0$ everywhere and the initial node positions correspond to the node positions in the reference state for $t \leq 0$. Regarding the choice of algorithm parameters, this method corresponds for $\beta = 0$, $\gamma = \frac{1}{2}$ to the explicit central difference method, for $\beta = \frac{1}{4}$, $\gamma = \frac{1}{2}$ to the undamped trapezoidal rule and for $\gamma > \frac{1}{2}$ to a numerically damped integrator with damping proportional to $\gamma - \frac{1}{2}$. All simulations in this thesis used the parameter set $\beta = 0.5$ and $\gamma = 1$.

3.5. Force Modeling Overview

Various forces dominate cardiac motion during each phase of the cardiac cycle. Figure 3.4 gives an overview of the main force components.

3. Mathematical Modeling

For each component exist different complexity levels of mathematical modeling. While more complexity generally adds more accuracy to a specific behavior, it usually has the drawback of more parameters that are more difficult and less reliable to determine. Moreover, most of these models loose comprehensibility of each of the involved parameters, combined with more subtle limitations that are harder to understand. This increases the danger of using them in a context they were not designed for, and therefore are not able to show the expected qualitatively improved behavior. Opposite from what was actually intended, the produced results are even worse than when using simpler models with clear but fulfilled limitations.

The main contributors to the mechanical cardiac function are the following forces. *Active forces* are directly related to the contraction of muscle fibers in the myocardium. The range of model complexity starts at simple, explicit functions adapted to measurement data, that can be computed in a fraction of a millisecond. The other end on the complexity scale are sophisticated differential equation-based models of ion channels in the cell membranes, currents through them, gap junctions in-between separate myocardial cells together with Maxwell equations for the propagation. The calcium concentration in each cell is additionally paired with a model of sarcomere interaction including the current state of deformation, finally giving the force value of each single computational cell. The *passive forces* are counteracting active forces, representing material properties like the softness, volume conservation or memory-dependency of the tissue. An other type of internal forces are introduced by the dynamic behavior of the tissue, the *dynamic forces* comprise forces due to damping and mass inertia. The remaining two sources of forces are related to external forces due to the surrounding of the heart. The *endocardial forces* act on the blood-oriented side of the myocardium and are caused by blood pressure. The *epicardial forces* are also surface forces, but act on the outwards-pointing surface and summarize all forces created by (passive) motion of the surrounding like neighboring organ tissue or the pericardial sac.

While computational simulations without endocardial or epicardial surface forces are perfectly possible, active forces are needed to generate the motion of the heart, and passive forces generate the reactive forces and limit the motion as stated by Newton third law of action and reaction.

3.6. Modeling of Passive Stress

Up to here, stress and strain can be seen as completely independent, in reality however this is not the case. The relationship strongly depends on the material and its properties like stiffness, plasticity or an anisotropic structure. The relationship between the two describes a *constitutive* or *material law*. In *hyperelastic* material laws, the stress depends only on the current state of deformation and has no memory or path-dependency. Stress can then be derived from a potential describing the deformation energy, called an *energy*

3. Mathematical Modeling



Figure 3.4.: Complexity diagram of the different force components involved in modeling the motion of the heart. Color intensity and number of model parameters increase with model complexity.

3. Mathematical Modeling

density function $W(\mathbf{E})$:

$$\mathbf{S}(\mathbf{E}) = \frac{dW}{d\mathbf{E}},$$

$$S_{ij} = \frac{dW}{dE_{ij}}, \quad i, j = 1, 2, 3$$

Anisotropy of a material is the property of a stress tensor depending on preferred directions of deformation. If the behavior is in two directions the same, the material is called *bi-isotropic* or *transversal isotropic*. Myocardial tissue has a filamentous micro-structure and has an anisotropic passive behavior.

In this work, two constitutive models are used, both are non-linear and hyperelastic. The *Mooney-Rivlin material* is used for the atria and is a simple but commonly used isotropic material law. The energy density function is

$$W = C_1(I_1 - 3) + C_2(I_2 - 3)$$

with material parameters C_1 , C_2 and the invariant I_1, I_2 of Cauchy-Green tensor \mathbf{E} . In the current code, a more general formulation is implemented

$$W = c_{10}(I_1 - 3) + c_{01}(I_2 - 3) + c_{20}(I_1 - 3)^2 + c_{02}(I_2 - 3)^2 + c_{11}(I_1 - 3)(I_2 - 3).$$

The transversely isotropic *Guccione material law* is used for the ventricles, since it is a material law specifically designed for ventricular tissue (Guccione et al., 1991, 1995). Its energy density function is given by:

$$W = C \left(e^{Q(\mathbf{E})} - 1 \right)$$

with

$$Q(\mathbf{E}) = b_1 E_{11}^2 + b_2 (E_{22}^2 + E_{33}^2 + E_{23}^2 + E_{32}^2) + b_3 (E_{12}^2 + E_{21}^2 + E_{13}^2 + E_{31}^2)$$

and the four model parameter C , b_1 , b_2 and b_3 .

A number of similar material laws for soft tissue was developed by other authors as well, all of which having an exponential relationship in common (Fung, 1965; Fung and others, 1993). The Costa material law uses six parameters to additionally respect anisotropic behavior between sheet and sheetnormal directions (Costa et al., 2001). The Holzapfel material uses an own exponential term for each component of \mathbf{E} (Holzapfel and Ogden, 2009). Schmid et al. (2008) give a nice comparison between different laws for soft tissue.

To realize a penalization of volume change, both laws were extended by an additional penalty term

$$0.5b \ln(I_3)^2 \quad (\text{Mooney-Rivlin})$$

$$0.5K(J - 1)^2 \quad (\text{Guccione})$$

3. Mathematical Modeling

Table 3.2.: Different parameters for myocardial heart tissue found in literature. Since at least two versions exist of the Guccione energy potential exists ($W = Ce^{(\dots)}$ and $W = \frac{C}{2}e^{(\dots)}$), the ‘pre-factor’ denotes the scaling factor in front of the exponent. Costa law actually has seven parameters, but can be ‘reduced’ to be used with the Guccione formula. The pre-factor is given in Pa and the shape-factors $b_{1,2,3}$ are unitless.

pre-factor	b1	b2	b3	source, subject type	
C	1200	26.7	2	14.7	Omens1993, dog
C	831	14.3	4.48	0.762	Nash2000/Wang2009, canine
C	876	18.48	3.58	1.627	McCulloch1989, canine
C/2	57.5	14.4	5.76	10.08	$T_{max} = 143$ kPa Genet2014, in-vivo human
C/2	94.5	28.9	13.5	13.3	6 pig hearts, Xi2011 (actually Costa parameters estimated by schmid2008, Dokos measurements)

with penalty parameters b and K , respectively, and the third invariant of the Cauchy-Green tensor $I_3 = \det(\mathbf{C}) = (\det(\mathbf{F}))^2 = J^2$ indicating the relative volume. It has to be noted that in reality it can be observed that the volume of myocardial tissue is not constant, moreover it is compressible and changes in the range of ten percent during one heart cycle due to blood perfusion and other factors (Waldman et al. (1985); Ashikaga et al. (2008)). So an exact volume conservation (e.g. by Lagrange multipliers) is neither needed nor realistic.

Different parameters for the Guccione law were used in literature. An overview is given in Table 3.2.

3.7. Modeling of Active Stress

Modeling active stress is probably the most important part to reproduce the dynamic behavior of the heart, since these are the forces that trigger the motion. The active stress depends on fiber direction and is incorporated by adding an additional contribution to the second Piola-Kirchhoff stress tensor:

$$S = \frac{dW(\mathbf{E})}{d\mathbf{E}} + \begin{pmatrix} T(t, \mathbf{E}) & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (3.15)$$

The first summand represents the impact of passive force due to deformation, the second part the impact of active force due to contraction forces.

The active tension T may depend on time t and deformation expressed by \mathbf{E} . Strain in fiber direction λ_f is expressed by $\lambda_f = \left\| \mathbf{F} \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T \right\|_2$.

In this work, three different tension models are used.

- An *electrophysiological tension model* without stretch dependency, based on cell models by Fritz et al. (2014); Fritz (2015),

3. Mathematical Modeling

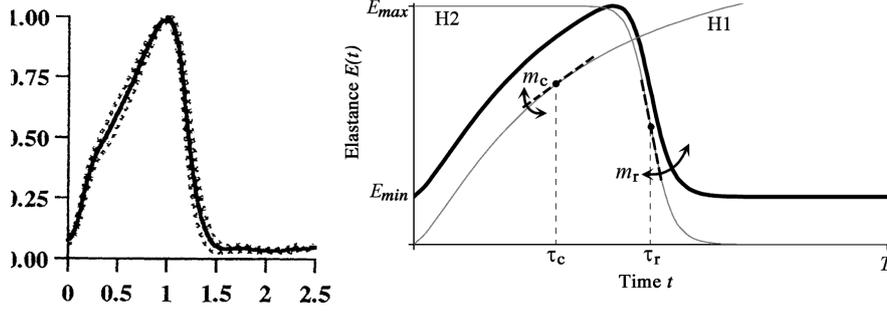


Figure 3.5.: Chamber elastance of the left ventricle as measured by Senzaki et al. (1996) in normal humans and defined by Mynard and Smolich (2015) (left), and construction of the DoubleHill function by Mynard (2011) (right). Taken from Schuler (2016).

- a *Double Hill tension model*, without stretch dependency Stergiopoulos et al. (1996), and the
- *Lumens tension model* with stretch dependency (Lumens et al., 2009).

A second tension model with stretch dependency by Land et al. (2017) was implemented as well, but not essential to the findings in this thesis.

Electrophysiological tension model For the tension development based on electrophysiological excitation, the *mono-domain equation* is solved for transmembrane voltages V_m using a separate simulation framework *acCELLerate* as described in Fritz et al. (2014); Fritz (2015):

$$\nabla \cdot (\sigma_i \nabla V_m) = (\kappa + 1) \beta \left(C_m \frac{dV_m}{dt} + I_{ion}(V_m) \right)$$

with model parameters σ_i , κ , β , C_m and transmembrane ion current density I_m . The *ten Tusscher model* (ten Tusscher et al., 2004) in the ventricles and the *Courtemanche model* (Courtemanche et al., 1998) in the atria yield the contributions to the transmembrane ion current density I_{ion} on right hand side. The intracellular calcium concentration arising from the cell models is transformed to a time-dependent force-per-cell value using the hybrid tension development model developed by Sachse et al. (2003). That value is then taken as static time- but not deformation-dependent input $T(t, \mathbf{E}) = T(t)$ to (3.15). Since in that case the geometry did not change over time, the developed tension does not depend on the current deformation.

DoubleHill tension model The *Double-Hill tension model* is based on an analytical function replicating the measurable ventricular elastance over time

$$E(t) = \frac{P(t)}{V(t) - V_0}$$

3. Mathematical Modeling

Table 3.3.: Parameters for the DoubleHill tension model. The rates m_c and m_r are unitless, τ_c , τ_r , t_0 and T are given in s.

	Mynard and Smolich (2015)		Reduced relaxation Ventricles
	Ventricles	Atria	
Contraction rate const. m_c	1.32	1.99	12.0
Relaxation rate m_r	21.9	11.2	
Contraction time offset τ_c	$0.269 \cdot T = 0.215$	$0.053 \cdot T = 0.042$	
Relaxation time offset τ_r	$0.452 \cdot T = 0.362$	$0.173 \cdot T = 0.138$	
Onset time t_0	0.15	0.0	
Period T	0.8	0.8	

by a normalized elastance

$$e(t) = \frac{E(t) - E_{min}}{E_{max} - E_{min}} = \frac{1}{k} \left(\frac{g_c}{1 + g_c} \right) \left(\frac{1}{1 + g_r} \right)$$

with

$$g_c = \left(\frac{t'}{\tau_c} \right)^{m_c}, \quad g_r = \left(\frac{t'}{\tau_r} \right)^{m_r}, \quad t' = \text{mod}(t - t_0, T), \quad k = \max(k, e(t))$$

and pressure-free volume V_0 (Stergiopoulos et al., 1996). Mynard and Smolich (2015) propose to apply tension based on such an elastance

$$T(t) = T_{max} e(t)$$

also for the atria. Parameters for ventricles and atria are given in Table 3.3. Justification of such a model is, that the contraction is dominated by the active forces. Using a tension following a specific time course imposes a specific time course of elastance as depicted in figure 3.5 as well. When coupling with a system of the circulatory system, all that circulation “sees” are the chamber elastances and since these are imposed based on measurements, the chambers in the finite element model “behave correctly” from the viewpoint of the circulatory system.

Lumens tension model The *Lumens tension model* in contrast is a phenomenological model directly based on measurements of the length-dependent tension in myocardial cells (Lumens et al., 2009). It respects the current length of the muscle cell, which makes it interesting because of a possible reproduction of the Starling effect. The original formulation of the model is triggered by a prescribed function $F_{rise}(t)$ that computes the activation C , a parameter that is “physiologically related to intracellular calcium concentration”. Figure 3.6 shows its time course. Hence calcium concentration effectively is assumed to follow the same time course in all cells. The original formula are:

3. Mathematical Modeling

Table 3.4.: Parameters for the tension model from Lumens et al. (2009). The value for C_{rest} was needed to be set to 0 in the 3D FEM simulations, in order to get a stress-free initial state.

Symbol	Unit	Value	Description
C_{rest}	--	0.02	diastolic resting level of activation
L_{sc0}	μm	1.51	Contractile element length with zero active stress
$L_{s,ref}$	μm	2.0	Reference sarcomere length at zero strain
$L_{se,iso}$	μm	0.04	Length of isometrically stressed series elastic element
v_{max}	μs^{-1}	7	Sarcomere shortening velocity with zero load
τ_D	ms	32	Factor for scaling time of contraction decay
τ_R	ms	48	Factor for scaling time of contraction rise
τ_{sc}	ms	425	Factor for scaling duration of contraction
σ_{act}	kPa	120	Factor for scaling active myofiber stress

$$\begin{aligned}
 F_{rise} &= 0.02x^3(8-x)^2e^{-x} \\
 &\text{with } x = \min(8, \max(0, t/\tau_R)) \\
 C_L(L_{sc}) &= \tanh\left(4.0(L_{sc} - L_{sc0})^2\right) \\
 T(L_{sc}) &= \tau_{sc}(0.29 + 0.3L_{sc}) \\
 \frac{dC}{dt} &= \frac{1}{\tau_R}C_L(L_{sc})F_{rise}(t) + \frac{1}{\tau_D} \cdot \frac{C_{rest} - C}{1 + e^{(T(L_{sc})-t)/\tau_D}} \\
 \frac{dL_{sc}}{dt} &= \left(\frac{L_s - L_{sc}}{L_{se,iso}} - 1\right)v_{max} \\
 \sigma_{f,act} &= \sigma_{act} C \cdot (L_{sc} - L_{sc0})\frac{L_s - L_{sc}}{L_{se,iso}}
 \end{aligned}$$

Parameters for the Lumens model are given in Table 3.4.

In contrast to the original paper, however some of the constants inside the formula needed to be converted such that the units represent time and space with m and s instead of μm and ms, respectively. Affected are the formula for C_L (4E12 instead of 4), T (0.3E6 instead of 0.3) and $\sigma_{f,act}$ (the whole formula needs to be scaled with 1E6). Moreover, for the 3D FEM simulations C_{rest} was set to 0 for being able to start in an equilibrium state with zero force (otherwise start of the simulation is not possible at all). The formula in the implementation are taken from the ‘‘appendix B’’ of the publication by Lumens et al. (2009) and compute the force $\sigma_{f,act}$ from strain ε_f :

3. Mathematical Modeling

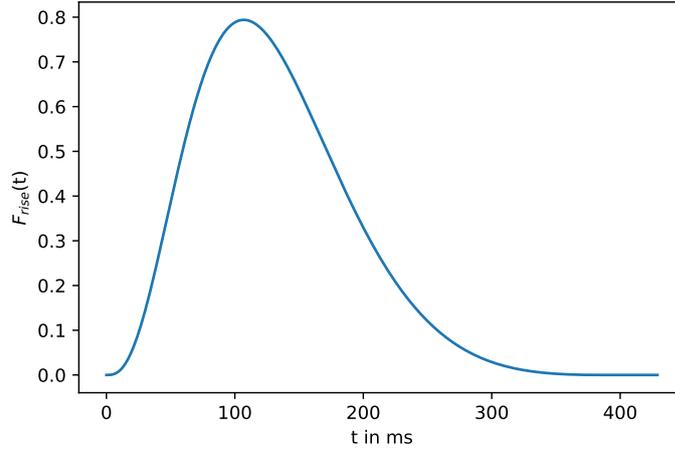


Figure 3.6.: Plot of F_{rise} , the invariable “rise of mechanical activation” as used in the tension model by Lumens et al. (2009).

$$L_s(\varepsilon_f) = L_{s,ref} e^{(\varepsilon_f)}$$

strain to sarcomere length,

$$F_{rise}(t) = 0.02 x^3 (8.0 - x)^2 e^{-x}$$

with $x = \min(8.0, \max(0.0, t/\tau_R))$,
and t denoting the time relative to the current heart beat

$$\frac{dL_{sc}}{dt}(L_{sc}, \varepsilon_f) = \left(\frac{L_s(\varepsilon_f) - L_{sc}}{L_{se,iso}} - 1 \right) v_{max}$$

$$C_L(L_{sc}) = \tanh \left(4 \cdot 1E12 \cdot (L_{sc} - L_{sc0})^2 \right)$$

1E12: needed if L_{sc} , L_{sc0} passed in SI units

$$T_{fun}(L_{sc}) = \tau_{sc} (0.29 + 0.3 \cdot 1E6 \cdot L_{sc})$$

1E6: needed if L_{sc} passed in SI units

$$\frac{dC}{dt}(C, L_{sc}, \varepsilon_f, t) = \frac{1}{\tau_R} C_L(L_{sc}) F_{rise}(t) + \frac{1}{\tau_D} \cdot (C_{rest} - C) / (1 + e^{(T_{fun}(L_{sc}) - t)/\tau_D})$$

with t as result of: $\{ \text{while } (t > t_{cycle}) : t = t - t_{cycle} \}$

$$\sigma_{fact}(L_{sc}, C, \varepsilon_f) = \sigma_{act} C (L_{sc} - L_{sc0}) \cdot 1E6 \cdot (L_s(\varepsilon_f) - L_{sc}) / L_{se,iso}$$

1E6: due to SI units

if $(L_{sc} < L_{sc0})$: return 0

if $(L_{sc} \geq L_{sc0})$: return σ_{fact}

These formula can be motivated fairly well. The first equation converts a fiber strain value ε_f to the current physical length L_s of the sarcomere. Actually that is already given

3. Mathematical Modeling

by the stretch ratio $L_s = \lambda_f L_{s,ref}$ and the definition of $\varepsilon_f = 1 - \lambda_f$, but the authors preferred an exponential formulation, which—according to Taylor series expansion—is very similar and does not make a big difference nevertheless. As described in section 2.2, the local strain in fiber direction ε_f is physiologically restricted to the range between -0.2 and 0.3, where the exponential approximation is almost perfectly linear with a difference below 2% for $\varepsilon_f < 0.2$ and below 5% for $\varepsilon_f < 0.3$. All formula incorporating sarcomere length L_{sc} could not be directly used but had to be scaled from mm to m to become applicable in the FEM simulations. The formula for F_{rise} and $\frac{dC}{dt}$ take t as input, but work just for the first heart beat correctly. Instead of t , these are fed with a “modulo-divided” equivalent time shifted to be within the first heart beat (“while $t > t_{cycle}$: $t = t - t_{cycle}$ ”). Additionally, in the last two lines the force $\sigma_{f,act}$ is set to 0 if the sarcomere length becomes unphysiologically small ($L_{sc} < L_{sc0}$) in order to prevent negative active forces. This also ensures the minimum-length restriction is mostly fulfilled in the FEM simulations.

For the implementation, state variables need to be stored and restored in a replaceable manner accordingly, which gets more complex with an increasing number of state variables. The tension model is called multiple times during a single solver step of the time integration by Newton’s method. In case of a failed solver step, the implementation needs to be able to detect and perform a “step back” of all state variables on its own. For easier handling, a state variable object was introduced which contains all state variables at once. A function “SaveAndResetStateVariablesAsNeeded” analyzes the time step size on every call of the tension model in order to detect a progress in time or a step back, and then saves or restores the state variable object with help of the copy constructor in case of a progress in time or a step back.

3.8. Modeling of the Pericardium

The pericardium is a semi-rigid, sac-like structure enclosing the four chambers of the myocardium. It restricts motion of the heart’s outer shape and evokes a hemodynamic impact especially through the thin-walled atrial region. Currently, two different modeling approaches of pericardial effects in simulations of heart mechanics are known: Kerckhoffs et al. (2007a) and Fritz et al. (2014).

3.8.1. Kerckhoffs Pressure Pericardium

Kerckhoffs et al. (2007a) used the pericardial influence in a 3D FEM elasticity simulation of a bi-ventricular canine heart coupled to a closed-loop model of the vascular system. Atria are only represented and modeled as time-varying elastance in the circulatory system part. The pericardial effects are incorporated by computing a single scalar-valued pressure term depending on the total heart volume enclosed by the pericardial sac. The total heart volume V_{total} is given by ventricular blood from the finite element mesh, atrial blood volumes from the circulation model, and a constant parameter for the myocardial

3. Mathematical Modeling

volumes:

$$V_{total} = V_{LV} + V_{RV} + V_{MYO} + V_{LA} + V_{RA}$$

The pressure volume relationship is exponential in its nature (and thereby similar to what Freeman and LeWinter (1984) observed in measurements of dogs). The computed pressure is added in two different manners. Its value gets added to the atrial pressures in the circulatory system, and applied as epicardial surface force onto the ventricles in the finite element simulation. Kerckhoffs et al. (2007a) gave for the pressure an exponential relationship

$$p(V_{total}) = \alpha (\exp(\beta(\frac{V_{total}}{V_{total,0}} - 1)) - 1)$$

with a “pressure scaling factor” $\alpha = 0.031$ (unit: mmHg) and an “exponential shape factor” $\beta = 6.76$ (unitless) for healthy hearts. That formula however did not work as described, the plausibility check $p(V_{total,0})$ does not yield the intended zero pressure.

After revisiting the brackets, the corrected and from now on used formula is

$$p(V_{total}) = \alpha(e^{\beta(\frac{V_{total}}{V_{total,0}} - 1)} - 1)$$

with the last -1 being moved out of the exponent. This neutralizes the exponential function in case of $V_{total} = V_{total,0}$ and creates the expected $p(V_{total,0}) = 0$.

The pericardium in this model creates a positive pressure (compression effect) at the end-diastolic state and a strictly positive but decreasing inward-pointing pressure during the progressing contraction.

3.8.2. Fritz’ Volumetric Contact Pericardium

To approximate the influence of the pericardium and the surrounding tissue, the mesh contains besides the myocardium of the four chambers a thin, approximately 2 cm thick layer of volume elements surrounding the myocardial elements. The inner surface of that surrounding tissue matches the outer myocardial surface. For each myocardial surface element (master element), a corresponding surface element on the surrounding tissue surface (slave element) is found in normal direction of each quadrature point \mathbf{x}^m and an additional surface force g_N is computed depending on the distance to the intersection point \mathbf{x}^s and a global penalty parameter ε (Fritz et al., 2014).

$$g_N(\xi) = \varepsilon(\mathbf{x}^s(\xi) - \mathbf{x}^m(\xi)) \cdot \mathbf{n}^m(\xi)$$

In order to stabilize the algorithm, especially with myocardial tissue parameters behaving less stiff than the ones used in Fritz et al. (2014); Fritz (2015), an additional scaling depending on the angle between the normals of the two elements needed to be introduced. A second improvement is a better search order of slave elements for an improved and significantly faster runtime on larger meshes. Exploiting the fact that there is at most

3. Mathematical Modeling

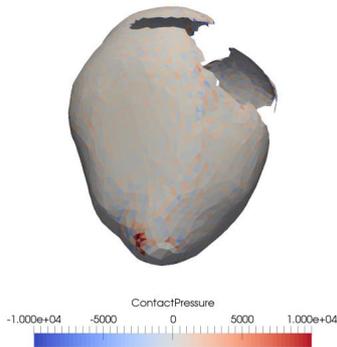


Figure 3.7.: Contact forces from the volumetric pericardium model acting on the epicardial surface during end-systolic state. Near the apex is a pronounced pressure peak, while in other areas the pressure seems to be distributed more homogeneously.

one corresponding slave element, the search can be aborted when it is found. This is now done such that the possible slave elements are traversed in an order depending on the proximity to the slave element found in the previous time step instead of the full vector of all elements. The neighborhood relationship map needs to be computed only once during initialization and before starting the loop over all time steps.

Fritz et al. (2014) incorporate the pericardial effects by extending the mesh. Their volumetric mesh of all four chambers (without volumetric representation of the valves) is extended by an additional volumetric mesh of approximately 2cm thickness representing the surrounding tissue. Between the two meshes, a contact interface problem gets solved throughout each computed time step. Thereby, the pressure forces needed to minimize the distance between the pericardial surface and the inner surface of the surrounding tissue. For each element on a manually defined subset of the heart surface (“master element”) a corresponding element on the inner surface of the surrounding mesh needs to be found in normal direction (corresponding “slave element”). The epicardial surface used to define the master elements of the contact problem was not closed and covers only an arbitrarily chosen part of the upper atria as well as most parts of the ventricular epicardium. The simulations were found to be highly unreliable and their success depends a lot on the addition and choice of these holes in the master elements surface, which have no physical representation. The end-systolic state matches the stiffness of the surrounding tissue by design of the surrounding mesh as a convex hull. The parameters of the surrounding mesh were chosen to be isotropic (Neo-Hook material, Mooney-Rivlin material with $C_2 = 0$) with stiffness C_1 chosen such that the end-systolic state matches visually to some slices in the same state from corresponding CINE-MR measurements.

However, the original publication Fritz et al. (2014) has a number of questionable points susceptible to criticism. Neither material properties like passive parameters, thickness of the mesh nor the general behavior of the surrounding tissue were based on physiological thoughts or with a certain justified behavior in mind. Many of these

3. Mathematical Modeling

necessary assumptions are neither described nor mentioned in that paper but still are important or even essential for a credible, correct, or at least reproducible modeling. As an example, an experimentally observed 8.8% change of pericardially enclosed volume (Carlsson et al., 2004) is so small that for that deformation the differences between linear and non-linear material law is mostly insignificant. Nonlinear tissue requires unnecessary additional computational overhead, but is still used with a seemingly arbitrary choice of parameters. As an other example, the impact of the amount of volume conservation in the surrounding tissue on deformation is huge with fixation of the outer surface combined with a thickness of only a few centimeters (volumetric locking). Finally, thickness of the surrounding mesh itself plays an important role as well. Twice the thickness allows the double amount of “amplitude” or orthogonal surface displacement under the same surface pressure created by the pericardial layer (compliance), but the choice of thickness was never mentioned and seems to be chosen arbitrarily. These points of criticism however do not inherently translate to the modeling approach itself, but rather indicate that further systematic investigations are required.

In this model, the pericardium creates no surface forces at all during the end-diastolic state, and an outward-pointing surface force increasing with progressing contraction (expansion effect), which is fundamentally different from the previously described pressure model.

3.9. Modeling of Circulatory System

A basic property of the aorta and the arterial system is a certain mechanical elasticity as described in section 2.4. Its vascular purpose is to create a smooth blood flow by storing part of the ejected volume during systole and releasing it during diastole, known as the Windkessel effect. An often used model to describe this behavior is the *three-element Windkessel model (WK3)*, consisting of a compliance paired with two resistances. Figure 3.8 shows the analogy between its modeling as hydraulic system and electrical circuit, the arising differential equations are the same. Blood flow Q corresponds to electric current, blood pressure P to electric potential and stored blood volume V to electrical charge. Electrically, Kirchhoff’s junction and loop rules are used to derive the differential equations and analogously ensure conservation of blood volume and continuity of flow. The characteristic values are named correspondingly as *aortic characteristic impedance* Z_C , *total peripheral resistance* \tilde{R} and *total arterial compliance* \tilde{C} .

The amount of blood volume V_C stored in the capacitance depends on the pressure drop $P_C = P_{in} - P_{out}$ and is related via the compliance C as

$$P_C = \frac{1}{C} V_C.$$

3. Mathematical Modeling

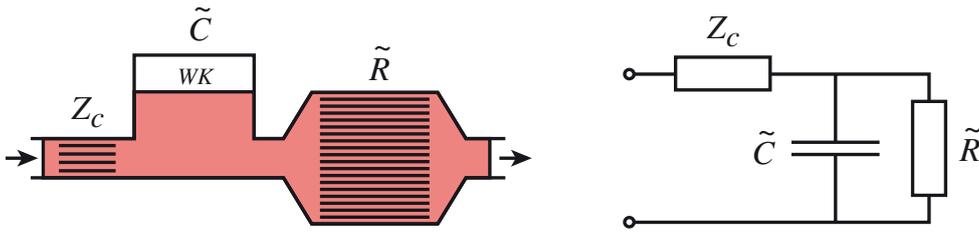


Figure 3.8.: Three-element Windkessel model in its representation as hydraulic system (left) and electrical circuit (right). Adapted from Schuler (2016); Westerhof et al. (2009).

Similarly, in- and outflows $Q_{in} = Q_{Z_C}$ and $Q_{out} = Q_R$ are related to the respective pressure drops by the corresponding resistance values

$$Q_{in} = \frac{1}{Z_C} P_{Z_C} \qquad Q_{out} = \frac{1}{R} P_C.$$

Finally, the stored blood volume changes with the difference between adjacent inflow and outflow

$$\dot{V}_C = Q_{out} - Q_{in}.$$

The inverse of compliance $E := 1/C$ is called *chamber elastance* and can be measured experimentally via the required pressure change to obtain a certain volume change $E \approx \delta P / \delta V$. In case of an actively contracting chamber, the corresponding component is characterized by a time-varying elastance $E = E(t)$, with its time course either prescribed by an explicit function or approximated by the coupling algorithm with $E \approx \delta P / \delta V$ via pressure perturbations within the FEM model.

In order to model the whole cardiovascular system, separate Windkessel models are needed for arterial system, venous system and the heart chambers (with separate V_{Art} , V_{Ven} and V_{Ventr} , respectively). These are either connected to ground (open-loop) or combined in a closed system (closed-loop). Diodes between the different parts of a closed-loop system restrict flows to one direction and can be used to model the unidirectional flow property of the heart valves, expressed by min/max functions in the respective equations for Q_{in} and Q_{out} .

Fritz et al. (2014) used two separate and unconnected open-loop Windkessel models to simulate left and right ventricular afterload. By the nature of such an open-loop approach, flow rates between the two chambers and ejection volume are not synchronized. Furthermore, parameters of the systemic circulation were used for the pulmonary circulation as well, with the consequence that the (usually weaker) right ventricle produced a significantly lower ejection volume than the left one. Santamore and Burkhoff (1991) therefore used a number of Windkessel models connected to each other to create a closed-loop system of the cardiovascular system. Each Windkessel model represents one

3. Mathematical Modeling

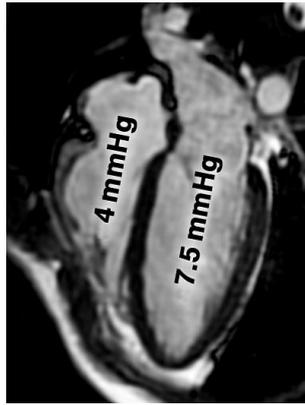


Figure 3.9.: Visualization of the end-diastolic in-vivo pressures acting on the endocardial surfaces during image acquisition, from Schuler (2016).

of the systemic and pulmonary arterial and venous systems. Blood conservation and equal ejection volumes are then automatically fulfilled by the electrical principles of charge conservation.

3.10. Stress-free State Estimation

During in-vivo image acquisition of living patients, the heart is exposed to the full set of internal and external forces – even if the segmentation is done at end-diastolic state (Figure 3.9). These forces comprise end-diastolic blood pressure and probably forces due to the pericardium and surrounding tissue as well (target state). Removing these forces results in a different state in which no endocardial pressures are applied (pressure-free state). Experiments show that even when removing all external forces, there still exists residual stress causing an opening of a ring of the left ventricle by a certain angle (Rodriguez et al., 1993; Costa et al., 1997). A strategy is needed to incorporate residual stresses, either by applying strain directly or increasing strain by applying stretch.

An elementary assumption often made for finite element simulations is that pressure-free state is approximately equal to the stress-free state. In that case, the task “reduces” to finding one pressure-free state, that yields the segmented state after inflation. This state is taken as (stress-free) reference state relative to which deformation and hence stress are computed. End-diastolic volume of the left ventricle lowers to about 55% when removing pressure (Klotz et al., 2006). In this section, *unloading* describes the determination of a pressure and stress-free reference state, *preloading* describes the whole process to obtain a target state with residual stress, i. e. including consecutive inflation. In a heart geometry comprising of several chambers, unloading is only possible simultaneously for all chambers at once since residual strains influence each other.

Unloading by image acquisition

This is surely a very elegant way since it does not involve any kind of “computational guessing”. Instead of using the end-diastolic state, Genet et al. (2014) used a whole-heart MRI data set from end-systolic state. This state is then taken as pressure-free state and inflated to the expected end-diastolic volumes, which creates the residual stress. The drawback is that this method needs to be taken into account already during image acquisition. Furthermore the heart’s state at time of acquisition (end of systole) is not as much in rest as it would be at the end of relaxation phase (end of diastole) and motion artifacts are more likely to occur.

Unloading by negative pressure

This is a simple one-shot method used by Fritz et al. (2014). It works as follows: (1) A negative pressure is applied to the endocardial surface, (2) that state is set as stress-free reference state, (3) a positive pressure is applied. The negative pressure needed to be adjusted manually to obtain a state approximately fitting the target state after inflating the chambers with their respective end-diastolic pressures. Often negative and positive pressures are set similar. It allows to control neither the global amount of residual stress (influenced only by passive properties), nor the actual shape of the surface after reinflation.

This works fine as long as material parameters are stiff and negative pressure causes only small deformation and a comparatively large unloaded volume (92% in Fritz et al. (2014)), which on the other hand is not a physiological case. In the right ventricle, where the myocardial wall is thinner and geometry is not distinctively convex, the resulting pressure-free state shows considerable unnatural self intersections and is not able to build up stress after inflation, with a shape very different from just a smaller-scaled version of itself. Similarly, the atrial walls are fairly thin compared to end-diastolic pressures and do not generate much passive stress either that could help to keep a stable shape under the negative pressure. In case of additionally required fitting meshes (e. g. a three-dimensional pericardium layer), these have to be created afterwards. Additionally, these have to be recreated whenever parts of the mesh or passive properties even slightly change (e. g. locally stiffer tissue due to modeling ablation scars), which makes comparison studies difficult.

Preloading by active stress

Another idea to create residual stress is to permanently add a small amount to the stress tensor, similar to an offset permanently applied to the active contraction force. This is also a one-shot algorithm, however it does not need to generate a separate, stress-free state. This way, residual stress is not modeled as passive material property (different reference system) but as part of the active material. While this might be a viable approach to generate the static equilibrium of forces in the fairly static end-diastolic state, the course of the passive forces behaves differently during motion of the heart.

3. Mathematical Modeling

A permanently active residual contracting force is an inherent property of the Lumens model, if the parameter related to calcium concentration at rest C_{rest} is not chosen to be zero (Lumens et al., 2009). The latter however was necessary to be able to start the simulation in an equilibrium state. Similarly, the atrial tension model by Land and Niederer (2017) has a permanently active residual force. The authors state that their contractile forces are sufficient to balance out pressure forces, so they did not use significant unloading in the atria. This method allows to control the amount of residual stress, but not the shape after inflation to end-diastolic pressure.

Unloading by Bols iterations

Iterative methods have the advantage that they reduce an error measure up to a desired accuracy, and were published by several authors (Rajagopal et al., 2007; Sellier, 2011; Bols et al., 2013). Rajagopal et al. (2007) developed their method specifically to “determine the reference state from a loaded configuration”. While Rajagopal et al. (2007) use a Newton method to determine the reference coordinates that minimize the error between inflated and target state of breast-imaging, less expensive fixpoint-based methods were developed later (Bols et al., 2013; Sellier, 2011). Bols et al. (2013) use it for blood vessels, their method is presented here. Applications however are not limited to soft-tissue mechanics, and comprise car tire manufacturing as well (Sellier, 2011). Sellier (2011) moreover gives a thorough literature overview and convergence analysis.

The *Bols unloading algorithm* starts with the segmented state as initial guess for the unloaded state. In each step, a new guess of the unloaded state is computed by: (1) applying positive pressure to the current unloaded state, (2) measuring the (positive) displacement of each node, (3) applying the displacement in negative direction to the node positions of the current unloaded state. Ideally, the error between this state plus inflation and the target state reduces in each step. Different from the previous methods, final node positions under chamber pressure and residual stress are arbitrarily close to the ones of the initial mesh.

The algorithm was found to work fine on mostly regular, convex geometries, like a mesh containing only the left ventricle. Problems occurred mostly in non-convex regions of the heart, especially in a thin-walled and almost flat part of the right ventricle. As a typical behavior, the error in the right ventricle (maximum node difference, but l2 norm of all node differences as well) decreases during the first 5 iterations down to a range of 10-20 mm, but then increases for all following iterations. This has several reasons, some of which are the non-linear passive material properties and a non-convex geometry. Step (1) basically does a linearization of the passive behavior under inflation. Introducing a scaling factor to lower the negative displacement in step (3) of the algorithm was found to add a certain level of robustness. Such a damping evokes that displacements are more reliably predicted by the previous inflation in step (1). Since the displacements of question result from inflating a smaller geometry than the target state, they have to be slightly smaller. Another improvement was obtained by applying the positive pressure only in small increments of the full end-diastolic pressure, thereby improving the approximation of the linearized passive behavior under inflation. The problem might

3. Mathematical Modeling

be that the original algorithm basically does a fixpoint iteration, which only converges if the underlying function has a contracting property, i.e. the displacements become lower for each iteration.

In the atrial region the algorithm worked less problematic, probably due to a generally more convex shape. In the atria however model reliability is generally low due to considerable uncertainties in several areas such as in-vivo passive material parameters, wall thickness and wall position from MRI, resulting effective compliance, and even numerical error from a low number of transmural elements (which was only one element throughout the wall). Another problem regarding the convergence problems with a four-chamber heart mentioned above are certain properties of the target state. In general, a thin geometry with positive pressure on its inner surface tends to have a smooth outer surface, even under the presence of residual stress. Unprocessed surfaces segmented from MRI images are not in general smooth, segmentation inaccuracies and fairly coarse voxels contribute to that. It is however known that the geometry was exposed to positive pressure at the time of acquisition. Thereby as a second, different application, Bols unloading method could be used as a tool to smooth out segmentation errors (sufficient inflation makes everything convex). After segmentation and meshing of the target state, simply applying one Bols iteration with a small (linearization) amount of positive pressure should improve the convexity of the mesh. Segmentation inaccuracies could be detected by investigating regions with a high local node position discrepancy before and after that first iteration. A reasonable aim should be to get an error of the segmented target state in the range of measurement error from MRI, i.e. 1-2 mm.

An open question is which effects should be respected during the procedure to determine the unloaded state. While in reality, residual stress from active tension models, or forces of the epicardial surface from the pericardium and surrounding tissue might be present during image acquisition, these definitely have an effect on the unloading procedure and its results. Activating a contact pericardium prevents the ventricles to become non-convex like they usually do during negative pressure unloading. However it is still unclear if pericardial forces are present at all at end-diastole and in which direction they point. Before that, the general role of the pericardium needs to be fundamentally investigated.

3.11. The Heart Model by Fritz2014

The model published by Fritz et al. (2014) is the base for the work in this thesis and is therefore described shortly here.

3.11.1. Modeling components

Fritz et al. (2014) used an MRI-segmented mesh of a human four-chamber model and a second mesh representing a truncated volumetric layer of the surroundings to show the effects of the pericardium, realized by a contact handling algorithm between two meshes as described in Section 3.8.2. Passive forces were represented by the Guccione material law with Omens parameters (Omens et al., 1993) for the ventricles, and Mooney-Rivlin material with parameters averaged from left atrium anterior (LA_{ant}) and left atrium

3. Mathematical Modeling

posterior (LA_{post}) from the publication of Di Martino et al. (2011) for the atria. Active forces were precomputed on a fixed and static geometry by solving the bidomain equation in combination with a tension model of high complexity developed to reflect the feedback mechanisms during electro-mechanical coupling (Sachse et al., 2003), which however does not use length-dependency due to the fixed geometry. Blood pressure was computed by a non-looped WK3 model for the left ventricle (the afterload of the aorta is connected to a zero-pressure condition instead) and for a single heartbeat. The same model and identical parameters were used for the right ventricle as well. No pressure was assumed in the atria in Fritz et al. (2014). In a later version, atrial pressures were also computed by a simple resistance R_1 and an assumed representative compliance of the ventricle C (Fritz, 2015), however without any exchange of blood flow rates between the separate models of the single chambers pressures. The afterloads of the single pressure models were defined by four discrete states that run through in a prescribed order by passing fixed pressure thresholds, again without information interchange. As consequence, the states do not fit (e. g. AV valve is open for the atrium, but closed from the ventricle’s perspective), and the simulations could comprise at most one heart beat.

The essential modeling of the pericardium was realized by their own modeling approach (“contact pericardium” in Section 3.8) and validated by visual comparison of the end-systolic and end-diastolic outer shape against cine MRI of the same patient. In (Fritz, 2015), the model was used to determine tension values over time in each cell. Instead of prescribing a tension model, tension values in each cell and time step are free parameters to be estimated by a specifically developed solver algorithm. Aim was, to measure the motion of the surface from MRI, and compute corresponding tension values that produce this specific motion, which are not directly measurable. Another application of the later model in Fritz (2015) was the evaluation of the outcome of atrial ablation scars. Therefore, five different ablation patterns were created with modified electrical and mechanical properties over the control case, and the effect on the atrial outcome (volume, pressure) was measured. The more scar tissue is present, the less force the atrium develops, with the consequence of reduced volume change and peak pressures.

3.11.2. A critical view on the Fritz2014 model

The Fritz model was designed for the purpose to show that two new developed methods – these are the pericardium modeling by contact and the inverse determination of tension values – principally can be realized. It served well for that sole purpose, but had some major flaws concerning already the physiological behavior of a healthy heart and the ones with ablation scars. For using the simulated behavior to draw real conclusions, a further development of the model towards a more plausible behavior is necessary. The major points are described here.

Available validation of global behavior showed that it lacks in enough displacement of the AV plane compared to cine MRI data (Fritz et al., 2014), and that the pressure-volume behavior of the isolated pericardium even contradicts the measurements “in-vivo pericardium” including impact from the surrounding areas from Freeman and Little (1986). While the original pressure pericardium by Kerckhoffs et al. (2007a) used volume

3. Mathematical Modeling

measurements of the pericardial sac as base, the original contact pericardium by Fritz et al. (2014) compared against cine MRI images. It is not directly obvious which of the two behaviors is the correct, to-be-expected one. Moreover, it is not directly clear why a preloading of the pericardium with a literally inverted direction of forces would not have a different drastic impact on the simulation outcome with the contact model.

In the modeling of the circulatory system, separate open-loop models were used for each chamber with the result that the ejection volumes of the right and the left ventricle are different. Moreover, the parameters of the pulmonary circulation were chosen to be the same as in the systemic circulation, although the pressures therein are one magnitude smaller. In order to reach the LV ejection volume, compliance, resistances, etc. are necessarily different with the lower pressure and force generated by the right ventricle. These approaches are acceptable when the aim is to produce an internal pressure load over no pressure boundary, but leave room for improvement and do clearly not represent a satisfying physiological case.

In the modeling of the contractile behavior, the tension values were pre-computed on a static geometry. While the spread of electrophysiological activation in a healthy heart is significantly faster and mostly finished when the mechanical deformation starts, it is definitely not true for everything happening after the electrical activation. The assumption is only possible when the parameters of interest depend on the beginning of activation e.g. the local activation time or the activation pattern. Neglecting length-dependency necessarily leads to a locally wrong temporal behavior in terms of deformation. The heart however is exposed to a non-trivial amount deformation, the enclosed left ventricular volume changes by 60-70% during one beat. The amount of developed tension as well as its time course are highly dynamic and depend on the current stretch in fiber direction much more than e.g. transmembrane voltage or intracellular calcium concentration (Sachse et al., 2003). Accordingly, the Starling effect, an increase of stretch leading to a larger developed tension and hence overall deformation and ejection volume after an increase in LV volume, cannot be reflected. Similarly, the lower bounds of deformation due to the physiological microstructure of myocytes, cannot be reflected by a pre-computed tension modeling as well. Adapting model parameters on a static geometry might be able to hide a wrong behavior of the model on one static geometry, but does not replace the necessary dynamic effects of length-dependency on a dynamic geometry.

Regarding passive material parameters, evaluating the isolated behavior is the least complex. The chosen parameters by Omens et al. (1993) are outliers amongst most other parameter sets found in literature (see Table 3.2). The resulting isolated passive behavior is almost certainly wrong, which comprise the pV response, relative volume increase under a certain pressure or unloaded volume (which all are only global parameters). The unloaded volume depends a lot on the stiffness behavior of the ventricle and changes significantly when the parameters behave weaker. This however makes different methods necessary that are able to handle non-linear pV response, the non-linearity lets the negative pressure unloading approach fail. In the model by Fritz et al. (2014) the unloaded volume was as high as 92% of the loaded volume, while it should be rather around 50% of the loaded volume (Klotz et al., 2006).

3. Mathematical Modeling

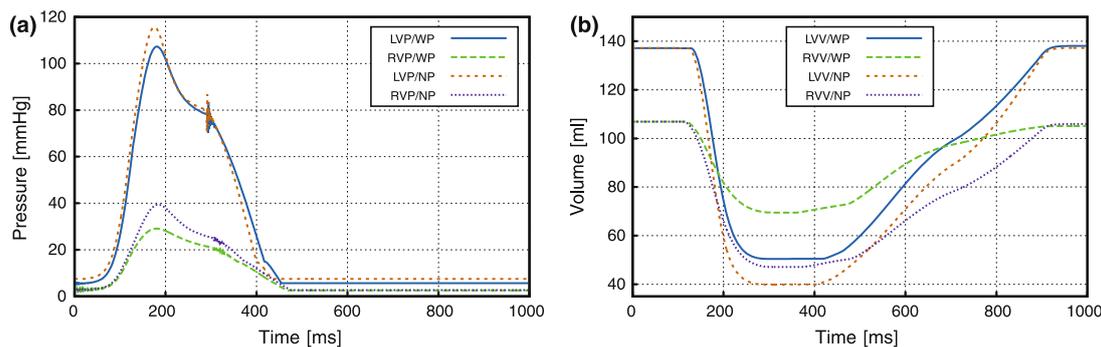


Figure 3.10.: Simulated pressure and volumes curves in the left (LVP, LVV) and right (RVP, RVV) ventricle with (WP) and without (NP) using a contact pericardium, from Fritz et al. (2014).

Concerning the behavior of surrounding tissue, a number of points were left unclear. The advantage of the modeling idea over simpler modeling were not clear besides the sole purpose of having “more complexity”. Evaluation of the isolated behavior was completely left aside, instead the outer motion of the fully-featured heart simulation was compared against the motion visible on cine MRI from the same patient. This similarity is not surprising, since the global stiffness and incompressibility parameters of the surrounding tissue (Mooney-Rivlin material) were modified until the motion fitted as a whole and possibly existing local errors are too small to be still visible. Contact problems are actually very common in the field of elasticity computations and widely available in commercial standard software like ABAQUS or COMSOL, but only rarely applied to cardiac mechanics. Due to the complexity of each involved component in a full-featured simulation, a plausible behavior of the model as a whole does not necessarily attest the isolated behavior of pericardium and surrounding tissue to be correct. These could be the passive pV response of a model comprising only the mesh of the pericardium and surrounding tissue. Measurements of the isolated, non-linear behavior are available by Freeman and LeWinter (1984); Freeman and Little (1986), or of the non-isolated volume variation of the whole heart by Carlsson et al. (2004). Preloading of the contact pericardium was left aside and mentioned as a limitation. Besides the publication by Fritz et al. (2014), a highly relevant but generally unanswered question are the fundamentally different behaviors of the contact approach and the pressure approach. A pushing or a pulling effect of the pericardium during contraction is completely opposite, and choosing the wrong one is twice as bad as leaving the pericardial forces away. Finally, while obtaining plausible local and global behaviors should be the aim, a whole-organ model should at least reflect the general expectancies found in measurements on organ level.

Difficulties for modeling improvements

Modifying most components towards a more correct isolated behavior did lead to an even worse behavior than before in combination of all modeling components. A resulting awkward behavior of the model can be observed by the volume diagram of the left ventricle in Figure 3.10. During relaxation phase, the simulated LV volume increases in the plot approximately linear (blue line in the right plot). Measurements however show that the relaxation is steep at the beginning and less steep at the end, similar to an inverted exponential function. A steeper slope at the beginning of the relaxation is needed. The total duration of the relaxation fits and is not too long.

In case of using a more correct passive material, the behavior of the left ventricle is less stiff and has less stored tension energy that can be released. As a consequence, the slope would be less steep in the beginning of relaxation, and less steep in the end, which is the opposite from what is needed! In case of a more correct unloaded state with significantly smaller volume, the passive material needs to be stretched during relaxation and counteracts the volume increase. Energy is stored in the passive material instead of being released, so the slope would be less steep in the beginning and way less steep in the end, which is again the opposite from what is needed! In case of a different and probably more correct preloaded pericardium, the pericardial forces are pushing the epicardial surface inwards during the end-diastolic state instead of pulling. Hence, the inward-pointing pericardium forces counteract the volume increase during relaxation. As a result, the slope would be less steep in the beginning and at the end, which is the opposite from what is needed! In all three cases, improving the respective passive behavior lead to a worse volume time course during relaxation and an unsuccessful improvement of overall relaxation behavior.

Since at each arbitrary point in time, the equilibrium of forces needs to be fulfilled, the direction of the forces vectors from the single modeling components is not irrelevant. For drawing conclusions not only from the forward simulation, but also from approaches that make use of a forward model e. g. the inverse determination of active tension (Fritz, 2015) or an estimation of passive material parameters as demonstrated in Section 6.1, the quality of the results can be only as good as the quality of the forward model, and a correctly behaving forward simulation is essential.

4. Implementational Aspects

The most limiting factor of any multi-component modeling are the computational costs required to obtain even a coarse representative solution. In order to achieve this aim, a number of essential factors were developed, realized, and are described in this chapter: An object oriented programming framework with well-designed interfaces for an extensible implementation, coupling of the different model components, a fast solution with an efficient use of computational resources through software parallelization and adaptive time increments. Even though cardiac modeling is a considerable software engineering challenge, the combination with a “good mesh” is key to obtain a working simulation. Hence, a brief description of the workflow to create a suitable geometric model starting with MRI data and the geometric models used in this thesis is added as well.

4.1. The Biomechanical Framework CardioMechanics

In this section, the most important components of “CardioMechanics”, the framework for cardiac biomechanics (CB) simulations are described. It relies on two external libraries, the VTK library for data export and the PETSc library for solving the evolving linear and non-linear equation systems. Its first implementation was written as in-house code by Thomas Fritz (Fritz et al., 2014; Fritz, 2015). CardioMechanics is implemented in C++ and makes extensive use of an object oriented programming structure. It is intended as a multiscale electromechanical simulation framework that uses the method of finite elements to solve the equilibrium of forces. At the state of Fritz (2015) however, it was never used with more than single-core due to a number of bugs in its parallel implementation. Figure 4.1 and Figure 4.2 show its main components, these are:

CBCardioMechanics provides the command line interface to the user. It parses the user input and starts the other main components depending on the user input in a reasonable manner. Configuration files are passed in the xml format and can contain parameters for configuring the Model/Mesh, Solver, Exporter, Materials and all SolverPlugins. All parameters can be overwritten from command line e.g. for quick testing of a different time step. Moreover different levels of verbosity can be chosen for the logging data to facilitate debugging but not clutter the output on production runs.

CBModel handles all the data containing geometric information. It gets filled during mesh loading by CBModelLoader and is used to pass the geometry information to the solver. It’s second purpose is during export of simulation data to vtk files. Therefore it gets updated with the latest node positions, and augmented with other additional

4. Implementational Aspects

information like active stress, relative fiber length and Cauchy deformation tensor, passed to the `CBModelExporter` and finally converted to the VTK.

CBSolver solves a non-linear problem in each time step (using PETSc's `SNESolve` function for solving non-linear equation systems) and implements the time integration in one of its derived classes (`CBSolverStatic` and `CBSolverNewmarkBeta`). It handles big-picture stuff like step size computation, calling solver plugins and mesh exporting in the `Run()` function (Figure 4.1). `CBSolver` has its own special data structures in a suitable format for efficient computation (parallel PETSc matrices and vectors), that need to be converted to a `CBModel` object prior to exporting. Instead of directly computing new node coordinates, in each time step displacements are computed in the actual `SolverStep()` function and added to node positions from the previous time step.

CBSolverPlugin provides a general interface for adding additional functionality to the solver. Examples of solver plugins are `CBContactHandling` plugin for realizing permanent contact to the Pericardium, or `CBCirculation` for coupling in a Model of the blood pressures in the circulatory system. This class provides the interface to functions like `Init()`, `Apply()`, `StepBack()` and `Export()` that are called before starting the simulation, for updating its own information during each solver step, for returning to the previous state after each non-successful solver step and for writing information into the model and/or a text file after each successful solver step matching an export time condition. SolverPlugins can influence the next time step in the Solver using return codes like `CBStatus::SUCCESS` (solver is allowed to proceed to the next time step), `CBStatus::FAILED` (solver should reduce time step because plugin was not satisfied with the result) or `CBStatus::REPEAT` (solver should compute the current time step again, so the plugin can do multiple iterations like `CBCirculation` does).

CBElement consists of two different type of elements, `CBElementSolid` and `CBElementSurface`. `CBElementSolid` provides functions to fill the stiffness matrix with node derivative entries in a proper petsc format. These are only called implicitly from PETSc's `SNESolve()` function that uses the solver's `NodalForcesHelperFunction()` to call each solid element's `CalcNodalForces()`, which computes the element's contribution to the system matrix. This complicated structure is prescribed by PETSc's `SNESolve()`, so unfortunately this core functionality of the solver needed to be implemented in a confusing manner. Each solid element knows about its own current and initial node positions, fiber directions, material model and tension model and uses these information to compute its deformation tensor and the resulting contribution to the global energy derivatives, respectively. Each solid element has a `CBConstitutiveModel` for computing the passive part and a `CBTensionModel` for computing the active part of the stress tensor during its `CalcNodalForces()` (Figure 4.2).

In contrast to `CBElementSolid`, `CBElementSurface` provides functions to apply surface forces, compute enclosed volumes and find contact partners for the contact Handling algorithm.

4. Implementational Aspects

CBElementAdapter is a global object carrying pointers to all important global and/or element-specific components within CardioMechanics, mainly due to efficiency reasons. Examples are a pointer to the solver, to the constitutive model, to the time stepping object, etc. This is used on the one hand to give each element read access to information that is actually technically not available within an element (global variables). To keep memory consumption low, node positions are stored only once in the memory in form of the solver's solution vector. The location in memory is provided by CBElementAdapter. On the other hand, it allows the elements to add their contributions to mass and stiffness matrix directly to the solver's data objects.

CBConstitutiveModel provides the functions `CalcEnergy()` and `CalcPK2Stress()` to allow each solid element to compute its deformation energy from a given deformation tensor due to the passive material properties. The structure is very similar to `CBActiveStressModel`.

CBTensionModel provides functions to calculate active energy contributions due to contracting muscle fibers from a given deformation tensor. `CalcTension()` yields the tension in the fiber direction and `CalcActiveStressTensor()` yields the contribution to the PK2 stress tensor. It is similar in structure to `CBConstitutiveModel`.

Due to the inheritance of object oriented languages, specializations of each module can be exchanged and extended by a different one of the same type without needing to rewrite large parts of the code. As specializations of the `CBElement` class there are `CBElementSolidT4` and `CBElementSolidT10`. Specializations of the `CBConstitutiveModel` are `CBConstitutiveModelGuccione` and `CBConstitutiveModelMooneyRivlin`, all of them providing the same interface functions, respectively.

4.2. Adaptive Time Increments

For a chosen time step length, Newton's method sometimes does not converge (diverge) within a certain number of Newton steps. The reason is often that the initial guess for the node displacements is too far away from final solution for the chosen time step. During the opening and closing events of the valves in the systole, the acceleration is high and demand a small step size, while the diastole is more relaxed. One possible solution is to re-run the whole simulation with a smaller step size, that is however only needed at certain points in the heart cycle. So the more efficient and elegant way is to use an adaptive step size that is small only when needed, freeing computational resources for other parts (multiple heart beats, stronger coupling of the systems, finer mesh size).

The requirements for an adaptive time increment algorithm are:

- Modify the step size depending on the solver return code (which is one out of "success" – system solved, "fail" – no solution with the current step size found and "repeat" – system solved, but coupling condition not yet fulfilled),
- avoid a large number of failing step sizes,

4. Implementational Aspects

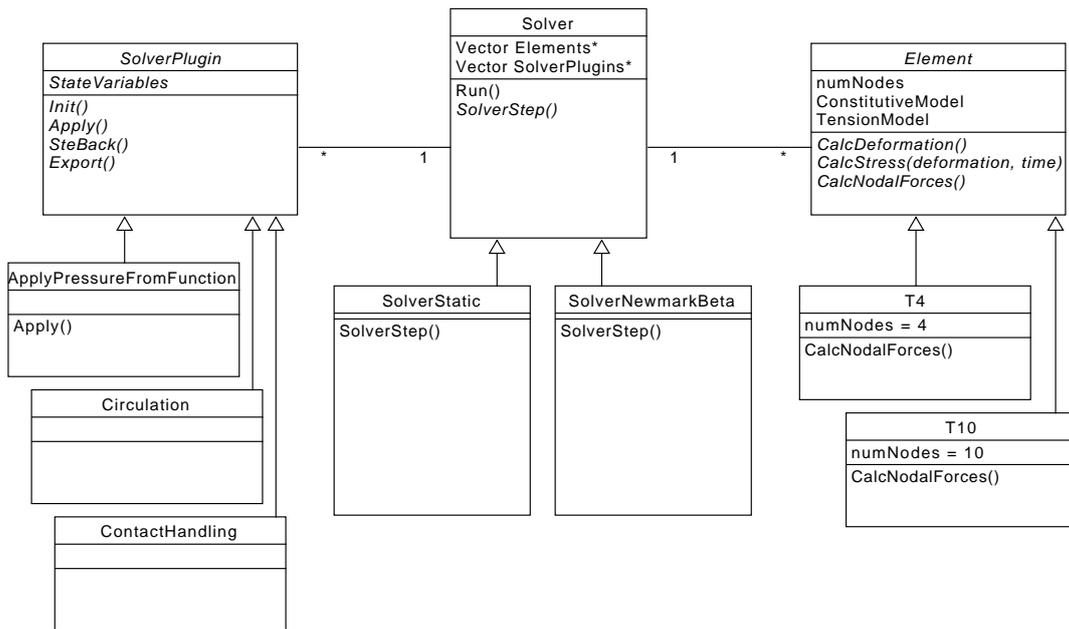


Figure 4.1.: Structure of the solver and interaction with the Elements and SolverPlugins. In the implementation, all class names additionally have the prefix “CB” (CBSolver, CBCirculation, ...).

4. Implementational Aspects

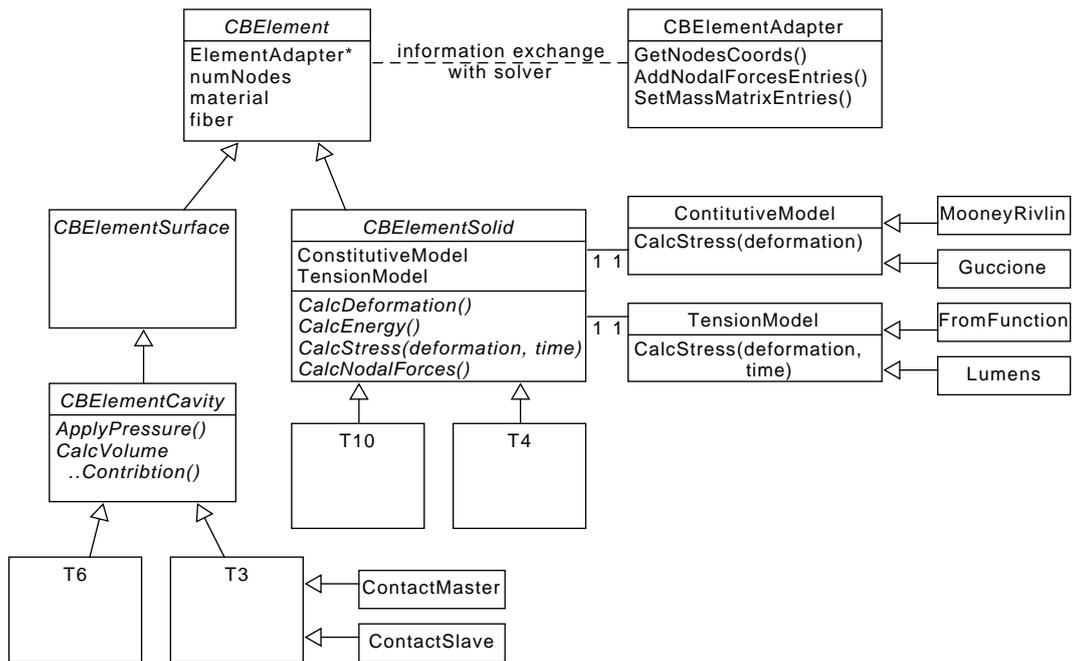


Figure 4.2.: Existing Elements in the CardioMechanics Framework and their interaction with the solver.

4. Implementational Aspects

- reach specific steps exactly.

It is important to hit certain points in time exactly, such that periodic export and the final simulated time are predictable and exactly comparable. Failing time steps usually compute a larger number of Newton iterations, so these are way more expensive than successful steps and need to be avoided. Therefore, the step size should be allowed to increase only when a certain number of successive steps were computed successful. A proper handling of all solver return codes (SUCCESS, FAIL and REPEAT) was done, to be compatible with the plugins of the solver structure that are doing iterative computations (circulatory system, unloading, see Section 4.1).

The developed algorithm needs only two internal variables, a floating point number containing the current “step size” and an additional integer-valued “counter”. The counter variable changes by a factor of two each time the step size changes, and increments by one if the step size does not change. The design of the counter variable facilitates to check if there is a corresponding point in time one level above, or if an increase would lead to a non-fitting time “in-between”, by checking the last bit of the binary representation.

The developed step size control algorithm works as follows: In the *REPEAT* case, all internal variables stay untouched for the next solver run. In the *SUCCESS* case, the counter increments by one and gets subsequently checked if an increase of the step size is possible for the next solver run as shown in Figure 4.4. For a *FAILED* case, the step size reduces by a factor of two and the counter variables doubles by the same factor for the next solver run.

In a non-optimal scenario, the chosen initial step size is slightly too large and never yields a solution, but every step on the reduced level is successful. The algorithm needs to stick to the step size one level below, periodically checking if the larger step size is possible. Without modification, the overhead of additional steps is 50% (one failing large step followed by two successful small steps). In reality, the computational overhead is higher since time steps take longer to fail (tolerance not met after a maximum number of Newton steps) than to succeed (tolerance met before that limit). Therefore, a certain number of successful steps on that specific level should be enforced to avoid trying the possibly failing larger step size too often. Introducing a modulo division (check e.g. only each fourth step) by an additionally given factor “MinSteps” mostly enforces that behavior and reduces the overhead in this scenario to 25%.

Picture 4.3 depicts an example case in which only two successful steps already caused trying a larger step size, despite “MinSteps” being set to four. The exception happens only after an immediate reduction, if a failing step followed by two successful steps leads to a modulo divisible counter variable, even though the last failing step size is less than four steps away. For these cases, an additional counter “lowestLevelCounter” was added to ensure a minimal number of steps also on the lowest level. The *lowestLevelCounter* needs to be active only after an immediate reduction of step size. After an increase, the necessary number of successful step sizes was realized with the help of smaller steps, so it is likely that a larger step size could work successfully.

An additional parameter enables the *fastRelaxation* mode, that allows to increase by multiple levels at once. It is useful for simulations where only single points in time need

4. Implementational Aspects

Table 4.1.: Maximal number of representable time steps depending on the chosen counter variable type. “Relative steps” gives the number of max time steps that can be represented when using the smallest time step. “Absolute time” assumes a typical configuration with min and max step size of $[1\text{E} - 3, 1\text{E} - 8]$, run exclusively on the level of min steps, and “Heartbeats” assumes a frequency of 180 bpm or 0.333 s to reflect a worst case scenario on purpose.

variable type	bits	value range	relative steps	absolute time	heartbeats
int	16	$[-2^{15}+1, 2^{15}-1]$	$2^{15}-1 \approx 3.28\text{E}4$	0.326 s	0.98
unsigned int	16	$[0, 2^{16}-1]$	$2^{16}-1 \approx 6.55\text{E}4$	0.655 s	1.97
unsigned long long	64	$[0, 2^{64}-1]$	$2^{64}-1 \approx 1.84\text{E}19$	1.84E14s	5.52E14

a small step size, but can be an unwanted effect if larger step sizes are likely to fail, because all step sizes up to maximal time step length are re-checked at each export step.

Numerical considerations When the number of levels is large, the counter variable might exceed numeric limits during step size reduction. Since computers data types typically stick to a fixed number of bits, the maximal representable value of the counter must be kept in mind. With the default *int* type (only 16bit guaranteed), this can happen already at $1/(2^{16}) \approx 1.53\text{E} - 5$ times the initial time step size, or 16 relative levels. Typically used step size limits (in s) are $1\text{E} - 2$ to $1\text{E} - 3$ for a normal step size and $1\text{E} - 8$ for the smallest step size, covering a range of $1\text{E}6 \approx 2^{19.93}$, or 19 possible relative levels. Computing more than one step at normal step size, or even multiple beats increases the needed range of the counter variable further. To represent 1 s of simulated time (one heart beat at rest) with a normal step size of $1\text{E} - 2$, the counter value thereby can go up to $100 \cdot 1\text{E}6 \approx 2^{26.58}$ for one heart beat, in the very last step at the smallest possible step size. *Unsigned long long* was used instead of *int*, which is according to the C++11 standard at least 64 bit, the largest representable value is $2^{64} - 1 \approx 1.84\text{E} + 19$. This corresponds to $2^{37.43}$ heartbeats when using the typical limits from above, and is enough even for simulating hundreds of heart beats as they occur to reach quasi-static periodic beats. For comparison, the heart of a 100 year old person (5.26E7 minutes) only undergoes at most $9.48\text{E}9 \approx 2^{33.14}$ beats at an extremely fast heart rate of 180 bpm – which is representable by the counter even at the lowest step size.

Computations of the variable for the step size in contrast is always exact due to the internal base-2 characteristic of the *double* precision floating point representation (step size = $\pm a \cdot 2^b$, 1 sign bit, 52 bit for the mantissa *a*, 11 bit for the exponent *b* = 64 bit). Multiplication and division with two only change the exponent *b* by one, which needs only 6 of the available 11 bits for 32 level reductions.

Moreover, the implementation allows to start with a smaller than maximum step size, that automatically increases over time until maximal time step length gets reached. This is a big benefit for ventricle inflation simulations. These have a volume change that is large in the beginning (small step size needed) and decreases rapidly (large step size possible) when the volume approaches the final volume. Typical use cases that benefited

4. Implementational Aspects

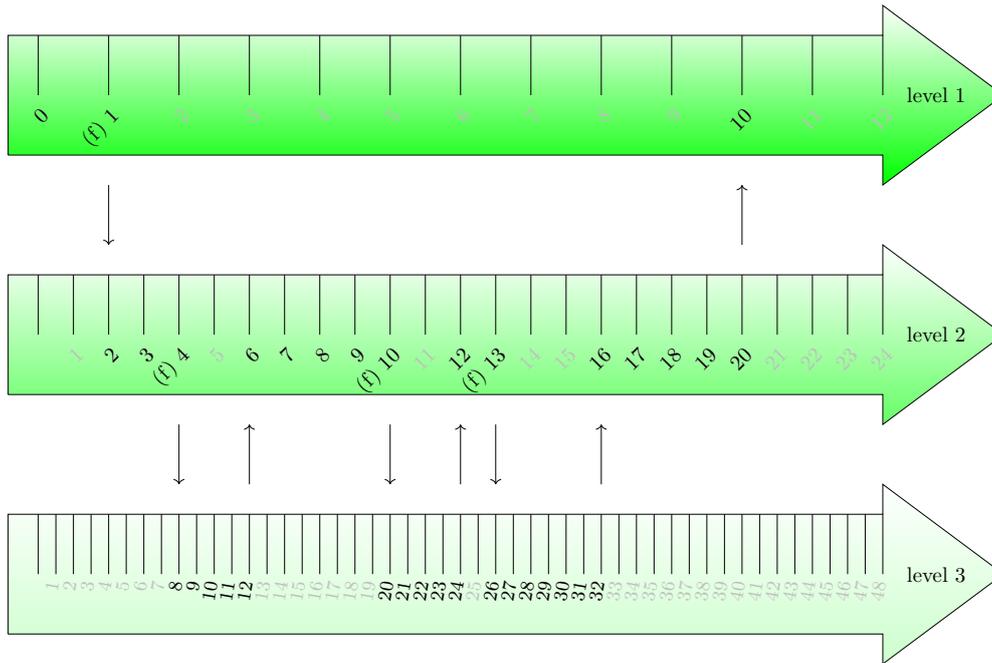


Figure 4.3.: An example case of a sequence of successful and unsuccessful (noted with an ‘f’) time steps which makes it necessary to introduce the additional variable *lowestLevelCounter*. The large arrows visualize the possible steps at three different step size levels, the black numbers denote the contents of the step counter variable. In the shown case, a value of 28 (in level 3) should not cause a change to 14 (in level 2) albeit 28 is a multiple of 4, because just two successful steps were computed on level 3.

a lot from such a behavior are the determination of unloaded state (Section 3.10), and of passive material parameters (Section 6.1.4), that require both a large number of ventricle inflations.

4.3. Coupling of Multiple Systems

Most of the more complex models for tension development or the circulatory system base on a set of ordinary or partial differential equations needing input from the current deformation. Since the result of the elasticity equations depends on these components as well, coupling is necessary. The choice of the type of coupling has great impact on the computational overhead, but on the credibility of the results as well.

The term *coupling* denotes the combined solution of two or more systems of equations, based on the fact that the result from one system is needed as input for the others, and vice versa. Different specific strategies for coupling in CardioMechanics exist:

The *weak coupling* strategy first solves one system and transfer the results as input or boundary condition to the other system. Drawback is that no control of the error between

4. Implementational Aspects

```
// advance to next iteration, increase timestep if possible
else if (shallIncrease) {
    counter_++;
    lowestLevelCounter_++;
    bool didIncrease = false;

    // modulo divisions ensure ony fitting steps
    while (counter_%2 == 0 &&
           counter_%minNumCnt_ == 0 &&
           lowestLevelCounter_ >= minNumCnt_) {
        counter_ = counter_/2;
        if (timeStep_ < maxInitTimeStep_) {
            timeStep_ *= 2;
            // fast relaxation: increase by multiple levels if possible
            if (!fastRelaxation_)
                break;
        }
    }
}
```

Figure 4.4.: Code to check if an increase of the step size is possible. This only needs to be done in case of the solver return code SUCCESS (`shallIncrease=true`), and ensures fitting points in time (`counter`), maximal step size (`maxInitTimeStep`), a certain number of consecutive steps (`minNumCnt` and `lowestLevelCounter`), and the fast relaxation mode (`fastRelaxation`).

4. Implementational Aspects

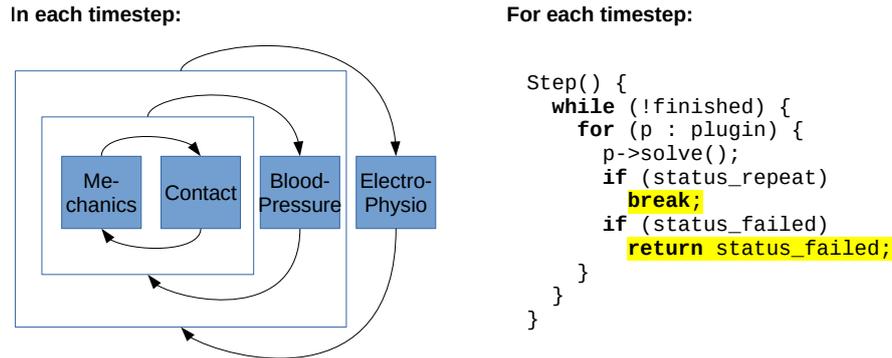


Figure 4.5.: Strong partitioned coupling of two and more systems (left) using the *Solver-Plugin* structure. Each plugin needs to accept the found solution before the next time step can be computed.

the two systems is possible. In CardioMechanics, active forces or cavity pressures can be read in from a pre-computed file (using `TensionFromFile` and `PressureFromFile` from Figure 3.4) if feedback is not necessary or desired, and was the method of choice by Fritz et al. (2014) for ‘coupling’ with active forces.

Strong coupling strategies can further be subdivided into the implicit, monolithic approach and the explicit, iterative or partitioned approach. In the *monolithic approach*, the discretized equations of all systems are collected in one common system matrix and solved simultaneously. While this seems wise because no additional specific coupling iterations are needed, there is no isolated control over the coupling error possible. Instead, the tolerance of the solver ensures that the residual error of the overall system is small, which however consists of an average value of each system like computed displacements and pressure differences besides the coupling error. This approach is realized in CardioMechanics for the ContactHandling (Fritz et al., 2014) by filling the `ApplyToNodalForces()` and `ApplyNodalForcesJacobian()` functions. They observed the gap (below 0.15 mm) to be small enough to retain “practically permanent contact”. The *partitioned approach* uses a separate solver for each system and iterates between the systems by explicitly minimizing the coupling error using e.g. fixpoint iterations or a Newton method. Thereby the coupling error is under direct control, and, moreover, highly specialized methods can be used to solve each of the separate systems (e.g. specific preconditioning, different time step lengths, explicit/implicit solvers, parallelization strategies). This strategy can be realized in CardioMechanics by implementing a *CBSolverPlugin* with appropriate return codes, solving the system in the `Apply()` method and returning `SolverStatus::REPEAT` until a tolerance is reached (see Figure 4.5). One example of this is the newly developed circulatory system plugin, which allows an exact control of the volume difference between the models for elasticity and hemodynamics to be permanently below 1 mm^3 .

4.4. Parallelization

Parallelization of the code consists of two main aspects. First, distributed storage of the nodes and elements. Nodes are equally distributed over the processors, whereby nodes corresponding to the same element are assigned to the same processor. Each processor computes only the contribution of the elements it has access to. Other element-wise aspects like tension and constitutive models inherit that parallel capability. This way, assembly of the matrices could be implemented in parallel. Computation of global values that need information from all nodes, needs to be implemented explicitly using the message passing interface (MPI). One example for this is the computation of cavity volumes. Each process can calculate only the partial volumes obtainable from its own data, sends that volume contribution to the first process, where it gets summed up afterwards using MPI's send, receive and reduce routines.

Decomposition of the domain can be done either by the node index, by a principal components analysis (PCA) of initial node coordinates, or using information from the stiffness matrix of the first time step as connectivity matrix (Figure 4.6). The latter tries to minimize communication cost between less-related nodes and becomes relevant when using more than a single-digit number of cores.

Second, solving the equation systems relies on an external library (PETSc). That library provides distributed and sparse data types for vectors and matrices, and an interface to a number of “scalable nonlinear equation solvers” (SNES) like Newton, quasi-Newton and multigrid methods. PETSc was used with one of the parallel external libraries “mumps”, “superlu_dist” and “pastix” for distributed solving the arising linear equation systems with a direct solver. “Mumps” usually showed the fastest performance on single and multicore, however on multiple cores it produced slightly different and non-reproducible results on each run. PETSc itself allows to use both, distributed memory (message passing interface MPI, for communication between nodes) and shared memory (OpenMP, pthreads, for communication between processes within a node) approaches. On larger meshes, available memory becomes the limiting factor and the arising linear equation systems can only be solved with iterative approaches. Efforts were undertaken to find optimized parameters for iterative solvers using the SMAC library, however a combination of an iterative solver and parameters working faster than a direct solver could not be found.

Parts that are not implemented in a parallel manner are data reading, model initialization and data export, which needs to be done only once or a few times per simulation.

4.5. Geometric Model Generation

Besides these described parts of modeling used to create the equations, there is yet another numerical aspect that becomes important when trying to solve these equations using a computer. Mesh generation of the geometry is an essential part of the spatial discretization and solving the model equations in the real application. The heart undergoes large deformations and a good mesh quality needs to be kept throughout the whole heart

4. Implementational Aspects

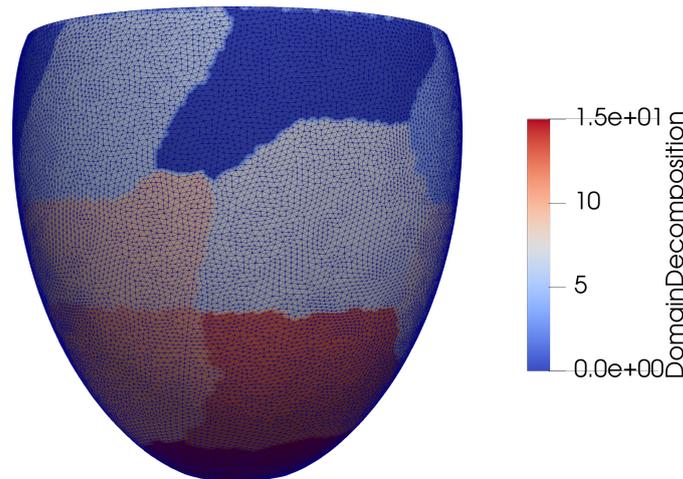


Figure 4.6.: Automatic domain decomposition of a high resolution left ventricle (850k nodes, 150k linear elements) into 16 sub-domains using stiffness matrix as connectivity information.

beat. The type of spatial and temporal discretization, higher convergence orders and an efficient use of space- and time resolutions can help with the needed to exploit the always limited computational resources. The choice of the spatial discretization however is also bound by the feasibility to generate meshes for it.

Deviations between the computer model and measurement data can have their root in both parts. Highly complex model equations in theory give correct results. These can in reality be only achieved with a good-quality mesh, sufficiently small step sizes in space and time, and a capable numerical solver. On the other hand, computational demands rise quickly with model and discretization complexity. The problem is not only limited by the available computational power, but – even worse – from the computational perspective the arising matrices lose their ability to be solved. From a specific spatial resolution onwards, appropriate algebraic reformulation becomes necessary (e.g. by problem-specific preconditioning) to get the matrix condition within numerically solvable limits.

Details about the model generation is usually not mentioned in scientific works, but still an essential and time consuming part. Since computation time is a factor during development, mesh size (total number of nodes) plays an important role. Aim is usually to get tetrahedron edges of equal length, the quality of the worst tetrahedron decides if the system is solvable or not. The model gets continuously deformed by a considerable amount throughout a heartbeat, so not only sufficient mesh quality of the initial tetrahedral mesh, but also all deformed states needs to be pertained. This results often in a large number of mesh generation–simulation iterations. Typical characteristic lengths of a four-chamber heart are 3–10 mm for the edges of a second-order tetrahedron (two to three elements transmurally in the LV), resulting a mesh containing 16.5k cells and 31.7k nodes for all four chambers. The final full heart model including ventricles, atria, valve plane, and truncated veins and surrounding tissue contains 24k cells and 44k nodes.

4. Implementational Aspects

Figure 4.7 gives a rough guidance of the work flow and the involved data formats. The general process chain is as follows: medical images (DICOM) → segmentation → volume mesh → surface mesh → tetrahedral mesh → tetrahedral mesh with material and fibers. A short overview of the used tools and their purpose in the meshing process:

- *ITK-SNAP*. Create surface meshes of the endocardial surfaces (blood volume) and – if visible in the MR images – epicardial surfaces. *ITK-SNAP* has an option to use active contours with a classification-based pre-segmentation, exploiting intensity, neighboring intensity, intensity gradients and patterns of labeled pixels (Yushkevich et al., 2006) . This allows a quick processing of image data by demanding only a small number of partly labeled slices (‘some’ blood, ‘some’ myocardium, ‘some’ background, in 10-20 slices) as input data for the algorithm. Figure 4.8 gives a visual example of such a segmentation and the provided input data.
- *Blender* is a tool that allows modification of triangular surface meshes (Blender Online Community, 2017) . It is used to modify the segmented surface meshes, apply surface thickness to blood-only meshes (atria), define surfaces for boundary conditions like Dirichlet boundary, pressure boundary, cavity volumes and contact interfaces.
- *Instant-meshes* is a tool for automatic coarsening or refining of surface meshes to a desired target triangle size, while maintaining topology and improving quality of the triangles all at the same time (Jakob et al., 2015) .
- *Tetgen* and *Gmsh* are both tools that allow to convert triangular surface meshes to tetrahedral volume meshes. Both do not modify given surface meshes, making an input mesh of good triangle quality essential. Since the myocardial thickness is in a similar magnitude as the targeted mesh resolution size, the final number of nodes/elements is mostly determined by the resolution of the surface mesh. *Gmsh* can be used as CAD tool to create volumetric meshes of good quality from triangular meshes as well (Si, 2015) . In the latter case it needs closed, non-manifold and non-overlapping surfaces of one single material, each. In order to allow manual merging of the meshes in an additional step with a separate tool, all triangles and nodes of the triangles between the material interfaces need to match. *Tetgen* accepts also non-manifold surfaces and assigns material number automatically to each cavity (Geuzaine and Remacle, 2009) . This was used to create truncated veins consisting of three surface, with one matching endocardial, epicardial and truncation area, each.
- *Python*, *Matlab*, *C++* are programming languages (van Rossum, 1995; Moler et al., 1982; Stroustrup, 2013) used for self written code, mostly in combination with the *VTK library* (Schroeder et al., 2006). Tools exist for data conversion (tetgen to vtk format and vice versa), setting of Dirichlet boundary conditions (node fixation), finding node correspondents between a Blender-extracted stl surface and a tetgen-based volume mesh (e. g. for defining cavity surface triangles), conversion between first and second-order tetrahedra, generation of fiber orientations, etc.

4. Implementational Aspects

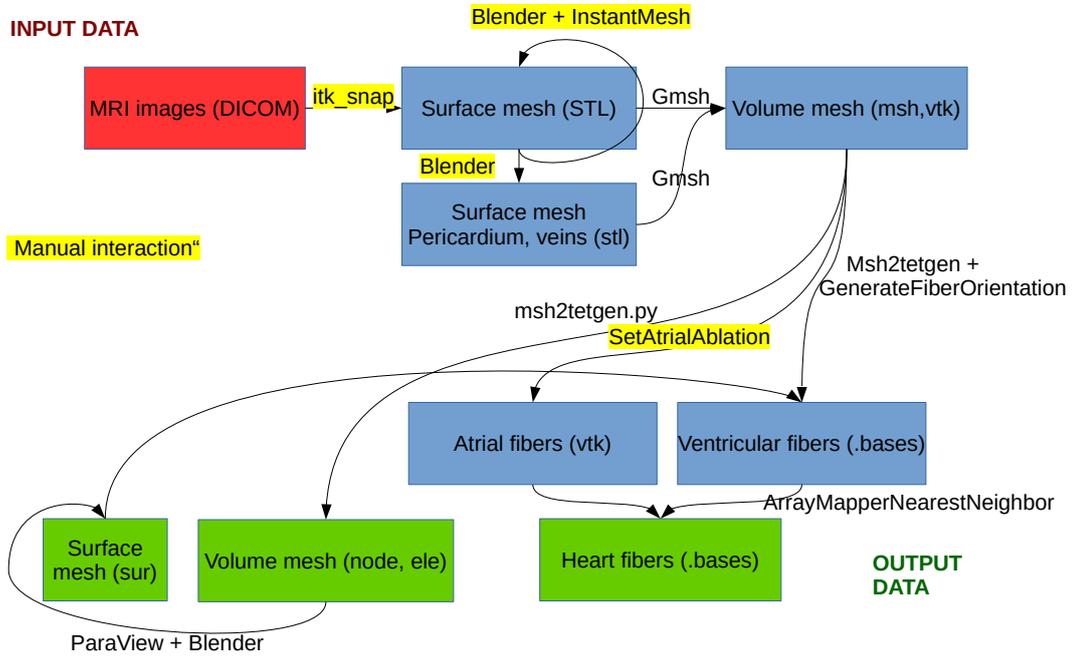


Figure 4.7.: Overview of the mesh generation workflow. Input data is in red, output data in green and steps involving manual interaction (non-scriptable) are highlighted with a yellow background.

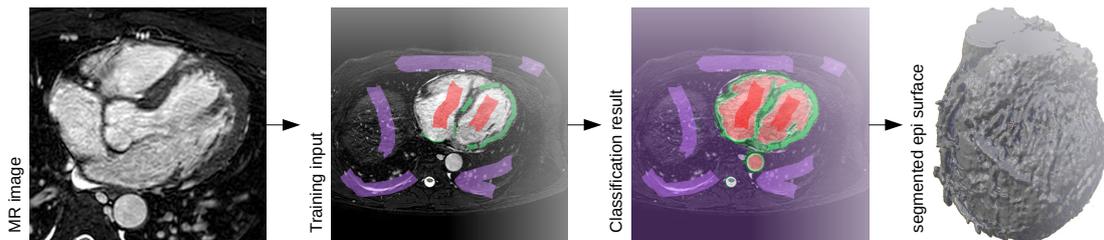


Figure 4.8.: Example of a classification-based segmentation. For the training input, parts of blood, myocardial and background volume were labeled in red, green and lilac (left) and result in a highly-detailed surface mesh containing endo- and epicardium (right).

4. Implementational Aspects

For all the simulations in this thesis, two different geometries were used. One contains an analytical description of an ellipsoid and resembles a left ventricle, the other geometry contains all four chambers of the heart and is based on MRI data of a real person.

Geometry of the left ventricle This is the same geometry that was used for code validation of ventricle inflation and ventricle inflation under active contraction in (Land et al., 2015). Myocardial volume is defined using the parametrization of an ellipsoid

$$\mathbf{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_s \sin(u) \cos(v) \\ r_s \sin(u) \sin(v) \\ r_l \cos(u) \end{pmatrix}$$

with ranges for radius in short axis direction $r_s \in [7, 10]$ mm, in long axis direction $r_l \in [17, 20]$ mm, azimuth angle $v \in [-\pi, \pi)$ and an inclination angle u such that the ellipsoid truncates at the base plane $z = 5$ mm or $u \in [-\pi, -\arccos(r_s/r_l)]$. Fiber orientation were defined from -90° at epicardial to $+90^\circ$ at endocardial surface using derivatives of the ellipsoid parametrization. Such an analytically described geometry had the big advantage that mesh generation could be fully automated – with mesh resolution as the only parameter – and allowed to generate reproducible meshes of different size useful for a number of investigations (mesh convergence, necessary resolution, parallelization, domain decomposition, limiting parts of the solver).

Geometry of the whole heart The geometry used for simulations of the whole heart are based on the work by Fritz et al. (2014); Fritz (2015). It is based on the segmentation of a 28 year old healthy male provided by the University clinics of Heidelberg. Three different meshes (Figure 4.9) were used and are the result of certain improvements over the time: The 1st version is the one used by Fritz et al. (2014) and Fritz (2015), and contains the myocardium of all four chambers. The resolution “through-the-wall” is one element in the atria and the right ventricle, and between one and two elements in the left ventricle. Dirichlet boundary conditions were applied at the orifices and ventricular apex. The 2nd version adds truncated representation of aorta and veins to the 1st version in order to allow some range of movement for the atrium. Dirichlet boundary conditions were moved away from the orifices to the more remote ends of the “veins”. The 3rd version is a complete rework of the second version to overcome many of the previous geometry’s limitations, sharing only the same segmentation:

- Added volumetric representation of the valve plane, to allow better interaction between atrial and ventricular pressure difference on the motion, and to obtain pressure surfaces that are closed and thereby do not generate an overall force towards the apex only due to “applying a pressure”.
- Added volumetric representation of fat around the valve plane, such that myocardium + fat have a convex outer shape.

4. *Implementational Aspects*

- Introduced a two-level truncated representation of the veins, with the intermediate plane fitting the epicardial surface,
- Use a closed, convex surface of master elements without any holes to vastly improve reliability of the contact pericardium algorithm. This is only possible due to added fat and fitting intermediate planes at the orifices from the previous two points.
- Overall, the mesh has slightly more elements, and especially in the left ventricle an increased resolution of two to three elements through the wall.

All three meshes have a tightly-fitting volumetric mesh of roughly 2 cm thickness surrounding the myocardial mesh to represent pericardial layers and surrounding tissues. In the 1st and 2nd version of the mesh, the contact master surface was not closed (orifices, fat) and cause arbitrarily failing simulations for some combinations of parameters, whilst other actually more demanding parameters (e. g. larger time step size, weaker tissues, larger pressures) worked fine.

4. Implementational Aspects

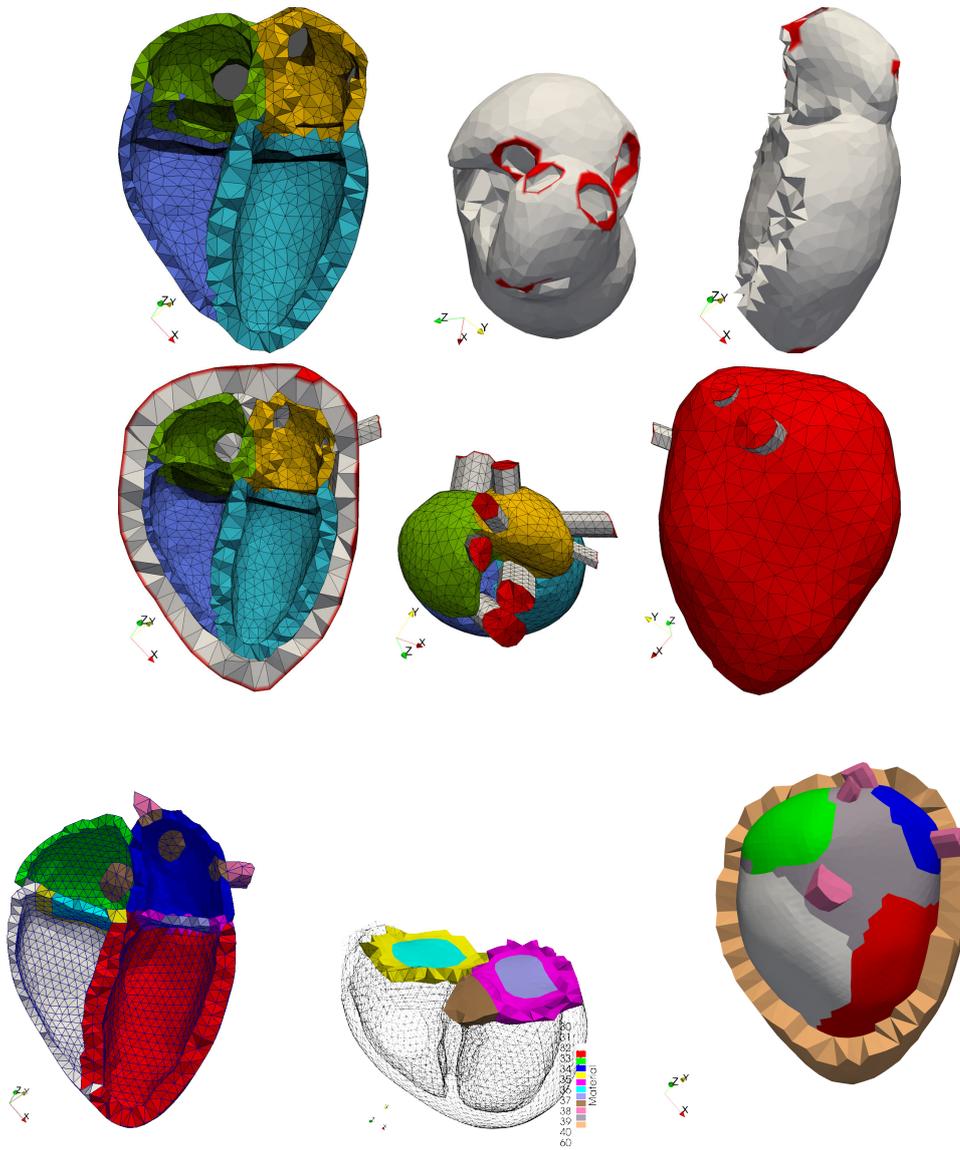


Figure 4.9.: Evolution steps of the four-chamber mesh. The top row shows the version used by Fritz et al. (2014) with node fixations in red, the middle one has additional truncated representations of the veins, and the bottom mesh is a complete rework at the same patient with a different surface grid, a volumetric representation of the valve plane, the addition of fat for a convex contact surface and slightly higher resolution. The pericardium layer is available in all three meshes.

Part II.

Approaches and Results

5. Validation of the FEM Solver

One of the most important aspects of numerical solutions is validation. This chapter presents investigations to clarify the technical limitations of the numerical methods (discretization errors) that are independent from the limitations of the models (modeling errors). Thin walls surrounding comparatively large compartments are well known in civil engineering (e.g. silo walls) and construction engineering (e.g. ship hulls) as shell-type problems Belytschko et al. (2014). These are known to demand a large number of elements to get a sufficiently accurate numerical solution.

The first section answers the question of needed mesh resolution to get a confidential solution, evaluating simulation results on the same geometry at different resolutions in order to show mesh convergence. The difference in the solutions between the meshes gives a hint what resolution is needed to get in the range comparable to image resolution of current MRI acquisition. Due to the three-dimensional structure, improving the mesh resolution by a factor of two increases the number of unknowns and hence the computational effort drastically.

In the second part therefore, the parallel performance of the solver is evaluated. Amdahl's law makes a high fraction of parallel execution time a necessity to obtain a significant speedup. A scalable code does not only reduce computation time for a specific mesh size, but also allows to increase the problem size to obtain sufficiently fine resolutions. This allows proper use of several hundred processing units available on high performance computing clusters.

Besides the results presented in this chapter, additional validation was done by the author of this thesis and published in the publication Land et al. (2015) by means of three simple heart-related test cases on an ellipsoidal ventricle (inflation, active contraction, inflation and active contraction). Additionally, Fritz (2015) showed a convergence analysis of linear and quadratic tetrahedron on a bending beam with the initial implementation of the finite element code.

5.1. Necessary Mesh Resolution for Mesh Convergence

5.1.1. Introduction

Consistent results are one of the major properties that a numerical solution should provide, independent of the specific method or the applied method parameters. Mesh resolution plays a special role, it has a large influence on computation time, while the accuracy of the result still is vastly influenced by the type of finite element chosen for discretization. Due to the complexity of the meshing process, anything but simple geometries is a domain mostly exclusive to tetrahedral meshes. A still open question in

5. Validation of the FEM Solver

cardiac computation is the necessary mesh resolution to reach a sufficient numerically accurate solution.

Most of the existing groups participating in the comparison paper by Land et al. (2015) on the field of cardiac deformation use either a high resolution mesh with low order elements (high memory demands) or a lower resolution mesh with higher order elements (high implementation demands) (Augustin et al., 2016; Gurev et al., 2015). From the modeling perspective, high resolution meshes have the advantage to represent physiology in detail like transmural fiber rotation, material inhomogeneities on a small scale (fibrotic tissue, scar tissue, infarcted regions) and the geometric shape itself. From the computational perspective, this makes certain non-trivial points a necessity: parallelization to be able to use enough computation power, matrix preconditioners to get solvable equation systems, or iterative solvers to better cope with memory limitations. All this still goes at the cost of a longer computation time, and a more difficult code development and debugging cycle.

Even the geometry creation process needs to be carried out with more care, accuracy and elaborateness to take real advantage of the higher mesh resolution. This also prohibits to exclusively rely on model-based methods that usually provide ‘general information’ averaged from multiple individual patients via a template mesh. Currently there are two major use case scenarios, that would benefit from an adequate accuracy to computation time ratio: Large scale study on a significant number of patients, and model parameter estimation. Both need a large number of forward calculations to draw reasonable conclusions with confidence.

5.1.2. Methods

This study focuses on the numerical mesh convergence, so the design keeps the geometrical influence as small as possible. A truncated parametrized ellipsoid is used to define the geometry of a left ventricle, it is the same as used in “Problem 2” in Land et al. (2015):

$$\vec{x} = \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r_s \sin(u) \cos(v) \\ r_s \sin(u) \sin(v) \\ r_l \cos(u) \end{pmatrix}$$

Myocardium is defined as the volume between endocardial surface ($r_s = 7$ mm, $r_l = 17$ mm, $u \in [-\pi, -\arccos \frac{5}{17}]$, $v \in [-\pi, \pi]$), epicardial surface ($r_s, r_l, u \in [-\pi, -\arccos \frac{5}{20}]$, $v \in [-\pi, \pi]$) and the base plane ($z = 5$ mm). The material properties are chosen to obey the Guccione material law

$$W = C \left(e^{Q(\mathbf{E})} - 1 \right)$$

$$Q(\mathbf{E}) = b_1 E_{11}^2 + b_2 (E_{22}^2 + E_{33}^2 + E_{23}^2 + E_{32}^2) + b_3 (E_{12}^2 + E_{21}^2 + E_{13}^2 + E_{31}^2)$$

with the isotropic material properties $C = 5$ kPa, $b_1 = b_2 = b_3 = 1$. The isotropic property of the chosen material parameters removes the influences by more detailed fiber representation, while the nonlinear Guccione material law still reflects the specific

5. Validation of the FEM Solver

Table 5.1.: Number of nodes and elements (T10) for the analytical left ventricle at different resolutions. Additionally, computation time ([hh:]mm:ss) is given for each simulation. The mesh with resolution 16 was aborted after 48h without solution.

resolution	1	2	4	8	16
nodes	1.280	4.235	23.601	152.004	1.107.874
elements	1.013	3.408	18.613	117.889	846.673
cores	4	4	4	16	64/256
time	1:42	2:57	39:28	9:36:50	–

non-linear passive behavior of heart tissue. The mesh was generated using Gmsh, which takes the parametric description as input and generates accurate meshes of arbitrary resolution with good triangle quality using one parameter controlling a characteristic discretization length. Second order tetrahedra (T10) can be generated directly within Gmsh, which means that non-corner nodes are located on the actual surface, rather than being averaged from adjacent corner nodes as when converted from linear elements. As solver, the static solver gets used, which ensures completely converged final states by omitting time-dependent effects like damping or mass inertia. The ventricle gets inflated to a pressure of 10 kPa in 100 equally-spaced pressure increments. Four meshes of different resolutions were generated, characterized by 1, 2, 4 and 8 transmural elements. Doubling the resolution in all three dimensions means that the number of nodes is expected to increase roughly by a factor of eight, which only becomes prevalent at the higher resolutions due to the shell-type structure. Table 5.1 gives information about the used meshes and the number of computing cores and computation times. As reference of the computational effort, with linear elements (T4) the resolution 8 setup took only around 10 minutes instead of more than 9 hours (T10), however its number of nodes is significantly lower and more comparable with the second-order resolution 4 mesh. All simulations were carried out at the super computing center of KIT on the uc1 cluster and took between 2 minutes and 10 hours for the successful simulations.

5.1.3. Evaluation criteria

Resolution is defined as the number of transmural elements through the ventricular wall, which can be controlled using a discretization parameter during mesh generation with Gmsh. Aim of this study is to quantify the numerical error of different mesh resolutions and answer the question of needed transmural elements for an accurate representation. The used evaluation criteria thereby are designed to measure common key values in the simulation of cardiac mechanics.

1. The *apex z position*, defined as the z-coordinate of the point with the initially most negative z-coordinate,
2. the *enclosed volume*, obtained by integrating over the surface using Gauss law and a density of $\rho = 1$, and

5. Validation of the FEM Solver

3. the *surface distance* between two different resolutions, determined by the vtk filter `DistancePolyDataFilter` which defines an implicit function to allow computing distances between a surface and arbitrary points in 3d space. It measures the distance from each point on the lower resolution mesh to the higher resolution mesh. These distances are then linearly interpolated to the triangles in-between, and integrated over the whole surface domain using first-order Gauss quadrature.

Certain of these values have special meaning: The displacement of the epicardial apex reflects the displacement of the valve plane. At the same time, it is the most remote point from any boundary condition in this simulation and serves an upper limit of the positional error. The enclosed endocardial volume represents the chamber volume and is an important factor when coupled to a circulatory system model or used during parameter adaption to the Klotz curve. The epicardial surface distances represent the error of approximating the outer shape, important when coupling to a pericardium representation via a surface contact handling.

All evaluation criteria are measured on the initial mesh, describing the principal or minimal reachable error through a discrete representation of the geometry (“initial”), and after inflation, which describes the error introduced by the finite element formulation (“inflated”). Since the exact solution after inflation is unknown, the error of each criterion is determined twice, once between two consecutive mesh resolutions, reflecting the accuracy gain by choosing the next higher resolution (denotes as “change”), and once with respect to the finest computed resolution 8 as an approximation to the absolute error (“error”).

5.1.4. Results

Generally, all criteria were observed to decrease with increasing number of transmural elements. The relative “change” values were in all criteria slightly below the absolute “error” values, but in the same order of magnitude.

Figure 5.1 shows the errors of the apex displacement in z-direction. Before inflation, the endocardial and epicardial apex positions are directly given by the ellipsoidal description and are represented exactly (-17 mm and -20 mm), hence in both cases the discretization error is 0.0 mm. After inflation, the maximum change in apex position starts at 0.11 mm between resolution 1 and resolution 2 for both, endo- and epicardial apex position. Figure 5.2 focuses on the enclosed volumes. Here, an effect of the discretization becomes visible, and the volume change is with 0.04 ml on the epicardium larger than on the endocardium with 0.03 ml between the meshes of resolution 1 and resolution 2. After inflation, the volume change is the largest on the endocardium and starts at 0.25 ml, but quickly drops to 0.05 ml between resolution 2 and resolution 4, which is the discretization error of the resolution 1 mesh. Figure 5.3 gives the distance errors on the endo- and epicardial surfaces. Here again, an effect of the discretization becomes visible and is most pronounced on the endocardium, starting at 0.011 mm between resolution 1 and 2, and reducing to 0.004 mm between 2 and 4 and further down to 0.001 mm between 4 and 8. After inflation, the largest distance change is observed on the epicardium with 0.044 mm

5. Validation of the FEM Solver

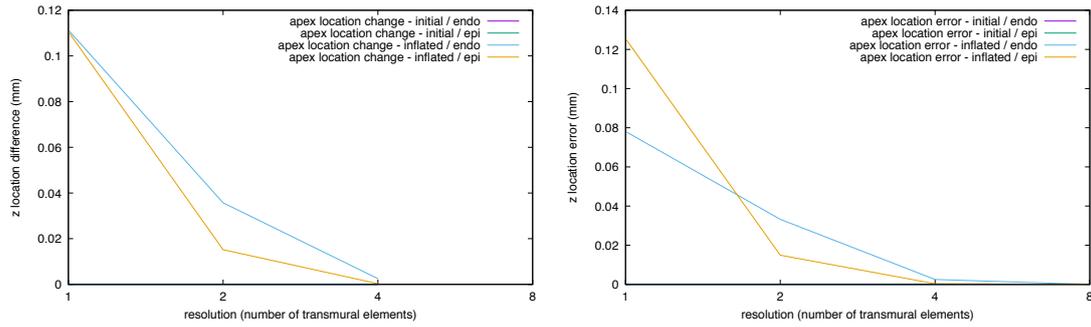


Figure 5.1.: Apex location on the long axis over mesh resolution. Shown is the difference in apex positions when switching to the next higher mesh resolution (“change”, left) and to the highest available mesh resolution (“error”, right).

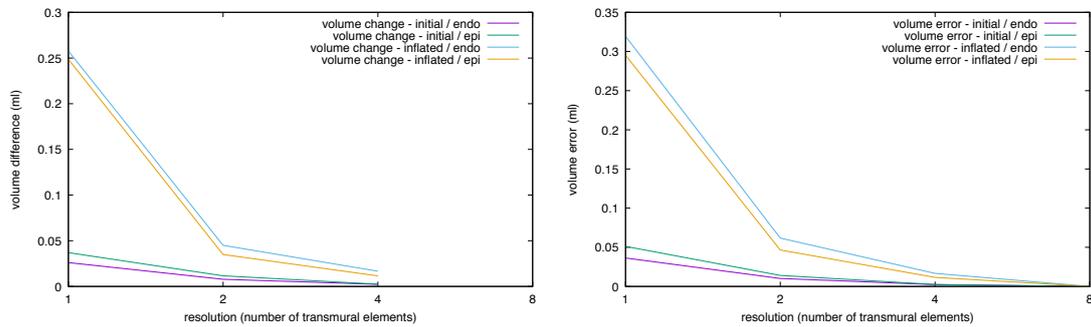


Figure 5.2.: Enclosed volume over mesh resolution. Shown is the difference to the next higher mesh resolution (left) and the highest available mesh resolution (right).

between resolution 1 and 2 and reducing down to 0.004 mm between resolution 4 and 8, the same as the discretization error of the coarsest mesh. Interestingly, the distance error – measured in relation to the resolution 8 mesh – appears to be negligible, which was not the case for the criterion of distance change.

Absolute values on the finest computed mesh are given as follows. The inflated apex positions are located at -26.6174 mm on the endocardium and -28.2829 mm on the epicardium. The absolute enclosed volumina before inflation are 2.4914 ml on the endocardium and 5.7260 ml on the epicardium, and the absolute volumina after inflation are 10.7297 ml and 13.9629 ml.

5.1.5. Discussion and Conclusion

The dimensions of the ventricle were chosen according to the analytically described geometry from the benchmark paper Land et al. (2015). Its original purpose was to define a simple-to-describe common geometry useable by a number of groups. Its size however is significantly smaller than a human heart. The ellipsoidal ventricle has a volume of 2.5 ml before inflation, and represents probably rather a rat or rabbit heart than a human

5. Validation of the FEM Solver

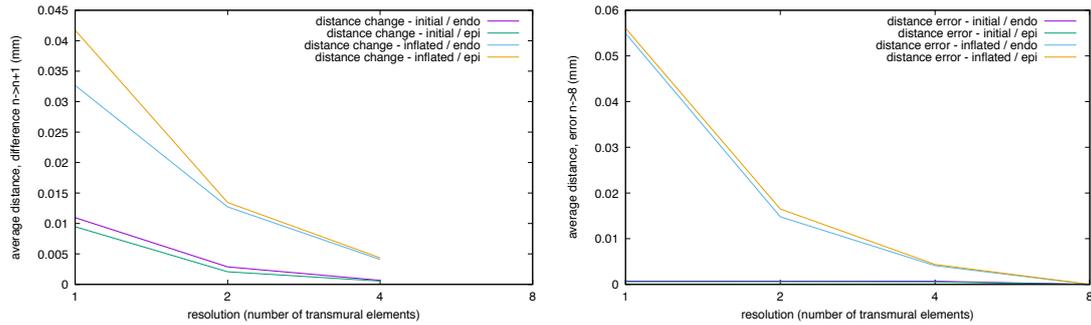


Figure 5.3.: Epicardial surface distance over mesh resolution. Shown is the difference to the next higher mesh resolution (left) and the highest available mesh resolution (right).

ventricle. Therefore, the results can not directly be applied to human heart simulation and need conversion to allow an adequate interpretation. Simulations from Section 6.3 showed 60-70ml ESV and 130-160ml EDV e.g. in Figures 6.31, 6.32 or 6.33. Assuming a human ventricle volume of 60 ml before inflation and 120 ml after inflation, the volume is about 17 times larger and the dimension about 2.6 times the size in each direction. Since the FEM solver's convergence criteria internally work with relative errors, scaling of this study's results with the respective factors is viable.

As a reference, the MRI data set used for segmentation had a resolution of $0.72 \text{ mm} \times 0.72 \text{ mm} \times 1.8 \text{ mm}$ per voxel, which will be considered as limit for the geometric error (due to discretization, before inflation) and sufficient for the numerical error (due to approximation, after inflation). A simulation staying significantly below that threshold would pretend to contain information that is actually not available due to the characteristics of the input data. Already the resolution 1 mesh reaches an apex location change in a similar range of $0.11 \text{ mm} \cdot 2.6 = 0.29 \text{ mm}$ after inflation, and an inflated volume change of $0.25 \text{ ml} \cdot 17 = 4.25 \text{ ml}$ on the endocardium (which is there the largest). However, a change from the mesh of resolution 2 to resolution 4 was needed to get the apex location change an order of magnitude below the reference ($0.036 \text{ mm} \cdot 2.6 = 0.094 \text{ mm}$ on endocardium after inflation). In order to obtain an accurate apex position, thereby a mesh with a resolution of at least 4 transmural elements is needed. A geometric error of the measured apex locations was not present in any of the generated meshes, the meshing tool placed the apex always exactly.

Obtaining a reference for volumetric accuracy is more difficult. The accuracy used for the coupling condition of the circulatory system in Section 6.3 was 10^{-7} ml which is much smaller but necessary to prevent side effects (numerical oscillations on the time scale). A second, probably better volumetric reference could be derived from the MRI accuracy, which itself limits the possible accuracy of the segmentation. Assuming an uncertainty of 1 voxel on the boundary, i.e. two voxels per dimension (one at beginning and ending of a slice, each), this adds $1.44 \text{ mm}/1.8 \text{ mm}$ per dimension. Assuming further that a human ventricle is bounded approximately by a box sized $5 \text{ cm} \times 5 \text{ cm} \times 10 \text{ cm}$, this

5. Validation of the FEM Solver

gives an “inaccurate” volume of $5.144 \times 5.144 \times 10.36 \text{ cm}^3 = 274 \text{ ml}$ instead of 250 ml, an increase of 9.6% or 24 ml. This is more than five times the observed volume change even on the coarsest grid (4.25 ml on endocardium after conversion) and also any of the other simulations stayed below that threshold as well. Due to the definition of the “change” values, thereby for accurate results at least a mesh of resolution 2 should be used.

Surface distance is a special case as it is not directly related to a physiological meaning and more of relevance for the contact handling algorithm. As reference, Fritz et al. (2014) state that the surface distance stayed below 0.15 mm throughout a whole cardiac cycle. Surface distance shows a considerable effect of geometric discretization in relation to the numerical accuracy, given by the necessarily large triangle size compared to the curvature on the resolution 1 mesh ($0.011 \text{ mm} * 2.6 = 0.026 \text{ mm}$ on the endocardial surface). An explanation is that surface distance actually uses the curvature, which gets represented better, the more elements are used. This is an important observation for the choice of the contact handling’s penalty factor, as the surface distance in general does not go beyond the discretization error for a given mesh. That mesh of resolution 1 also reached only an accuracy increase similar to the reference when going from resolution 1 to 2 ($0.043 \text{ mm} * 2.6 = 0.11 \text{ mm}$ after inflated on the epicardium), which however still increased when comparing resolution 2 with resolution 4 mesh by $0.0135 \text{ mm} * 2.6 = 0.0351 \text{ mm}$ and resolution 4 with resolution 8 mesh ($0.045 \text{ mm} * 2.6 = 0.117 \text{ mm}$). In an MRI dataset none of these would be visible, however as even the change between resolution 2 and 4 is only 5 times smaller than the contact surface distance, a volumetric contact pericardium should be used with a resolution of at least 4 transmural elements.

As finite elements ensure accuracy only in the weak formulation, these results are no surprise. By solving a volume integral representation, the solver has no direct control over a single point’s accuracy. The authors expectations thereby were to require the finest mesh resolution in apex position (single-point evaluation), followed by the two-dimensional surface distances, and coarsest mesh for the enclosed cavity volume. Only the required high resolution for the surface distance opposes these expectations, however in that case a different reference was used.

Concludingly, the necessary mesh resolution strongly depends on the value interest, averaged values like the volume need less elements than the position of a single point. This is totally clear, since finite element method determines the solution of underlying equations from a weak formulation, that is only true in the sense of a volume integral average.

The real-life geometry used in this thesis had a typical resolution of at least 2 transmural elements in the left ventricle, which according to these results was found to be sufficient for volume evaluation, and at the lower limit for point evaluation as well as surface distance determination. This lets conclude that the results obtained from evaluation of e. g. valve displacement, circulatory system’s cavity volumes or pericardial contact problem do not contain significant errors larger than the errors present due to resolution of the underlying MRI data set.

5. Validation of the FEM Solver

Limitations One of the drawbacks of this study is the unavailability of a ground-truth solution, which is why a relative comparison was done. Simulations with the generated mesh at resolution 16 unfortunately did not provide a solution within 48 hours on 256 cores (hard limits of the used computing cluster). The finest resolution that yield a solution (resolution 8) did not yet show an extraordinarily good accuracy of the surface distance.

However, it has to be kept in mind that besides the computational aspect of better approximating the analytical solution, higher resolution in MRI-based data will contain more detail especially in irregular structures, both in better approximating the real shape like showing the papillary muscles, trabeculae or the valve planes as well as on the material assignment side needed for the addition of ablation scars modeling or irregular fiber orientation often found in pathological tissue. In the same manner, the inclusion of realistic fiber directions that change their orientation transmurally and the ability of an accurate representation of these depend a lot on the chosen resolution (and thereby even with interpolation needs at least representations at two transmurally different locations due to Nyquist-Shannon sampling theorem).

Future works As this work is based on the second benchmark case of Land et al. (2015), a simple and straight-forward extension of this work is the inclusion of active contraction as in the third benchmark case. Switching to anisotropic material parameters and adding active contraction impacts the process of mesh generation as it requires the definition of a fiber direction and at least two transmural points for an adequate representation. Due to incompressibility the tissue will shorten in that direction and elongate in the other two directions during contraction. The results will be different due to the introduced anisotropy, even if only passively inflating such a model. The expected additional benefits for the question of required resolution is modest, since contraction counteracts inflation and yields to less deformation, not more. Thereby the expected outcomes will be in a similar range or even less pronounced, but due to the variety of involved effects, more difficult to see.

As remedy for the required high resolution of the mesh and the associated computational effort to obtain an accurate solution for some of the values could be to change the underlying type of finite element. Two-dimensional elements called “shell elements” were specifically developed to solve similar problems in mechanical or construction engineering with reasonable computational effort (Belytschko et al., 2014, Chapter 9), and even exist in variants supporting multi-layered composite materials (Hauptmann et al., 2001) applicable to large deformation in car crash analysis. These could provide substantially better results especially in the region of the very thin-walled atria, that are known to comprise of at least two physiologically very different layers, and where meshes of good quality are difficult to create due to an almost vanishing thickness.

5.2. Scalability Study and Parallel Performance

5.2.1. Introduction

As heart simulations are the combination of a number of complex models, they demand a non-trivial amount of computational power. On a current Desktop computer from the year 2014 (Core i7-4771 4x3.5GHz, 32 GB RAM), the typical duration of a simulation of the model used by Fritz et al. (2014) takes up between twelve and 24 hours for one heart beat. Improving aspects of modeling e. g. by a different tension model (Section 6.2), an increased mesh resolution (Section 5.1), running multiple heart beats to obtain convergence (Section 6.3), or realizing inverse-type applications that need multiple forward simulations like parameter estimation of passive material (Section 6.1) or determination of contraction forces (Fritz, 2015) and different combinations of these (Coupling, Section 4.3) are all use cases that have the potential to dramatically increase the simulation time towards a non-practical level. Parallelization of the main parts of the code is a way to cope with this problem of increased computational demands, by using multiple cores for the same simulation (Section 4.4). Speedup is defined as the quotient between sequential runtime on a single core (time_1) and runtime on N cores (time_N) via $\text{speedup} = \text{time}_1 / \text{time}_N$. A bad scalability means that increasing the number of cores does not improve computation time noticeably, a good scalability however gives the potential to use hundreds of cores on a professional computing cluster like the uc1 at a fraction of sequential runtime.

5.2.2. Simulation setup

The first part of this project tests scalability of the implementation by means of the benchmark test case “Problem 3: inflation and active contraction of a ventricle” on an analytically describable geometry (Land et al., 2015). Meshes of different resolutions were generated using the scripting language of GMSH and Table 5.2 lists the number of nodes and elements for each of them. A second-order mesh was used that comprises of a through-the-wall resolution of 8 elements resulting in 456k unknowns, and provides at least 1.7k elements per core to impede negative effects due to a small mesh distributed on too many cores. Simulations on an even finer resolution mesh comprising 16 through-the-wall elements of second-order did not even provide a single time step within the computation time limit of 48h (imposed by uc1 on 256 cores), possibly due to ill-conditioning and a different preconditioning needed for such a mesh size.

Specifically, investigations were done to compare runtimes on

- a single node using 1, 2, 4, 8, 16 cores,
- 1, 2, 4, 8, 16 nodes with 1 core per node, and
- when using more than 16 cores.

Since not all simulations finished within the given time limit, runtimes were evaluated after 10 % of the maximum pressure was computed, which took between 10 min and 10 h.

5. Validation of the FEM Solver

Table 5.2.: Overview of the mesh sizes for the analytical left ventricle geometry depending on the number of elements through-the-wall. Number of elements is the same for linear (T4) and second-order tetrahedra (T10). Degrees of freedom can be computed by 3 times the number of nodes.

setting	#nodes (T4)	#nodes (T10)	#elements
vent1	222	1279	616
vent2	722	4233	2070
vent4	3717	23597	13275
vent8	21870	151921	96912
vent16	149467	1107874	764393

The second part of this project additionally aims at testing the scalability of the implementation in a more feature-rich setup, including all four heart chambers, length-dependent tension development and fully coupled circulatory system on a mesh of moderate resolutions (2 transmural elements), but without pericardium modeling. The simulations are done on the “LumensMitLAT” setup as described in Section 6.2 with the addition of a volumetric contact pericardium.

Two different computing systems were used for this study. The first one is the *institute’s cluster* located in the basement of the Institute of Biomedical Engineering. It consists of several mid and late 2012 Mac Pro workstations running macOS 10.12, each equipped with two six core processors (Intel Xeon E5645 6x2.4GHz Q1/2010) and 32 GB RAM (ibt140-142) or 64 GB RAM (ibt143-146), respectively. These are connected via Gigabit Ethernet, which basically limits its reasonable usage for this application to one node and 12 processes per simulation. The nodes are accessed directly by the institute’s students and employees using ‘ssh’ and simulations are run using ‘screen’, but exclusive use of a node is not guaranteed. The other one is the computing cluster *uc1* at the Karlsruhe Institute of Technology intended to provide a basic computing service to all members employed at one of the universities in Baden-Württemberg. It is consisting of 2 login nodes and 512 ‘thin’ computing nodes with 16 cores per node (two Intel Xeon E5-2670 8x2.6GHz Q1/2012, 64GB RAM) and InfiniBand 4xFDR as low-latency inter-node connection. The nodes are accessed indirectly by ssh to a login node and submitting one or several computing jobs to the queuing system. This allows to use up to 16 nodes with a total of 256 cores per simulation and guarantees exclusive usage of the nodes.

Simulations of the ventricle inflation were performed on the *uc1* to evaluate scalability up to 256 cores, the maximum of available computing cores. The only sequentially running data export was turned off in this case. Simulations of the fully-featured heart model were done on the *uc1* and the institute’s cluster with up to 16 cores, to allow comparison on two different computing systems. To prevent slow downs on the institute’s cluster due to simultaneous memory access from multiple simulations, only one simulation was allowed at once even if sufficient cores for a second simulation were free.

5.2.3. Results and evaluation

Scalability of the benchmark ventricle on uc1 Figure 5.4 shows the speedups obtained on uc1 for the ventricle inflation. As intended by the authors of the PETSc library used for solving the linear equation systems, the problem scales almost linearly with the number of processes. At a total number of 16 processes (left plot), the simulation reaches a speedup of 11.8 when distributed over 16 nodes, and a speedup of 8.1 running on the same node. It is remarkable that when looking at a fixed number of processes, simulations using multiple nodes (blue) were consistently faster than the ones running on the same node (green), although these require additional and potentially expensive communication in-between the nodes. The InfiniBand network used to connect the nodes on uc1 was specifically developed for HPC applications, it allows very low latencies for inter-node communication. However, computations with data never leaving the processor should still be faster than computations involving even high-performance communication. This might be an indication that the problem is neither bound by raw computational power nor communication. A possible explanation for this paradox is total available memory bandwidth as a bottleneck. On single-node computations this needs to be shared between multiple processes, but is fully available to the one process per node on the distributed computations and multiplies with the number of computing nodes. Thereby for a fixed number of processes, the distribution over several nodes can indeed be faster (more memory bandwidth per process), than on a single node (shared memory bandwidth).

Looking at the right plot in Figure 5.4, the number of processes per node is not fixed anymore but increases from 32 nodes onwards. The speedup graph however shows a clear drop in slope at 64 cores. Due to the limitation to a maximum of 16 nodes per simulation, the 64 processes are distributed on 16 nodes. One possible explanation is that at this point the number of unknowns drops below 10.000 unknowns/process and the communication overhead becomes prevalent. Using a mesh with a smaller degree of freedom, that point would probably be reached already at a lower number of processes. Another possible explanation might be the physical hardware of the nodes, which has 4 parallel memory channels, meaning that simulations with 8 and 16 processes per node are significantly more impacted by the memory bandwidth limitation. The obtained speedups of 26.8 (64 processes on 4 nodes) and 32.2 (64 processes on 16 nodes) still increased further on 128 and 256 cores, and finally achieved a value of 55.7 on 256 cores.

Scalability of a full 4ch model setup Table 5.3 shows computing times for the four-chamber model running on uc1 and on the institute’s cluster, all using the same “LumensWithLAT” setup. On uc1, the scalability is better and the speedup there is consistently higher than on the institute’s cluster peaking at 6.99 vs 2.60 for single node usage. On 8 cores for example, uc1 allows a speedup of 3.27, whereas the institute’s cluster shows only 2.55. The sequential runtime – which is taken as reference for the relative speedup – however is on uc1 much longer (27h 33min vs 11h 15min) and the runtime on uc1 is only faster when using at least 16 cores or more. Regarding the absolute runtime on 8 cores – the maximum physical 2^n cores available on both systems – results were obtained on the institute’s cluster in 4h 25min about twice as fast as on uc1 with

5. Validation of the FEM Solver

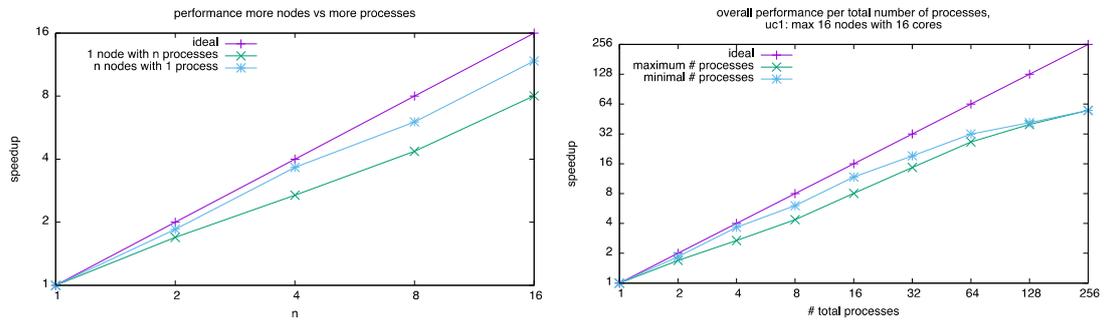


Figure 5.4.: Relative runtime compared to single-core (speedup) of the benchmark ventricle for “increasing cores” compared to “increasing number of nodes” with a fixed number of processes n . The right plot shows the speedup achieved on up to 256 cores where from 32 process onwards used either all physical cores of a node (green) or as little cores as possible (blue) under the limitation of 16 nodes at most.

Table 5.3.: Absolute runtime of the simulations with the scalability setting based on “LumensWithLAT” on the institute’s servers and on ucl. The theoretical minimal runtime denotes sequential runtime divided by number of MPI processes. Adapted from Müller (2017).

Number of MPI processes	Computed on	Theoretical minimal runtime	Real runtime	Real obtained speedup
1	IBT140	-	11:15:14h (40514s)	1.0000
2	IBT141	5:37:37h (20257s)	8:18:56h (29936s)	1.3533
4	IBT142	2:48:49h (10129s)	5:34:23h (20063s)	2.0193
8	IBT141	1:24:24h (5064s)	4:24:36h (15876s)	2.5519
16	IBT142	0:42:12h (2532s)	4:19:52h (15592s)	2.5984
1	UC1	-	27:32:53h (99173s)	1.0000
2	UC1	13:46:27h (49587s)	17:15:13h (62113s)	1.5967
4	UC1	6:53:13h (24793s)	9:59:04h (35944s)	2.7591
8	UC1	3:26:37h (12397s)	8:26:07h (30367s)	3.2658
16	UC1	1:43:18h (6198s)	3:56:37h (14197s)	6.9855
32	UC1	0:51:39h (3099s)	2:15:04h (8104s)	12.2375
64	UC1	0:25:50h (1550s)	1:37:43h (5863s)	16.9151

5. Validation of the FEM Solver

Table 5.4.: Runtime of the tension model simulations on the institute’s servers using 12 cores. Standard deviation is a measure of reproducibility of the results. The smaller, the more precise the runtime was reproduced. Cells in green point to the best values achieved with each setting. Adapted from Müller (2017).

	DoubleHillWithLAT	DoubleHillNoLAT	LumensWithLAT	LumensNoLAT
Computed on	IBT143	IBT144	IBT145	IBT146
Runtime 1st run	51721s (14:22:01h)	59788s (16:36:28h)	60984s (16:56:24h)	50527s (14:02:07h)
Runtime 2nd run	51766s (14:22:46h)	59802s (16:36:42h)	61097s (16:58:17h)	50488s (14:01:28h)
Runtime 3rd run	51875s (14:24:35h)	59816s (16:36:56h)	61076s (16:57:56h)	50581s (14:03:01h)
Runtime 4th run	51778s (14:22:58h)	59982s (16:39:42h)	61169s (16:59:29h)	50545s (14:02:25h)
Runtime 5th run	51761s (14:22:41h)	59894s (16:38:14h)	61046s (16:57:26h)	50606s (14:03:26h)
Runtime Average	51780,2s (14:23:00h)	59856,4s (16:37:36h)	61074,4s (16:57:54h)	50549,4s (14:02:29h)
Standard deviation	51,121s	72,745s	60,731s	41,215s

8h 26min. Using a full node, results were obtained in a similar time on both systems with 4h 20min on the institute’s cluster and 3h 57min on uc1.

On the institute’s cluster it is noticeable that there is only a small runtime improvement of 26.4% between 4 and 8 cores from 2.02 to 2.55. On uc1, the improvement is with 18.4% from 2.76 to 3.27 even lower, however from 8 to 16 processes it is more than twice as fast (113.89%) which is faster than actually possible, indicating an unusual inefficiency on 8 processes only. In case such an inefficient runtime on 8 cores would have been observed only on the institute’s cluster, it might have been perturbances due to non-exclusive access by otherwise running jobs with extensive memory accesses. Since this anomaly was visible on uc1 as well, where the nodes are used exclusively for this one simulation, another possible explanation could rather be e. g. a disadvantageous mesh distribution of the four-chamber mesh to 8 partitions, which vanishes when using more or less partitions.

The minimal runtime difference on ibt cluster between 8 and 16 cores (speedups of 2.55 and 2.60), exposes that – although macOS shows 24 cores – only 12 physical cores are available. The four additional cores are used through hyper-threading, actually impacting eight processes, four running on a physical core and four running on the virtual core. Hyper-threading was only used for this 16 core simulation on the institute’s cluster. CPUs with hyper-threading have a complete second set of registers for each core and can switch between these contexts whenever the execution pipeline has to wait for new data e. g. from a small CPU cache or lower clocked memory, thereby increasing the effective usage of the cores for some types of code. The inefficient use of hyper-threading is an indication that the `mumps` library used to solve the linear equation systems is already highly optimized.

For absolute time until the results are delivered, an additional waiting time due to queuing system needs to be considered as well. The more nodes and cores are requested the longer the waiting time usually becomes. A huge advantage of uc1 however is that up

5. Validation of the FEM Solver

to 50 jobs can be submitted to the scheduler’s queue and are all executed simultaneously whenever the resources are available. Furthermore, each job can be assigned to up to 16 nodes (256 cores in total) without any limitation of simultaneously executed jobs.

The slow speedup on the institutes cluster could be caused by either an inefficient parallelization, but also by an unsuited reference point for the single core runtime as well. Further, the huge difference between single core runtime on uc1 and single institute’s cluster can not be explained only by the different clock speed of 2.6 GHz instead of 2.4 GHz. Modern processors of that era provide features intended to increase the perceived speed for desktop usage including hyper-threading and turbo boost, which dynamically overclocks the core frequency when not all cores are in use. On uc1, the turbo boost feature is intentionally disabled in favor of an increased power efficiency, however the state of turbo boost on the institute’s cluster could not be clarified. Regarding turbo boost, here it would also have been interesting to see if e.g. 12 sequential simulations running simultaneously on the same node would show the same runtime as a sequential simulation on an otherwise empty node. A look at the processors’ data sheets also revealed a different number of memory channels between the CPUs, which is 4 channels CPU on uc1, but only 3 channels per CPU on the institute’s cluster, giving another possible explanation for a low maximum speedup on the latter.

Runtime reproducibility on the institute’s cluster Investigating the reproducibility of computing times on the institute’s cluster, Table 5.4 shows the runtimes using all 12 cores for four different tension models. Each Simulation was conducted five times and evaluated for average and standard deviation to reveal negative impacts due to simultaneous simulations from other users on the same node or backups running in the background. The fastest simulations took between 14 (“LumensNoLAT”) and 17 hours (“LumensWithLAT”). The standard deviation was at most 72.7 seconds for the setup “DoubleHillNoLAT”, giving good confidence that the results are valid for evaluation and there were no other processes running that drastically increase simulation time. For the DoubleHill tension model, using locally inhomogeneous activation was expectedly faster (14h 23min vs 16h 38min on average) as it leads to temporal distribution of peak tension, less deformation and easier-to-solve equations. Similarly for the simulations with locally inhomogeneous activation, the more complex Lumens tension model took longer than the precomputed DoubleHill tension model (14h 24min vs 16h 58min on average). Interestingly, this was not observed for the “LumensNoLAT” setup, which showed the fastest runtime of all conducted simulations and was even faster than the “DoubleHillNoLAT” setup with a less complex tension model (14h 2min vs 16h 38min) and faster than the “LumensWithLAT” which showed less deformation (16h 58min).

During evaluation of this study, one of the institute cluster’s nodes (ibt146) provided results significantly faster than the other ones. A closer look at the hardware, revealed that this specific node was unknowingly equipped with a different CPU configuration (two Xeon X5675 6x3.06GHz instead of E5645 6x2.4GHz), making the runtime results obtained on that node difficult to evaluate for this study (Table 5.4). It also turned out that one of the other nodes (ibt143) had only 48GB RAM instead of 64GB. CardioMechanics

5. Validation of the FEM Solver

however never used more than 10GB RAM in any of the simulations in this study, thereby this should not have had any impact on the results. Further reasons could be different implementations of the basic linear algebra subroutines (BLAS), which were used from Intel’s math kernel library (`mk1`) on `uc1` and Apple’s `vecLib` on the institute’s cluster. Especially `vecLib` might or might not use an implicit multithreading, even when explicitly run on a single `mpi` process and using the environment variable intended to control that behavior (`VECLIB_MAXIMUM_THREADS=1`). An other oddity only observed on the institute’s cluster was that, at least in case of sequential simulations, one CPU core was permanently occupied completely by the `mpi` master thread, effectively downgrading a dual-core CPU to single-core performance.

The systematic investigation of the parallel behavior also revealed some reproducibility issues, eventually causing random simulation failures of long-running simulations with several cores (64-256). Small deviations of the simulation results however already existed on two cores, where rounding errors due to floating-point arithmetics and a nondeterministic order of mathematical operations in parallel should make no difference. The problem could finally be tracked down to the external linear solver library `mumps` intentionally not producing predictive results in favor of a faster runtime, and several bugs in the step length algorithm. The remedy was to switch to the `superlu_dist` as linear solver instead (two times slower, but deterministic) and do a complete rework of the adaptive step length algorithm as described previously in Section 4.2.

Summary Two different simulations were performed on two different computing systems. On the computing cluster `uc1`, the scientific ventricle inflation delivered a speedup of up 55.7 on 256 cores distributed over 16 nodes and the 4ch simulation a speedup of 6.99 on 16 cores using 1 node. The same 4ch simulation using 8 cores was about twice as fast on the institute’s cluster compared to `uc1`, however scalability on `uc1` was better and using a full node, both systems delivered results in a similar time. Running the same setup, `uc1` was faster in absolute runtime when using 16 cores onwards, and it also has more capacity of up to 50 computing jobs. Weird behaviors on the institute’s cluster regarding scalability and absolute runtime could partly be tracked down to undocumented CPU changes on one node and RAM changes on another node, a difficult-to-control implicit multithreading of Apple’s blas implementation `vecLib` and a strange CPU usage behavior at least of sequentially running `mpi` processes, making scalability results obtained on that system difficult to evaluate.

Future works For similar studies, it should be kept in mind that the number of unknowns per core is an important factor and should not be chosen too low. An alternative to the here used strong scaling with a fixed problem size is to test weak scaling, that is to scale the problem size with the number of cores and only fix the problem size per processor. That way the relation between communication and computation cost should be similar over a large range of computing cores. The second point that should be investigated further is to evaluate the behavior of a fully featured simulation on a higher resolution mesh, including a larger number of cores. This however involves the laborious task of

5. Validation of the FEM Solver

recreating a higher resolution mesh as depicted in Section 4.5. Since mesh refinements of 3d tetrahedral meshes is a non-trivial task, this means starting mostly from scratch at the segmentation, reassigning material numbers, creating fiber orientation, cavity surfaces and pericardial layers, and last but not least iterating over a large number of simulations with mesh variations of problematic areas until mesh is found working for a full heart beat. Results from the previous section (Section 5.1) gave hints about necessary mesh resolutions for different applications on a scientific geometry which could serve as a well-founded guide for the mesh creation of such a human geometry.

6. Improving the Physiological Behavior

This chapter presents the improvements applied to the heart model of Fritz et al. (2014) in order to obtain a more physiological behavior.

In the first application, a more valid passive behavior of the left ventricle is obtained by inflation under absence of any contraction forces and a procedure for automatic determination of passive model parameters. Besides passive material parameters, that behavior is also closely linked to the pressure-free state, which needs to be determined as well.

The second improvement is a better active behavior by implementing an ode-based tension model that respects the current length of myocytes and allows to obtain a plausible local stretch in each cell.

The last two applications cover the computation of external forces acting on the myocardial surfaces. On the blood side, it comprises a new model for the circulatory system based on an equivalent electric circuit, that ensures blood conservation by connecting all four chambers in the correct order. That model additionally allows to simulate multiple heartbeats and hence to obtain a periodic quasi-steady-state. Defining individual opening and closing events depending on the pressure difference plays an important role to reach that goal.

The last application covers external forces on the outer heart surface with an alternative approach to model the effects of the pericardium and its surrounding tissue. The previous approach depends on a highly complex volumetric mesh and mesh-generation procedure with parameters that were not based on measurements. Homogenization of the pressures and replacing the model by a zero-dimensional pericardium with one value for the whole epicardial surface allows to choose model parameters related to measurements while the pericardial mesh generation process simplifies to just defining the epicardial surface.

6.1. Application I – Estimation of Passive Material Parameters using Mathematical Optimization

For the simulation of a realistic behavior of the deformation of a human heart, different components play together. Amongst active and passive behavior of the myocardium, surface forces act on the endocardial surface due to blood pressure and forces on the epicardial surface due to the surrounding tissue. During the relaxation phase of the ventricles however, the re-inflation of the ventricles is dominated by inflowing blood from atria and pulmonary veins, restricted by passive behavior of the ventricles.

Passive behavior is modeled in this framework using the Guccione constitutive law, which is a reduced version of the Costa law. The latter was found by (Schmid et al.,

6. Improving the Physiological Behavior

2008, 2009) to have the most identifiable parameter set amongst the evaluated material laws. In literature, very different parameters for reproducing passive behavior of the left ventricle can be found (see Table 3.2), obtained from various types of tissue at in-vivo as well as ex-vivo state.

Unfortunately, these parameters showed all a different pV behavior of the left ventricle when applied to our patient's geometry (Figure 6.3). While it is desirable to have parameter sets able to describe multiple patients, experience showed that the passive behavior in terms of absolute pressure volume response differs from patient to patient and – even worse – differs between two segmentations of the very same patient, probably due to a slightly different segmented wall thickness. In this project, a method is presented to reproduce the correct global passive behavior of one specific patient, by finding suitable ventricle parameters for the Guccione material law.

As data available for comparison, Klotz et al. (2006) measured the in-vivo pressure-volume response of the left ventricles at eighty freshly harvested human hearts ex vivo, and validated their results by 36 in vivo patients. Their major conclusion is, that after normalization to a standardized volume V_{norm} , the response is independent from sex, age or health state of the patients, and an exponential function can be fitted. Their resulting plot is shown in Figure 6.1 and can be described by an exponential relationship

$$p(V_{norm}) = \alpha \cdot (V_{norm})^\beta$$

with $V_{norm} = \frac{V-V_0}{V_{30}-V_0}$, $\alpha = 27.78$ and $\beta = 2.76$. V_0 and V_{30} are the volumes measured at 0 mmHg and 30 mmHg.

Additionally, Klotz et al. (2006, 2007) give a formula for computing the pressure-free volume V_0 from a given loaded volume V_m and corresponding loading pressure P_m :

$$V_0 = V_m(0.6 - 0.006P_m)$$

Assuming an end-diastolic pressure of 8 mmHg and the corresponding volume V_8 , this provides an expected volume ratio between pressure-free volume and end-diastolic volume of $V_0/V_8 = 0.55$.

In the model used by Fritz et al. (2014); Fritz (2015), the pV relationship is very different. Figure 6.2 shows the pressure-volume relation of the left ventricle from that model, which is almost linear. Moreover, the ratio between pressure-free volume and volume under load computes to $V_0/V_8 = 0.92$ which is significantly larger than the expected ratio of 0.55 and the simulation struggles to reach a sufficient volume reduction during systole. They used in the ventricles Guccione material with the parameter set by Omens et al. (1993). In summary, the parameter set by Omens et al. (1993) produces a vastly too stiff behavior in that model.

Realizing that the parameters by Omens et al. (1993) behave too stiff, a more appropriate parameter set could resolve a number of following problems observed with the model by Fritz et al. (2014):

1. passive behavior, volume V_8 increased over V_0 by only 8%,

6. Improving the Physiological Behavior

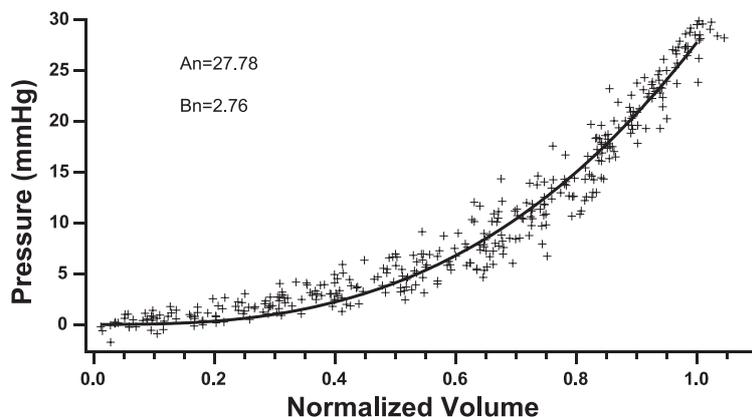


Figure 6.1.: Pressure-volume response of eighty human hearts with normalized volume $V_{norm} = (V - V_0)/(V_{30} - V_0)$, from Klotz et al. (2006).

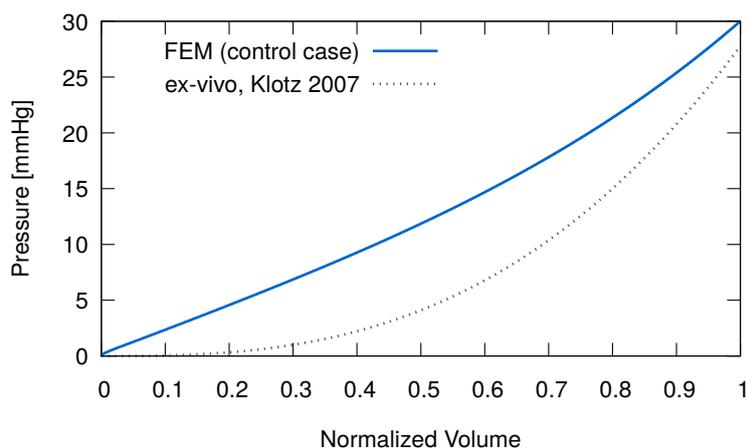


Figure 6.2.: Previous passive behavior of the left ventricle as present in the model by Fritz et al. (2014), without pericardium. Pressure-volume relation was obtained by inflating the left ventricle according to Klotz et al. (2006). The pressure was increased linearly over a simulated time of 5 s, and thereby includes mass inertia. The computed volumes are: pressure-free $V_{p=0 \text{ mmHg}} = 132.6 \text{ ml}$, end-diastolic (segmented) $V_{p \approx 8 \text{ mmHg}} = 143.5 \text{ ml}$ and inflated $V_{p=30 \text{ mmHg}} = 214.3 \text{ ml}$.

6. Improving the Physiological Behavior

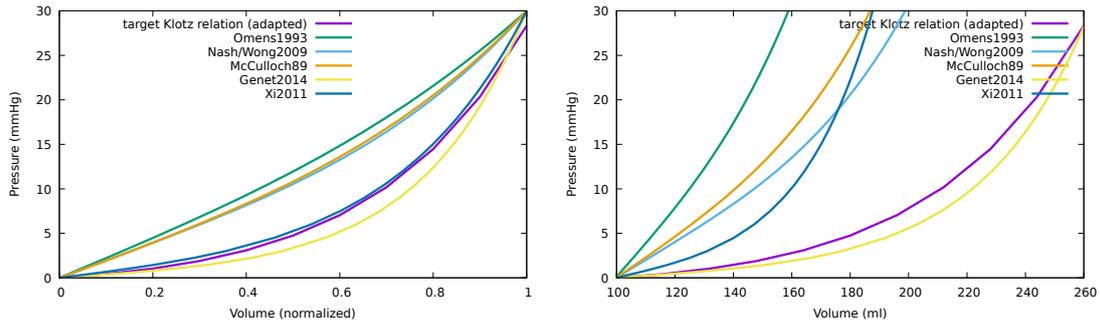


Figure 6.3.: Comparison of different literature parameters for the Guccione model in the 4ch geometry with (left) and without normalization of the volume (right).

2. end-systolic volume, ejection fraction of only 30 % was too small,
3. strange pressure and volume time courses during ventricular relaxation,
4. too large fiber strain, went down to 50 % although physiologically only $> 84\%$ possible, ventricular volume reduction however is still not sufficient,
5. too small end-diastolic fiber strain, which should be $\geq 1.1 = \text{end-diastolic sarcomere length} (2.1 \text{ micro meter}) / \text{unloaded sarcomere length} (1.9 \text{ micro meter})$, and consequently develops a too small tension with a stretch dependent tension model

Literature Overview

In literature, numerous parameters for ventricular myocardium were presented, however with very different behavior. Table 3.2 on page 32 list some of the parameters and Figure 6.3 shows the corresponding pV response of the left ventricle. Two distinct behaviors are already visible with normalized volume, however comparing absolute volumes the situation is worse. None of the published parameters was able to produce a satisfyingly correct volume increase and passive pV relationship. Reasons are that a certain set of parameters is only assured to work well in the original context including modeling features and the specific geometry of the corresponding publication.

A number of material parameter studies for the Guccione law were presented in literature:

1. Omens et al. (1993): use measurements of dog
2. Xi et al. (2011): parameter estimation for left ventricle using an unscented Kalman filter
3. Genet et al. (2014): parameter estimation for human left ventricle using a derivative-free quadratic bound optimization.

An elaborate parameter estimation method for Guccione parameters using a finite element model of the left ventricle was presented by Genet et al. (2014). They applied their

6. Improving the Physiological Behavior

estimation method in a small study of five normal human volunteers, including validation against measured strains, resulting in the estimated parameter set $C = 57.5 \pm 4$ Pa and $B_0 = 14.4 \pm 3.18$ (originally $C = 115 \pm 8$ Pa in their formulation of the Guccione law). While the variance of their estimation seems to create a reliably reproducible parameter set suitable for a number of patients, these parameters were still found to be inappropriate when applied to our patient's model. This specific parameter set behaved too weak, possible reasons could be numerous differences in the forward model, a different wall thickness in the left ventricular geometry, the absence of the right ventricle and both atria, their fixation of the valve plane and a different mesh resolution (3500 hexahedral elements / 5000 nodes in the model used by Genet, compared to 7300 2nd order tetrahedral elements / 13600 nodes alone in the left ventricle). Additionally, in the relevant pressure range during relaxation (between 0 and 8 mmHg), their parameter set fitted the least. This creates the conclusion that parameter estimation works more or less independent from the specific patient's geometry, however the material parameters resulting from such an inverse method are highly dependent on the forward model.

At the current state, the most promising approach to obtain a correct left ventricular pV response is to use a parameter estimate procedure and compute parameters suitable for the specific model and geometry.

6.1.1. Methods

Aim of this application is to obtain parameters for the passive material using an optimization procedure. The final solution should yield a certain prescribed pV relationship under passive inflation of the left ventricle.

An optimization consists of two components. First, a *forward model* converts a set of input parameters to an objective value, which describes the quality of the result. In this case, the forward simulation is used to compute the pressure volume relationship for a given set of material parameters. To be useful in later applications, it is important that the forward model resembles the final production simulation as close as possible. Second, an *optimization procedure* modifies the input parameters of the forward model in order to decrease the objective value accordingly.

Geometric Modeling Two different geometries were used in this study for the forward simulations:

- 1ch: An ellipsoidal geometry described by analytical functions, that resembles the geometry of a left ventricle. This is the very same geometry as used in the benchmark publication by Land et al. (2015) for the second and third test cases for testing the correctness of a simulation code under ventricle inflation. The main advantage is that a mesh of good quality and arbitrary resolution can be generated. A resolution of two second-order elements transmurally was used, which is a small but still predictably behaving mesh allowing fast forward simulations beneficial during algorithm development. This geometry was used to test reproducibility of the result under different start values.

6. Improving the Physiological Behavior

Table 6.1.: Resolutions of the two meshes that were used for the parameter estimation.

geometry	# nodes	# elements	included parts
1ch	4235	2072	LV
4ch	13592 (LV), 39067 (total)	7341 (LV), 21780 (total)	LV, RV, LA, RA, valve-plane, truncated veins

- 4ch: A segmentation-based geometry that includes all four chambers, the valve plane and truncated representations of the veins, but here without the surrounding pericardial mesh. The goal is to measure the isolated passive behavior by resembling the experimental ex-vivo setup from Klotz et al. (2006, 2007), where surrounding tissue and pericardium effects were not present. The obtained parameters however are usable for full-featured simulations including the pericardium.

The corresponding unloaded states were created as follows:

- 1ch: It is assumed that the described geometry already represents the pressure-free geometry and can directly be inflated as in the second benchmark case from Land et al. (2015).
- 4ch: An unloaded geometry was generated using one Bols iteration with an inflation pressure of 8 mmHg. The parameter C from the parameter set by Omens et al. (1993) was manually decreased to reach a volume reduction of approximately 50 %.

Details about the two meshes are given in Table 6.1. Although both meshes were created with a through-the-wall resolution of two elements, the surface triangles of the 4ch mesh are smaller and thereby even the left ventricle of the 4ch mesh has a larger number of nodes.

Determination of an unloaded geometry for a four-chamber heart is a non-trivial task itself and depends besides the initially unknown passive parameters on fiber orientation and other factors as well. In order to get at least an approximation that can be used for inflation, a geometry of the pressure-free state was generated using a single Bols iteration and modifying the pre-factor C . It is assumed to be sufficiently accurate due to a volume of roughly 50 % of the segmented volume. A similar procedure was described by Nikou et al. (2016), who iteratively applied parameter estimation followed by an unloaded state estimation, and accepted the procedure if the unloaded volume does not change by more than 5%.

Setup of the Forward Simulations The forward simulation uses the Guccione law for describing the passive behavior of the left ventricle, with its four model parameters C , b_1 , b_2 , b_3 to be determined by the optimization algorithm. The strain energy function of

6. Improving the Physiological Behavior

the Guccione material law is given by

$$W = \frac{C}{2}(e^Q - 1),$$

$$Q = b_1 E_{ff}^2 + b_2(E_{ss}^2 + E_{nn}^2 + 2 E_{sn}^2) + b_3(2 E_{fs}^2 + 2 E_{fn}^2).$$

In case of the whole heart model 4ch, the atria use the Mooney-Rivlin law, and the right ventricle uses the same material and parameters as for the left ventricle, which are thereby part of the optimization. The volume conservation penalty parameter of 1E6 Pa in the 1ch setup was reduced to 2E5 Pa in the 4ch to facilitate computation of the inflation.

Forces on the endocardial surface are given by a pressure boundary in the left ventricle, with a surface pressure increasing from 0 mmHg to 30 mmHg in at least ten steps (3 mmHg increments) applied to the unloaded geometries to create a step-wise inflation. Pressure-volume pairs are exported for each of the computed pressure steps. In case of the whole heart model, pressure is only applied to the left ventricle, to stay in accordance with the experimental setup of Klotz et al. (2006).

In all setups, neither active forces nor epicardial forces were present.

In the 1ch setup, the base plane is fixated, for the 4ch model the ends of the truncated veins. The apex is allowed to move freely as a fixation here would drastically impede the volume increase during inflation above the end-diastolic volume (> 8 mmHg).

The static solver was used in both setups for solving the equilibrium of forces without damping or mass inertia and prevents negative impacts from inflation duration or a changing step size. Although Klotz et al. (2006) did not describe precisely the time duration over which they inflated the left ventricle, their rather small volume increments of 0.025 ml might be an indication that all measurements were done in an equilibrium state.

Simulations of the 1ch setup were conducted using 8 cores and took between 15 s and 30 s for one inflation. Simulations of the 4ch setup took between 30 min and 60 min per inflation. Weaker parameters with a larger volume increase took longer in both setups.

The effect of mass inertia would be most prevalent at the beginning of an inflation. Passive forces are small and inertial forces can have a significant impact on the pV relationship. Starting in an equilibrium state, the computed volumes for a certain pressure are smaller than without inertia. Unfortunately this is exactly the range of pressures during the diastolic phase between 0 mmHg and 8 mmHg, the phase of the cardiac cycle where passive forces are dominant. Using mass inertia while determining passive parameters, the parameter estimation would try to counteract such a behavior by using significantly weaker behaving parameters than actually necessary. Mass inertia was also found to increase the dependency from the chosen time step size as well, since the solution depends not only on the current forces, but also on the accumulated inaccuracies of previous time steps.

The forward simulation yields for each parameter set at least ten computed volumes $V_{sim}(p)$, corresponding to the equidistantly distributed pressures in mmHg at $p \in \{3, 6, \dots, 30\}$. If a pressure value is required for a specific volume that was not directly computed, the two neighboring pressure values are linearly interpolated. If the pressure

6. Improving the Physiological Behavior

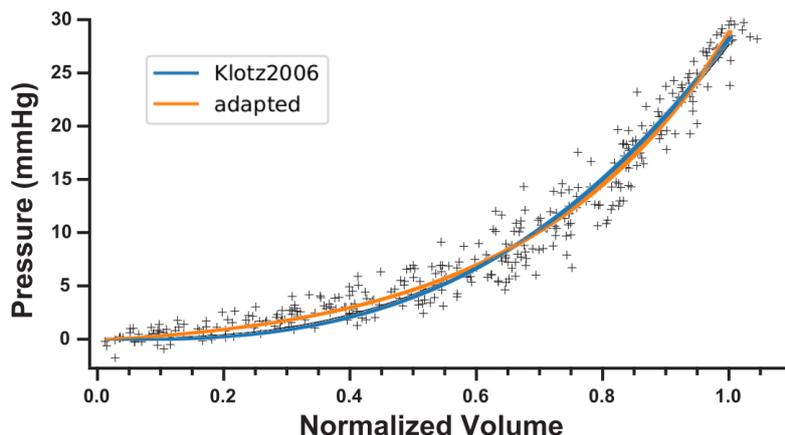


Figure 6.4.: Comparison between original pressure-volume relationship and an extended formulation adapted to produce a better fit for lower pressures.

increments reduced, e. g. due to a too large deformation, the intermediate steps are available for interpolation as well. Since the simulation basically computes volumes for certain pressures $V_{sim}(p)$ with given fixed pressure increments, a continuous function $p_{sim}(V)$ was approximated by linear interpolation from neighboring volumes.

Target relationships for pressure-volume (pV) pairs As choice for the objective function's target pV relationship, different formula were investigated. The Klotz relationship was designed to compare the pV response of human hearts with different volumes using a normalized relationship

$$p(V_{norm}) = A \cdot (V_{norm})^B, \quad \text{with } V_{norm} = \frac{V - V_0}{V_{30} - V_0}$$

with parameters $A = 27.78$ and $B = 2.76$. That curve describes globally the non-linear relationship of pressures and volumes under passive inflation of the left ventricle in an explanted four-chamber heart.

The original Klotz curve in Figure 6.1 however does not fit the data very well for pressures below 5 mmHg, or normalized volumes below 0.6. Almost all data points located in this region are above the curve. Unfortunately, this is the range of pressures applied when passive behavior matters the most, during the relaxation phase when all other forces are absent. In order to better treat that problematic region, an extended formulation was generated by extracting the data points from the plot and fitting an exponential relationship with an additional linear term cV_{norm} that better treats the lower part:

$$p(V_{norm}) = a(e^{b \cdot V_{norm}} - 1) - cV_{norm} \quad \text{with } V_{norm} = (V - V_0)/(V_{30} - V_0)$$

The obtained parameters are $a = 1.43887$, $b = 3.05987$ and $c = 0.91460$. Both curves are shown in Figure 6.4.

6. Improving the Physiological Behavior

These normalized relationships do not include information regarding absolute volumes or ratios like V_8/V_0 or V_{30}/V_0 . Two hearts of different stiffness that will obviously inflate to different volumes can still produce the same non-linear behavior. Applied as target behavior of an optimization problem, this translates to non-uniqueness of the solution. The missing information is to prescribe a certain volume relationship between two pressures and use absolute volumes for the target behavior of the pV response. For de-normalization, the unknown but necessary value V_{30} needs to be provided. A factor k was introduced, that scales the volume $V_{30} = k \cdot V_0$ depending on the known volume of the pressure-free state V_0 . The de-normalized pV relationship with prescribed volume scaling factor k using absolute volumes is given by:

$$p(V) = a(e^{bV_{norm}} - 1) - cV_{norm} \quad \text{with } V_{norm} = (V - V_0)/(kV_0 - V_0)$$

At $k = 2.57$ the pressure-free volume doubles under inflation with a pressure of 8 mmHg, the same assumption was made when generating the pressure-free state.

Optimization procedure In order to estimate parameters for the Guccione material law, a non-linear minimization problem of the form

$$\min_{\vec{x}} f(\vec{x})$$

is solved. The optimization parameters \vec{x} are given by the model parameters of the Guccione material law and the objective function $f(\cdot)$ describe the difference between a simulation outcome and a prescribed target behavior. The success of such an optimization approach vastly depends on the choice of a good and suitable objective function and the right initial values. An objective function not known to be convex might get stuck in a local minimum, similarly without the known existence of a unique minimum it is like to get a solution that vastly floats around without really reducing the objective value any further. Goal is to resemble an absolute pV behavior of the left ventricle, e. g. by measuring the difference between the two pV curves generated by forward simulation and a target pV relationship.

Input parameters are the four model parameters of the Guccione material $[C, b_1, b_2, b_3]$, the same parameter set gets applied to both ventricles. When using all four parameters as free variables, it can be observed that either b_2 or b_3 converges to zero. Such a behavior certainly does not exist in reality as it would describe a tissue without any shear forces. The issue can be overcome by introducing a parameter B similar to Genet et al. (2014) and restricting the non-linear constants to $b_1 = B$, $b_2 = 0.4 \cdot B$ and $b_3 = 0.7 B$. A restriction function $g(\cdot)$ is used to reformulate the minimization problem according to the constraints. The minimization problem with restriction function g then writes as

$$\min_{[C, B]} (f \circ g)([C, B]), \quad \text{with } g([C, B]) := [C, B, 0.4 B, 0.7 B]$$

converting between optimization parameters $\vec{y} = [C, B]$ and Guccione model parameters $\vec{x} = [C, b_1, b_2, b_3]$.

6. Improving the Physiological Behavior

As data available for the objective function, a number of volume values $\vec{V}_{sim}(\vec{x}) = \{V_3^s, V_6^s, \dots, V_{30}^s\}$ gets computed by simulation of the ventricle inflation with parameter set \vec{x} , and a number of pressures values is calculated from the target relationship $\vec{V}_{target} = \{V_3^t, V_6^t, \dots, V_{30}^t\}$, both at the same equidistant pressure steps. The objective function with equidistant pressure steps is given by

$$f(\vec{x}) := \left\| \vec{V}_{sim}(\vec{x}) - \vec{V}_{target} \right\|_2^2 = \sum_{i=1}^{10} |V_{sim}(p_i) - V_{target}(p_i)|^2.$$

As a simulation provides volumes at specific pressure steps, a simple straight forward approach for computing the objective function would be:

1. Compute volumes V_3, V_6, V_9, \dots for certain pressures (forward simulation) and normalize these to V_{30} to get the Klotz relationship of the simulation, and
2. calculate the difference to the target Klotz relationship directly at these pressures.

This approach however creates a bad fit and struggles to approximate especially the lower part of the volume range. A number of problems are inherent to such an approach. First, when using the unmodified Klotz relationship, there are no assumptions concerning the absolute volume increase, which makes it difficult for the optimization to find a solution (uniqueness is missing). Moreover, the original Klotz formulation itself has problems with approximating the original data in the lower part (inadequate original Klotz formulation) and hence will struggle to approximate a simulation of that data. Second, the optimization prefers a better fit at larger pressures over lower ones, and the volume for a certain pressure can be easily off by a factor of two without much impact on the objective. The reason is that there are more pV pairs with higher volumes than with lower ones as the volume increase is higher at lower pressures than it is for larger pressures. Unfortunately the passive behavior at low pressures is very relevant to fit, especially for the relaxation phase. The first two points can be resolved by switching to an extended formulation that better represent these data, and prescribe the volume increase for an absolute target relationship.

To circumvent the problem of a bad optimization fit in the lower pressure range, the solution is to use different evaluation points than the ones obtained from forward simulation, and compare pressures instead of volumes. With equidistant volumes, determined from V_{30} in the target relationship, the lower pressure region gets represented better. Additionally, by evaluating pressure instead of volume differences, the objective function becomes more sensitive to deviations especially right at the beginning of the inflation.

The corresponding objective function at equidistant volume steps then computes for a given set of volumes $\vec{V}_{sim}(\vec{x}) = \{V_3^s, V_6^s, \dots, V_{30}^s\}$ at equidistant pressure steps as

$$f(\vec{x}) = \left\| \vec{p}_{sim}(\vec{x}) - \vec{p}_{target} \right\|_2^2 = \sum_{k=1}^{10} |p_{sim}(V_k) - p_{target}(V_k)|^2.$$

The objective thereby measures the distance between target pressure for a certain volume and the pressure that would be necessary in the simulation to obtain that volume.

6. Improving the Physiological Behavior

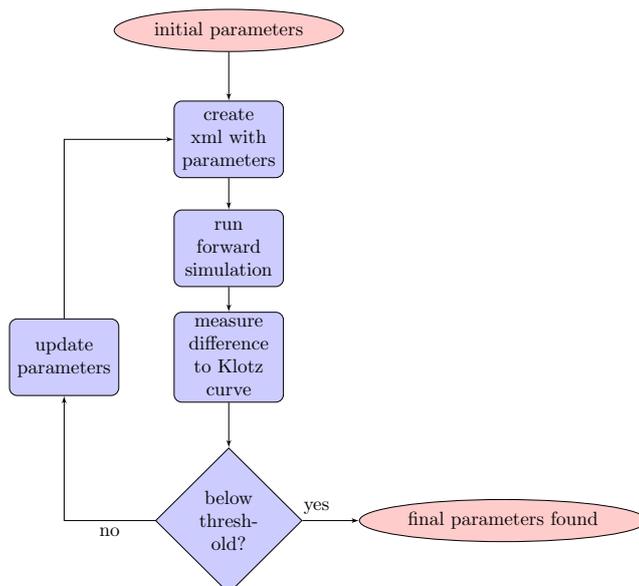


Figure 6.5.: Schematic view of the optimization process.

Since simulation yields only volumes at equidistant pressure steps, linear interpolation was applied to obtain the continuous pressure function $p_{sim}(\cdot)$ and pressures $p_{sim}(V_k)$ evaluated at equidistant volume steps $V_k = \frac{k}{10}V_0$, $k = 1, \dots, 10$. The volume before inflation V_0 is the same in all simulations and represented exactly, therefore V_0 does not need to be considered.

Task of the optimization procedure is to decrease the objective value by modifying the input parameter set, run a forward simulation and measure the simulation results by means of the objective function. The general optimization procedure is depicted in Figure 6.5 and is realized in a python script iteratively performing the following steps:

1. Create a xml configuration file containing the current parameter set,
2. trigger an external simulation run using the created configuration,
3. digest simulation outcome by extracting relevant data from the logfile and vtk files,
4. evaluate the objective function, and
5. determine a new parameter set using one of the optimization algorithms provided by the `scipy.optimize` package.

The derivative-free Nelder-Mead simplex algorithm was used for finding the minimum of that non-linear least squares problem. Tests with Newton method in combination with a finite difference approximation of the derivative were observed to be very dependent on the start value and often got stuck in local minima. Moreover, the finite difference approximation used more function evaluations by forward simulations in each optimization

6. Improving the Physiological Behavior

Table 6.2.: Initial values of all optimization runs. Start values were chosen to produce an initial ratio V_{30}/V_0 around 2.57. The start value of the 4ch setup are close to the solution from the 1ch optimization.

geometry	C_{initial}	B_{initial}	V_{30}/V_0
1ch	50.	14.	2.55
1ch	1000.	5.	2.60
1ch	5000.	2.	2.60
4ch	150.	18.	2.14

step. Rios and Sahinidis (2013) compared 22 implementations of different derivative-free optimization algorithms on a test bed of 502 convex, non-convex, smooth and non-smooth problems from engineering and science. Unless an explicit formulation of the derivative is known, it is unlikely that Newton with finite difference approximation works better than using a derivative-free method like Nelder-Mead directly, which could be observed during development as well. Extensions of Nelder-Mead however exist, that additionally use derivative information for the next guess to escape from local minima with non-zero derivative.

A treatment of failing time steps and simulations is necessary. If the algorithm is not able to handle a failing forward simulation, V_{30} is not available for normalization. Replacing the objective value by a large enough constant loses continuity and differentiability and simply does not work. This was overcome by using absolute volumes as optimization target instead of relative ones, and replacing the unknown pressures/volumes with the largest computed ones, which creates a continuous objective function. Reliability of the forward simulations also increased by a rewritten step length algorithm which furthermore had the benefit that it hits the initially intended pressure steps exactly even after reducing and increasing the step length.

6.1.2. Setup of the Optimization

Two different types of optimization setups were created, both using Nelder-Mead as optimization algorithm. For testing general robustness of the optimization procedure and reproducibility of the results, three different optimizations were run using the analytical left ventricle geometry “1ch” with start values as given in Table 6.2. In a second optimization setup, patient-specific parameters were determined for a real heart geometry using the 4ch heart. Initial values $[C, B] = [150, 18]$ were chosen close to the results of an intermediate version of the first optimization setups, manually modified to inflate to about twice the initial volume. As a single inflation is expected to take much longer here, fewer optimization steps are possible and it is beneficial to let the optimization already start in the estimated target range. In both setups, each forward simulation was conducted in parallel on eight cores.

6.1.3. Implementational Aspects

It has to be kept in mind that one “evaluation of the objective function” for a given set of parameters already corresponds to running a full forward simulation. An optimization typically runs a large number of forward simulations (100 simulations and above) with very different passive material parameters and hence a wide range of different deformations. Special requirements in terms of robustness and computational time are thereby needed for the code of the forward simulation. First, it needs to be fast enough to compute a sufficient number of optimization steps (i.e. forward simulations) in a reasonable time. Second, the code needs to be reliable and robust enough to return a solution for any set of input parameters. Both could be fulfilled by a parallelization of the essential parts of the code as described in Section 4.4 and the implementation of a robust algorithm for step size computation as described in Section 4.2. Parallelization was tested to provide correct results also in the parallel case and used eight cores for the optimization of all setups. Third, for an accurate approximation of the derivative, the code needs to be able to produce the same results even when a different number of intermediate steps was used. Using the static solver circumvents any dependency from the exact step size. A smaller step size leads to an initial guess of node displacements that is closer to the final solution (constant velocity assumption), thereby improves convergence of the forward simulation’s Newton method. The solution however does not change. Especially weaker material properties turned out to demand more time steps and smaller pressure increments, resulting in a longer computation time.

External optimization loop The optimization loop was implemented in a separate application outside CardioMechanics and realized using the python scripting language. Python libraries were excessively used for reading and writing xml configuration files (`xml.etree`), solving optimization problems (`scipy.optimize`) and reading csv data files (`numpy.genfromtxt`), which allowed a fast prototyping and easy testing of different optimization algorithms. One configuration file performing a single forward simulation with CardioMechanics (xml data format) was set up for each of the two geometries and served as template for the forward simulations of the optimization. A second configuration file in python format controls parameters of the optimization script and prescribes algorithm, initial values, parameter coupling, corresponding xml keys and the template xml file to use (Figure 6.6).

The `scipy.optimize` framework expects a function that takes a set of optimization parameters and returns an objective value. That function needs to be provided by the optimization script. One evaluation of the objective function consists of the following steps, each split into a separate function: (i) `setup_opti_sim()`: Convert the optimization parameters to Guccione parameters by applying parameter coupling, read the template xml, write an new xml file with modified parameters for the current forward simulation, (ii) `run_opti_sim()`: call CardioMechanics with the generated xml file in order to run the forward simulation in a subprocess, (iii) `evaluate_data()`: read csv file containing pV pairs, compute the targeted absolute pV curve, interpolate simulated pressures to the equidistant volumes used in the target pV curve, compute the sum of squared differences

6. Improving the Physiological Behavior

```

class ConfigHeart17:
    """ parameters 4ch heart, LV inflation with parameter coupling """
    INPUTFILE = "inflate_static.xml"
    OPTIFILE = "inflate_static_opti.xml"
    KLOTZFILE = "inflate_static_data/PressureVolume.dat"
    EVALINDEX=131
    MODELVOLUMEFILE = "inflate_static_data/ModelVolume.dat"
    _xmltags_short = ["C", "b1", "b2", "b3"]
    XMLTAGS = ['Materials/Mat_30/Guccione/' + t for t in _xmltags_short] \
        + ['Materials/Mat_31/Guccione/' + t for t in _xmltags_short]
    COUPLED_INDICES = [0, 1, 1, 1, 0, 1, 1, 1]
    COUPLED_FACTORS = [1.0, 1.0, 0.4, 0.7, 1.0, 1.0, 0.4, 0.7]
    X_INITIAL = [150., 18.0]
    TARGET_VOLUME_RATIO = 2.570 # v0/v8 = .50
    ALGORITHM = "NelderMead"
    FORWARD_SIM_CMD = "(mpirun -np 8 CardioMechanics -settings \
        + OPTIFILE + " -verbose -pc_factor_mat_solver_package mumps) >&1"

```

Figure 6.6.: Configuration of the optimization with the 4ch geometry. It contains paths to input and output files, the cavity to use for evaluation (EVALINDEX), configuration of parameter coupling (XMLTAGS, COUPLED_INDICES, COUPLED_FACTORS), configuration of the optimization (X_INITIAL, ALGORITHM), and the command line for running a forward simulation (FORWARD_SIM_CMD).

at these volumes and return it for use as objective value. If the chosen volume penalty is too small, the largest myocardial volume change is not reached in the last step, but in between. Moreover, physiologically the volume change should stay below 10%. In order to detect misbehaviors of these kinds, the objective function additionally evaluates and tracks all myocardial volumes and warns about an unsuited penalty parameter if the volume change does not behave as expected. If the cavity volume decreases below initial volume, a warning message is shown as well.

For use with derivative-based methods, the script is also capable of computing finite difference approximations via multiple simulation runs. To prevent unnecessary recomputations, all simulation results were stored in a cache and taken whenever the objective function is called with a certain parameter set a second time. In that case, a dictionary holds for each parameter set the previously computed pressure-volume pairs in order to avoid running simulations with the same parameter set multiple times. But also for Nelder-Mead, one optimization step takes multiple forward simulations. Nelder-Mead evaluates in each step the objective at all simplex corners, thereby eventually calling the objective function multiple times for a specific parameter set.

Major limitation of the python approach was, that the provided implementation of Nelder-Mead is not designed for evaluation of a computationally expensive objective function. The actual optimization overhead itself is tiny compared to the computational costs of the forward simulation. It is not capable of doing multiple function calls at once, even if the current optimization step needs the evaluation of multiple sets of optimization parameters. Moreover, it does not avoid multiple function calls with the same parameter

6. Improving the Physiological Behavior

set, which typically happens occasionally during derivative approximation with finite differences. Remedy of the first point was to run the actual forward simulation in parallel instead. Solution of the second point was to introduce in the objective function a cache of optimization parameters and corresponding results. Before creating an xml and running an expensive forward simulation, the respective function first checks if the passed parameter set is already in the cache and returns the corresponding simulation results. If not, the results are computed by a full run of the forward simulation and added to the cache. Moreover, such a caching strategy additionally allowed to continue previous optimization runs, which is quite useful on a parallel computing clusters where the runtime of a job is limited and did not provide a converged solution yet.

The external approach to modify the xml input file accordingly allows to estimate suitable parameters not only for passive tissue, but rather for any model that exposes its parameters to the configuration file, including circulatory system and pericardium properties.

6.1.4. Results

The optimization progress of the analytical ventricle geometry “1ch” is shown in Figures 6.9 and 6.10. Figure 6.9 shows the progress of the parameters C and B for each of the three different start values over optimization iteration $kIter$. C is an order of magnitude larger than B and was scaled by a factor of 0.1 for visualization in a common plot. Figure 6.10 shows the progress of the objective function over $kIter$ for three different start values. The values of the initial parameter sets are not included in the plot, $kIter = 0$ shows the result of the first optimization step instead. After 60 iterations, the exact estimated parameters are $C = 264.83, 263.96$ and 263.91 , and $B = 8.489, 8.498$ and 8.498 with an objective of $2.3402, 2.3395$ and 2.3421 for the three different initial parameters, respectively. The plot shows no further improvement of the objective over the remaining 20 iterations. The final states of all three cases are close together, in terms of both parameters as well as the objective values. The computation time of a single forward simulation on eight cores varied between 34 s and 1 min 30 s, and took the same 1 min 16 s for the final states of all three start parameters.

In case of the real heart geometry “4ch”, the progress of both parameters and the objective function over the optimization iteration is shown in Figure 6.11. Again, C was scaled for visualization by a factor of 0.1. During the first 10 iterations, the objective decreased quickly and fell below a value of 10 after only 7 optimization steps. After that, the objective continued to decrease and reached at iteration 13 already an objective of 1.353 with corresponding parameter set $C = 162.776$ and $B = 11.01$. The final objective value was almost reached, however further changes in C and B were still observed. The last iteration resulted in the parameter set $C = 142.40$ and $B = 11.53$, and an objective value of 0.8933. The resulting absolute pV curves from the initial, 13th and final optimization step are shown in Figure 6.7. The two curves from step 13 and step 33 are hard to distinguish and show the largest visual difference between 180 ml and 200 ml. The final value of the objective is still 34.0% lower than in step 13 and 61.8% lower than the final one in the analytical ventricle. Compared to the analytical ventricle, a single

6. Improving the Physiological Behavior

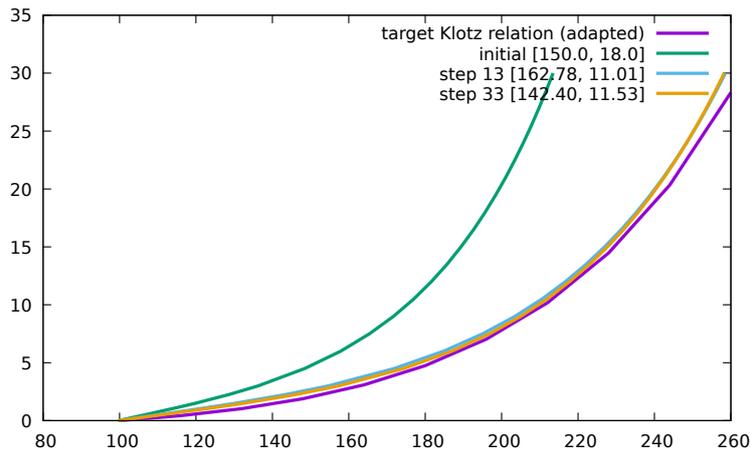


Figure 6.7.: Optimization progress of absolute pV curves from the 4ch setup, comparing the initial parameter set (green), one obtained during optimization (blue) and the final parameter set (orange) with the target curve (purple). The x-axis gives cavity volume (ml) while the y-axis shows pressure (mmHg).

forward simulation took significantly longer, and within the time limit of 24 hours only 33 iterations and 68 forward simulations could be computed.

6.1.5. Discussion

The simulations using the ventricle geometry 1ch showed that all three different start values lead to the same final parameter set, despite the start values varied in a large range of C between 50 and 5000. This is a good indication that a unique solution to the problem of finding passive material parameters exists, the solution finding process is independent from the start values and that the algorithm did not run into local minima, making Nelder-Mead an appropriate optimization algorithm.

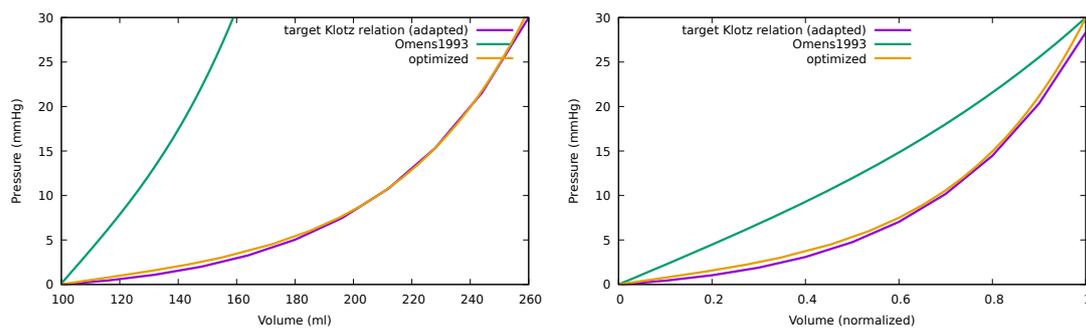


Figure 6.8.: Klotz curve of the 4ch geometry with the originally used parameter set by Omens before (green) and after optimization (orange).

6. Improving the Physiological Behavior

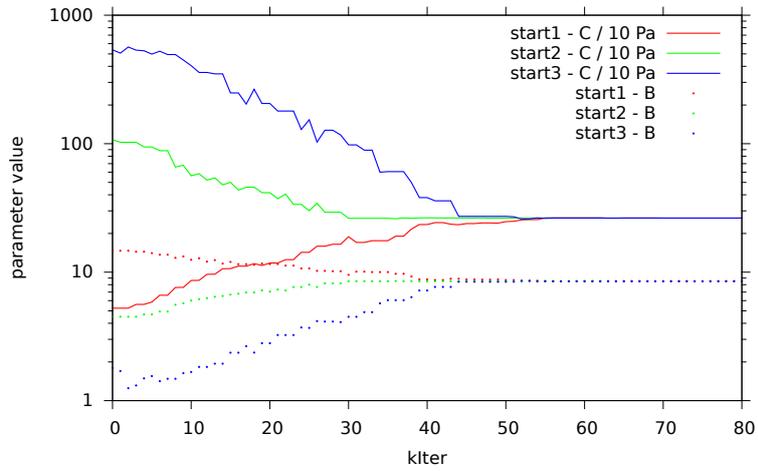


Figure 6.9.: Optimization progress for the 1ch ventricle geometry with three different start values. Parameters B (solid) and C (dots) are shown for each iteration of the Nelder-Mead algorithm. Initial [C,B] values are start1 [50., 14.], start2 [1000., 5.], and start3 [5000., 2.] and final solutions average [264.2, 8.495].

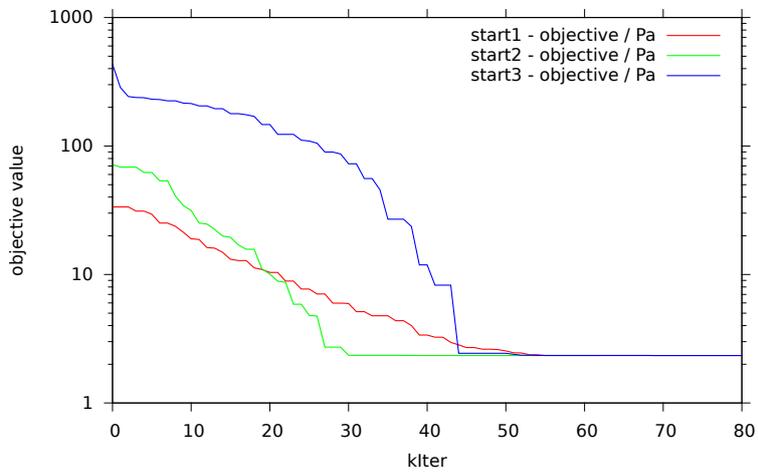


Figure 6.10.: Progress of the evaluated objective function of the 1ch ventricle geometry for the three different start values.

6. Improving the Physiological Behavior

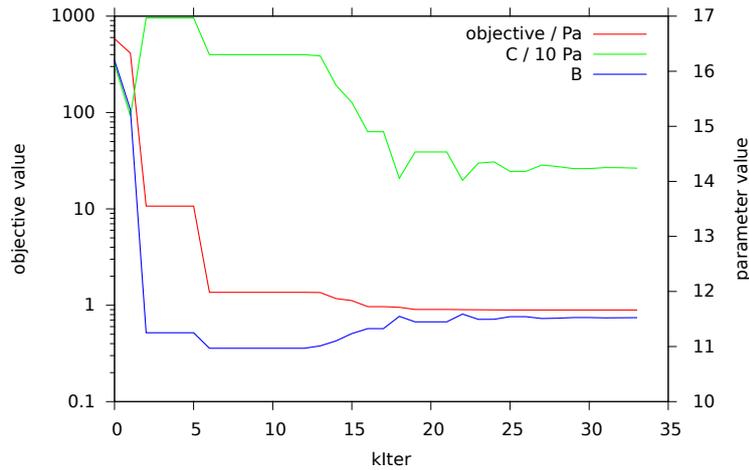


Figure 6.11.: Progress of the parameters C and B (green, blue) as well as the objective function (red) for the 4ch heart geometry.

In case of the segmentation-based four-chamber heart model, good parameters were already found after only a couple of iterations and the solution's objective values did not significantly change during the next iterations.

The final parameters between the two geometries were different (average of 1ch: [264.2, 8.495] vs 4ch: [142.40, 11.53]), thereby an important conclusion is the necessity to fit parameters for each geometry individually. One possible reason could be, that the size of the approximated ventricles is different, or just the ratio between wall thickness and long axis.

The objective values between different geometries can be compared directly, the objective value evaluates ten pressure differences with the exact same target pressures. Within ten iterations, the objective of the four-chamber geometry was already lower than the final objective from the simulations using the analytical ventricle. One possible reason is that the 4ch setup is closer to the experimental setup by Klotz et al. (2006) and thereby better able to reproduce their data, which includes three additional passively deformed chambers and realistic fiber orientations.

The dimensions of the analytical ventricle 1ch were the same as taken in Land et al. (2015) for verifying the correctness of the code and are certainly too small to represent a human ventricle (volumes of 2.46 ml and 6.37 ml before and after inflation). Additionally its mesh resolution was chosen to obtain a fast forward simulation in first place, but two quadratic elements transmurally might be too coarse to sufficiently represent the prescribed fiber orientation rotation from +90 to -90 degree for that model. That specific ventricle however was only used to check reproducibility for different initial values. It represent a well-described and reproducible geometry that readily exists within many groups that participated in the benchmark publication by Land et al. (2015). The simple ventricle inflation setup was solved by all of them, which makes it much easier to reproduce the method and its results.

6. Improving the Physiological Behavior

In case of the human-sized four-chamber simulation 4ch, the parameter C changes from $C = 142.4$ to $C = 162.8$ between iteration 13 and 33, however both are already better than the final solution of the 1ch setup by means of the objective value. The parameter B seems to be determinable more reliably and changes only from 11.53 to 11.01. This difference is measurable, but not visible when comparing the two Klotz curves visually in Figure 6.7.

Genet et al. (2014) used a similar approach to determine parameters for the Guccione model at five different patients. They applied a similar parameter coupling, but a different optimization strategy which estimates C and B repetitively one after another. Their forward simulation contained only the left ventricle with a hexahedral mesh of similar resolution ($5k$ nodes, $3.5k$ elements) to the analytical ventricle 1ch in this study ($4.2k$ nodes, $3.4k$ elements). They used a fixed baseplane as boundary condition and as unloaded geometry the segmentation of the end-systolic state, which both are similar to the 1ch setup. Moreover, they did not focus only on the pure passive behavior, but estimated after their optimization process the amount of active stress as well. Interestingly, they inflated the ventricle only to 9 mmHg instead of 30 mmHg and their final parameter set also showed a bad fit for pressures below 3 mmHg. The parameter set determined in that publication corresponds to $C = 57.5 \pm 4$ and $B = 14.4 \pm 3.18$, which is not similar to the results from the similar forward simulation setup “1ch” ($C = 264.2$, $B = 8.495$). It shows a more than four times smaller value for C compared to this study. However, the ventricular volume of the 1ch setup is very different from a human ventricle and has rather extreme fiber orientation, so it is probably not the best suited for comparison. The final parameters obtained from the 4ch setup ($C = 162.8$, $B = 11.01$) are closer, but still differ in C by a factor of more than two, which seems to contradict their estimated inter-patient variances. The 4ch setup uses a different geometry with e.g. the right ventricle influencing deformations mostly at large pressures. The factor B in contrast, which seemed to be determinable more reliably and showed less variation during the almost-converged steps of the optimization, fits with a factor of 11.01 into the variability range.

Outlook It was found that a gradient-based algorithm did not perform nearly as well as the Nelder-Mead algorithm. For a further runtime improvement of the optimization process, knowledge about deformation states under similar material parameters for a specific pressure could be incorporated into the mechanics solver in the future. Providing such a more intelligent initial guess to the static solver’s Newton method (the one that solves the equilibrium of forces) would produce a faster convergence and fewer Newton iterations by already starting in the range of locally quadratic convergence.

One of the main reasons for introducing the parameter coupling was to prevent disappearing shear forces. Experimental data from shear tests at myocardial tissue patches from pig hearts is available by Dokos et al. (2002). That data was used by Schmid et al. (2008) to estimate parameter sets for different material laws, including the Costa law, to produce a correct local behavior. The 7-parameter set of the Costa law was reduced by Xi et al. (2011) to parameters for the Guccione law ($C = 94.5$, $b_1 = 28.9$,

6. Improving the Physiological Behavior

$b_2 = 13.5$, $b_3 = 13.3$). Its parameter C is only 42% below the C estimated for the 4ch setup, but the parameter b_1 is more than twice as high. The factors between the $b_{1,2,3}$ are different and b_2 and b_3 are very similar to each other, however the resulting factor $b_3 = 0.46 b_1$ is similar to one used during parameter coupling ($b_3 = 0.40 b_1$). It is however difficult to derive the local behavior just by comparison of the parameters. Such data from local measurements could be incorporated in the objective as well, to produce not only a correct global but also a correct local behavior.

Concludingly, the parameter estimation of a four-chamber heart directly using CardioMechanics and a four-chamber heart geometry for the forward simulations is a step forward towards an accurate reproduction of the global behavior of the heart. Coupling the mechanics model to a lumped circulatory system model relies on a correct global behavior and exchanges information only by non-local values using pressures, volumes and their derivatives per chamber. Especially during ventricular relaxation or filling phase, when the ventricles are completely free from contracting forces, the passive behavior profits a lot from an accurate description of the passive material properties.

This also has some positive side-effects in other areas as well. The passive tissue behaves now weaker producing a larger inflated volume compared to the originally used Omens parameter set (Figure 6.8), or vice versa a smaller unloaded volume with a prescribed inflated volume (e.g. by segmentation). Such a smaller unloaded volume provides a larger initial strain. This finally allows to use better tension models that include stretch dependency.

6.2. Application II – Length-dependent Active Behavior

6.2.1. Introduction

One of the main purposes of the heart is to drive the circulatory system. Ejection of blood depends on pressure difference between ventricle and aorta. That pressure is created by contraction of the ventricles, a larger pressure gradient results in a faster ejection of blood. If the amount of consumed energy is the same, this leads to a more efficient pumping function. On cell level, contraction is triggered by electrical excitation and tension develops in a complex process respecting changing states of several ion channels, Ca concentration and the current length of the fiber cell. It is experimentally known that an isolated cardiomyocyte produces more tension when it is elongated before activation (Starling effect on cell level). A similar effect exists on organ level, where an increase in ventricular filling pressure and volume creates more ejection volume under heavy load (Starling effect on organ level). An adequate length-dependent tension model for application in a whole-organ environment should be able to reflect the mechanical Starling effect on organ level.

The electrical excitation on organ level is inhomogeneous. It starts near the apex and spreads towards the baseplane, which can be measured by a multi-electrode catheter measuring potentials at different locations simultaneously (in-vivo) or by exploiting fluorescence effects (ex-vivo). A completely homogeneous activation is not possible due to limitation of conductivity, but mechanisms like the fast-conducting Purkinje tree reduce

6. Improving the Physiological Behavior

the duration of ventricular activation. Mechanically however, the situation is unclear as time courses of local mechanical contraction are difficult to measure in an organ level setup. Electrophysiological models exist that resemble tension development for isolated cells, but especially the ones that respect Length-dependency via modeling biological processes inside the cell do not deliver as expected in simulations on organ level as their output does not scale well. A homogeneous mechanical activation would have several advantages. If one half of the ventricle contracts first and the other half at a later time, merely redistributes blood and stores the energy as potential energy in the currently inactive part (which has a large compliance at that time of no contraction) instead of building up a pressure difference. This happens in a bundle branch block and makes it so dangerous for the patient, the ventricles lose their ability to contract in synchrony and rather produce a lot of septal motion instead of efficiently ejecting blood. When all cells contract at the same time however, the obtainable peak pressure is higher, or requires less chemical energy to obtain the same peak pressure (efficiency increase). Potentially, a homogeneous mechanical contraction peak is the most efficient and length-dependent tension might allow to explain the apparent discrepancy to inhomogeneous electrical activation via the effect of myofibers with increased pre-stretch reaching their tension peak earlier. From the computational point of view, a homogeneous mechanical activation allows to simplify the modeling process in cases without detailed electrophysiological modeling and removes the need for generation of an activation time map and all related uncertainties while still maintaining some relevant effects on the mechanical side. The availability of a phenomenologically modeled length-dependent tension model published by Land et al. (2017) allows to answer a number of open questions.

Aim of this project is to clarify four main hypotheses. Hypothesis 1, homogeneous contraction improves the mechanical efficiency of the motion. Hypothesis 2, length-dependent tension development is a mechanism to smooth out inhomogeneities due to differences in activation time, pre-stretch, partially thinner walls or passive stiffness. Regions with lower initial stretch would develop a low amount of tension, however they experience a larger passive deformation due to contracting neighboring cells, which increases their pre-stretch and thereby their tension. Hypothesis 3, the Starling effect on cell level leads to Starling effect on organ level. Hypothesis 4, the effect of respecting length-dependency is more valuable than imposing inhomogeneous electrical activation times on mechanical activation with a pre-computed model.

This project tries to quantify the advantages of a length-dependent tension model over a pre-computed model. It is not focused on competing with state-of-the-art models on the electrophysiological side on cell level, rather on providing general insights that arise from respecting length-dependency. Input of the used length-dependent tension model is a prescribed function related with calcium, which makes a future extension to a differently obtained source possible – if needed. This project was realized by supervision of a student thesis Müller (2017). The investigations in this section were conducted before the investigations leading to the previous section, therefore do not contain the findings for the passive material properties from Section 6.1 and use the modeling from Fritz et al. (2014) instead.

6.2.2. Methods

For modeling of tension development in the cardiac cells, two different approaches are used.

The first tension model is the “Double-Hill” tension model. It uses a constant function as described in Section 3.7 to prescribe the tension in every single element. Advantages are a simple implementation and the fact that the resulting elastance of the chamber is under control (the elastance curve follows the tension curve). Drawbacks are that such a model does not inherently represent the Starling effect, and there is the risk that single elements experience more contraction than what is physiologically possible (must be above 0.84 times initial length).

The second tension model is called the “Lumens tension model”. It uses an ODE-based formulation solved by a Runge-Kutta method in each time step, and has the current length in fiber direction as one of its input parameters. It is described in Section 3.7 as well. The original formulation (Lumens et al., 2009, Appendix B therein) could be used almost unmodified, the only changes to the original model are related to computation with a 3d FEM code: Since the simulations require to start in an equilibrium state, the active stress needs to be assumed to be zero. Therefore, the parameter “Calcium concentration at rest” was set to $C_{rest} = 0$. The zero-dimensional equation for length-dependent passive stress used by Lumens et al. (2009) is replaced with the Guccione material law computed from 3d deformation tensor during the FEM simulation.

In general, tension models have in common that they all compute a scalar-valued *active tension* value T_{active} , which is assumed to be the stress in fiber direction and gets converted to a 3d *active stress* matrix S_{active} by addition at the respective position of the total stress tensor:

$$S = S_{\text{passive}} + S_{\text{active}}, \quad S_{\text{active}} = \begin{pmatrix} T_{\text{active}} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

For the dynamic dependency on the current fiber length, a new framework for length-dependent tension models needed to be implemented and is described in Section 4.1. It’s using the similarity between passive/constitutive models and active/tension models. Both depend on current deformation \mathbf{E} and contribute to the stress tensor by an additive decomposition $S(\mathbf{E}) = S_{\text{passive}}(\mathbf{E}) + S_{\text{active}}(\mathbf{E})$. In the previous implementation, the Cardiac Biomechanics framework had only one global vector to store precomputed active tension values for each element and S_{active} did not depend on \mathbf{E} . Whenever a tension was needed, the address pointing to the value was exchanged, the actual value however was constant. The tension vector could only be updated once per step from a pre-computed function or an external file – therefore it had no direct length-dependency.

In the reworked implementation, the single value is replaced by an object-oriented hierarchy of tension models that share a single interface. Instead of a single constant value, a separate tension object is assigned to each element and provides a function that can be evaluated whenever an element needs an up-to-date value that matches the current state of the deformation. Depending on the tension model, the function call

that is responsible for evaluation might implement a complete time integration (ode-based) including correct handling of changing or failing time steps from the FEM Solver ('step-back' events in CardioMechanics), or it returns only the simple result of a explicit function evaluation (function-based tension). Figure 6.12 shows a class diagram of the implemented tension models, including the Lumens model realized in `CBLumensTension` and the DoubleHill model realized as a generic `CBTensionFromFunction` paired with `DoubleHill` as corresponding explicit function. Figure 6.13 gives the other options available as explicit function. Concerning parallelization, the vector of elements already existed in a distributed manner, thereby this approach inherits the existing parallelization by design.

Finally, an LAT-reader was implemented that allows to delay the local time variable of each element's tension model by a separate duration. A map of local activation times can be specified element-wise in a simple text format as depicted in Figure 6.15.

6.2.3. Simulation Setups and Evaluation

Four different simulations were conducted to compare the effects on the mechanical function of (i) a length-dependent tension model, and of (ii) a locally inhomogeneous activation of ventricular cells.

The geometry includes all four chambers, the valve plane and truncated representations of the aorta and pulmonary veins as described in Section 4.5. Since the Lumens model generates a considerably larger amount of force when stretched, a consistent amount of pre-stretch of about 10% in all cells is important. An unloaded state was not prescribed, instead both ventricles were inflated to 8 mmHg over a time duration of 500 ms to ensure a positive pre-stretch of each cell, needed by the Lumens tension model (Figure 6.14). Pressure forces were computed with the fully coupled "WholeHeart" model of the circulatory system. In all setups, pericardial forces were intentionally not present to avoid side effects in the evaluation due to possible interactions with the length-dependent tension model.

Passive forces consisted of Mooney-Rivlin material for the atria and Guccione material for the ventricles, with parameters as estimated in Section 6.1 for a correct passive behavior. Active forces for the atria are computed by the DoubleHill model with $T_{max} = 35$ kPa and all cells are activated at the same time.

For the ventricles, active forces are either one of Lumens or DoubleHill model, depending on the setup. The Lumens tension model was used with the exact same original parameters as described in Section 3.7 in Table 3.4 including the tension scaling factor of 120 kPa. The only difference was $C_{rest} = 0$ to get an initial state at equilibrium and being able to start the simulation. The DoubleHill tension model was used in the ventricles with a maximal tension of 60 kPa (right) and 80 kPa (left) to reflect the physiologically observed lower generated right ventricular overall force. Parameters for the DoubleHill model are found in Table 6.3 and are the same as in Section 3.7. Earliest ventricular activation was delayed to begin 150 ms after the atrial activation, plus an optional additional local delay depending on the setup. That local delay was defined by a map of local activation times (LAT) containing a single float value for each cell separately, that was generated

6. Improving the Physiological Behavior

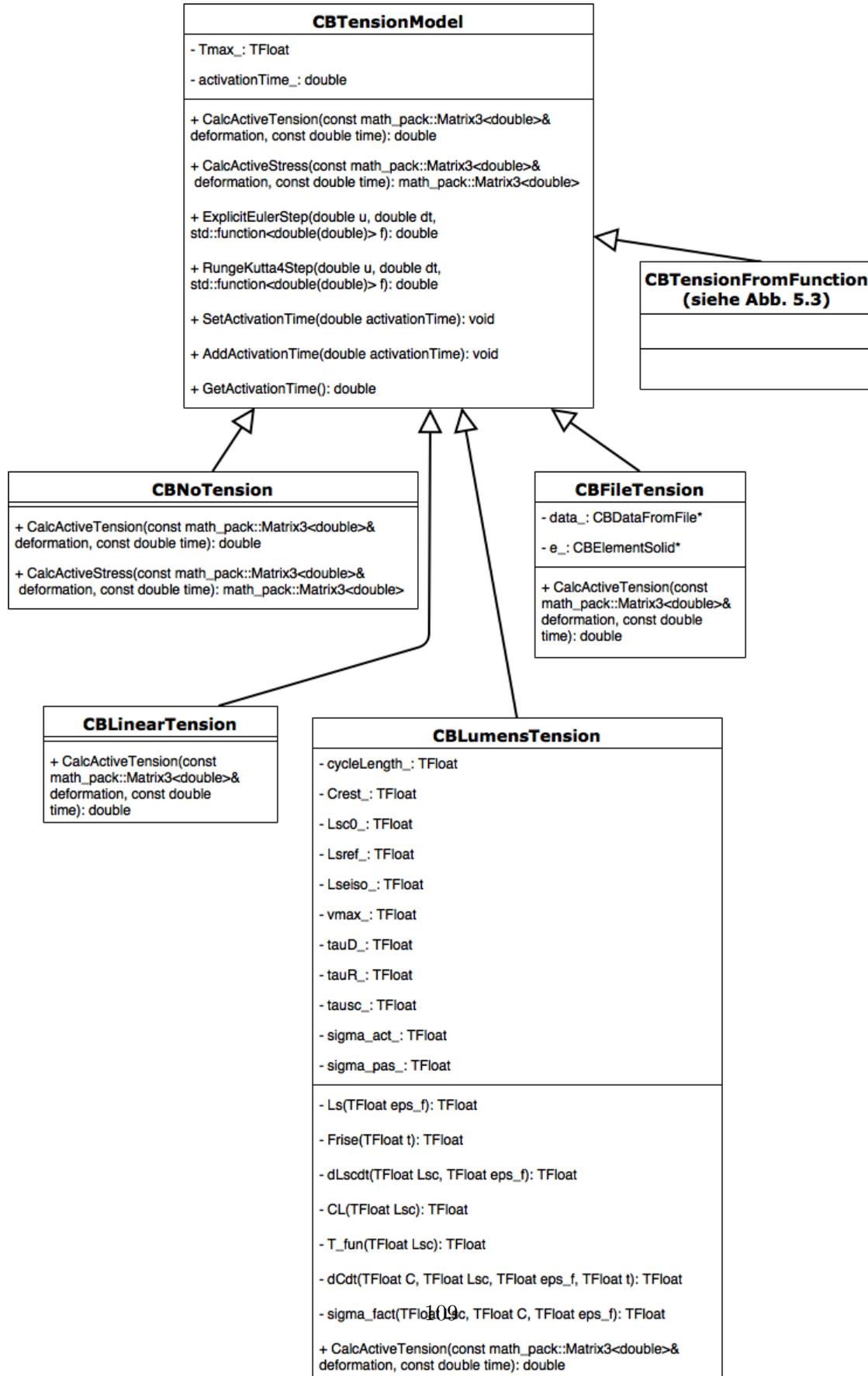


Figure 6.12.: Structure of the available tension models, from Müller (2017).

6. Improving the Physiological Behavior

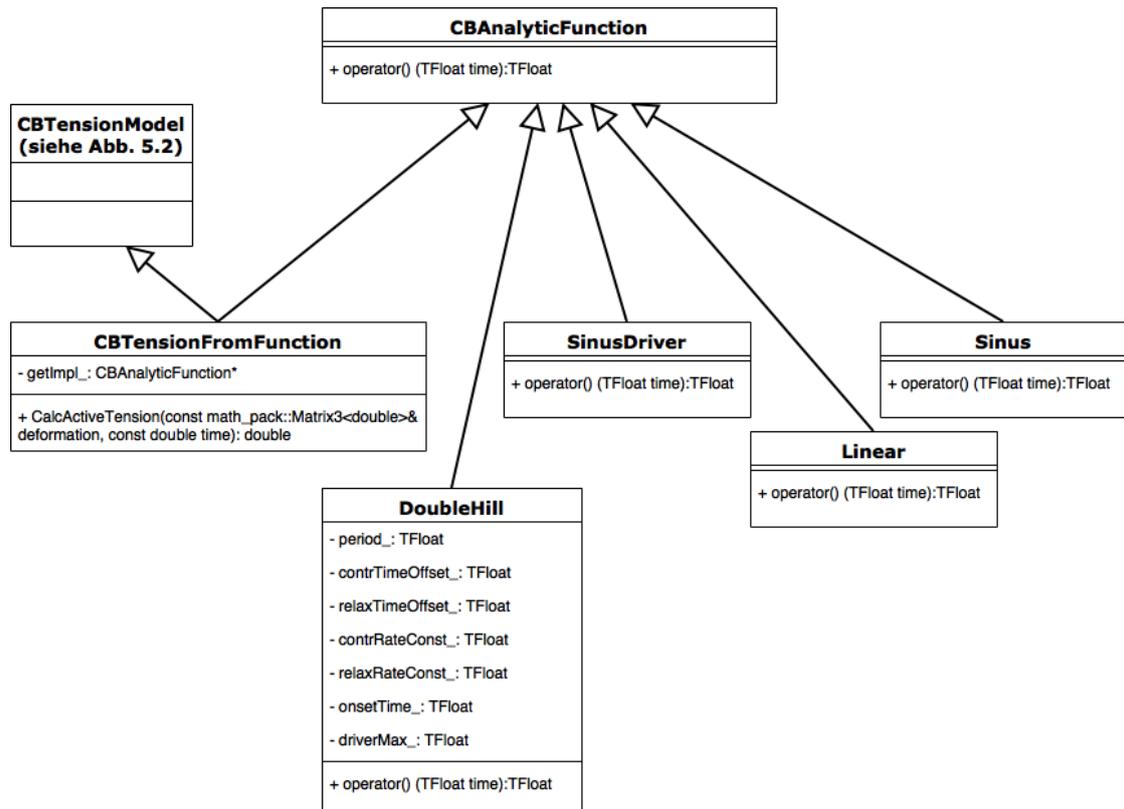


Figure 6.13.: Hierarchy of available analytical functions that can be used as simple tension models via “CBTensionFromFunction”, from Müller (2017).

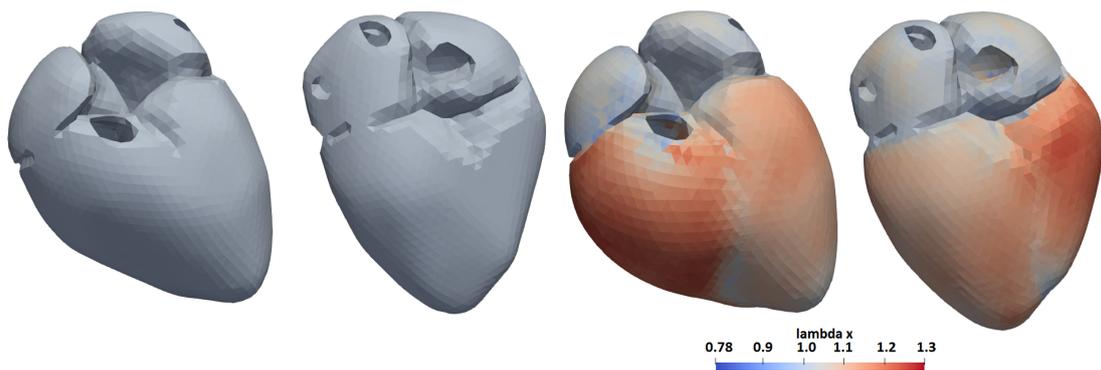


Figure 6.14.: Relative stretch in fiber direction before (left half) and after inflation (right half), from Müller (2017).

6. Improving the Physiological Behavior

Table 6.3.: Parameters that were used for the DoubleHill tension model, from section 3.7.

Parameter	atrium	ventricle (left/right)
ContrTimeOffset	0.042	0.215
RelaxTimeOffset	0.138	0.362
ContrRateConst	1.99	1.32
RelaxRateConst	11.2	21.9
OnsetTime	0.0	0.15

by solving the eikonal equation using the fast marching algorithm with initial excitation triggered at the left ventricular apex and an intentionally simple, homogeneously isotropic conduction velocity adapted to produce a ventricular activation spread over 120 ms (which corresponds to the length of QRS complex in ECG signal). Interestingly, taking the long axis position from a principal component analysis of both ventricles and scaling it to the desired time duration created almost the same pattern.

In summary, this leads to the following four setups:

- *DoubleHillOhneLAT*: double hill tension model, homogeneous activation
- *DoubleHillMitLAT*: double hill tension model, inhomogeneous activation
- *LumensOhneLAT*: Lumens tension model, homogeneous activation
- *LumensMitLAT*: Lumens tension model, inhomogeneous activation

The first two setups generate tension simply by evaluating an analytical function (length-independent), the latter two solve an ODE by time integration (length-dependent). Each simulation contained 5 heart beats and was conducted using the NewmarkBeta solver on tetrahedral finite elements of second order. The total simulated time of 4.5 s contains 500 ms initial inflation followed by 5 beats à 800ms.

For comparison of the simulation results, pressure and volume curves over time were created, as well as pressure-volume loops of all setups. Moreover, stretch in fiber direction λ_x was computed for each cell (x -direction in the local fiber coordinate system). Evaluated is the relative change of length compared to the undeformed mechanical reference state as described under λ_f in Section 3.1.2 (deformation measures) and Section 3.7 (tension models).

6.2.4. Results

Figure 6.15 shows the results from solving the eikonal equation to compute the activation times for the simulations with inhomogeneous activation.

Figure 6.16 shows the distribution of peak tension in the last beat as individual maximum of each cell. For the stretch-independent DoubleHill tension model, the peak tension is for all cells exactly the same with homogeneous as well as inhomogeneous activation. The tension patterns in Figure 6.17 show for the homogeneous case a perfectly

6. Improving the Physiological Behavior

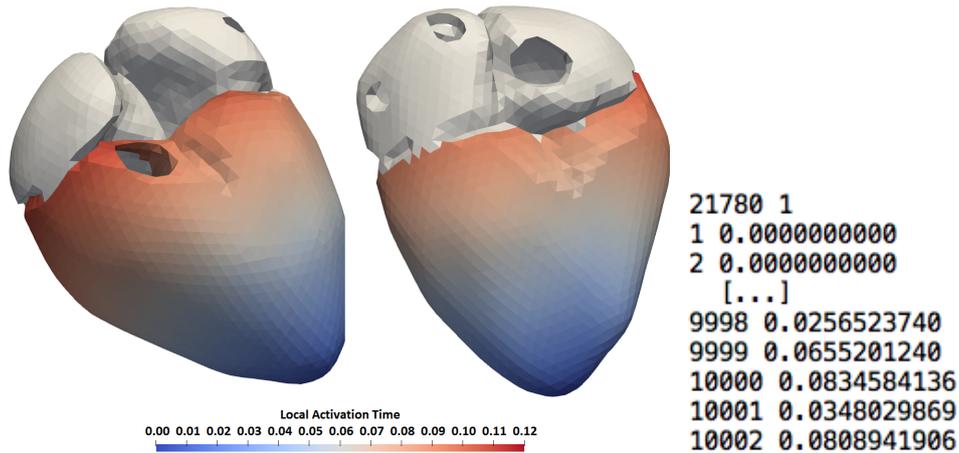


Figure 6.15.: Map of local activation times in seconds used to delay ventricular activation (left) and an extract of the data format for activation times per cell id (bottom right), adapted from Müller (2017).

homogeneous pattern in each time step. In case of inhomogeneous activation, there is a clear local delay of tension development with the contraction motion starting near the apex first.

For the Lumens tension model, the tension maxima are distributed clearly inhomogeneous in both activation cases, even for the setup with homogeneous activation. The resulting tension patterns look similar in both simulations, with and without using local activation times. The influence of local activation times on the motion of the contraction is less obvious than it is with the DoubleHill tension model. A contraction from the apex is – different from DoubleHill model – not clearly visible in Figure 6.17, however the patterns seem to be shifted by 50 ms. Instead, the pattern of developed peak tension is similar to the pre-stretch after initial inflation as shown in Figure 6.19.

The local stretch λ_x during contraction (this is not visualized) is consistently above 80% for the Lumens tension model, which is in accordance to the physiologically imposed limits mentioned in Section 2.2. In the simulations with the DoubleHill tension model in contrast, the tension “develops” independently from initial or current stretch and partially drops significantly below 80%, reaching even below 50% in some cells.

Figure 6.18 compares spatial tension distribution of the four setups in a snapshot taken during relaxation side by side. The two pictures of the setups with inhomogeneous activation times are shifted by 50 ms to take an overall later average activation into account. Notably, the simulation with DoubleHill and inhomogeneous activation times (DoubleHillMitLAT) behaves differently from the other three and still shows an ongoing contraction with reduced radius near the valve plane, while the remaining ventricle already relaxes. That jellyfish-like motion is not visible in the homogeneous case, nor do both simulations with the Lumens tension model show such a behavior. For the Lumens model, the spatial tension pattern does not change by including locally inhomogeneous

6. Improving the Physiological Behavior

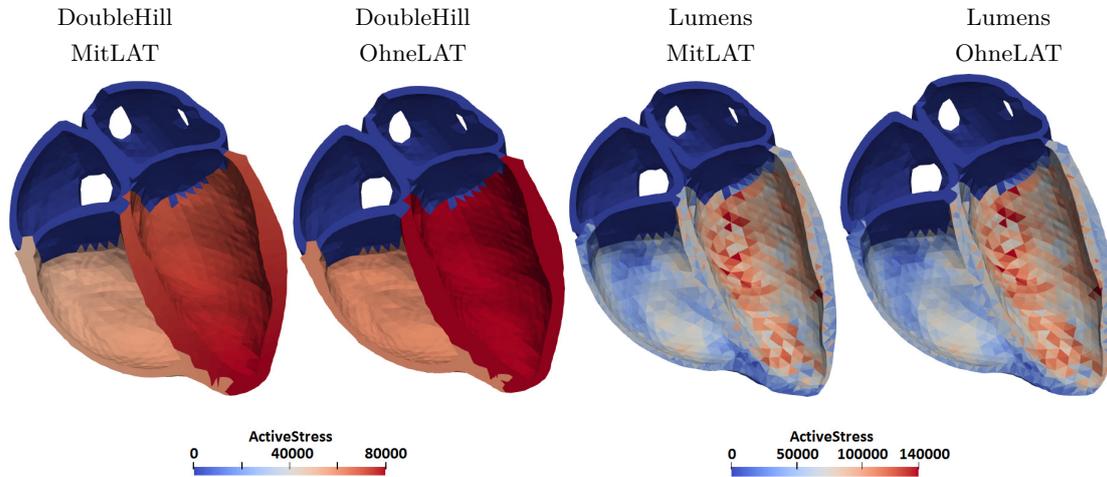


Figure 6.16.: Comparison of developed peak tension distributions for all four simulations on the endocardium obtained from the last beat, adapted from Müller (2017).

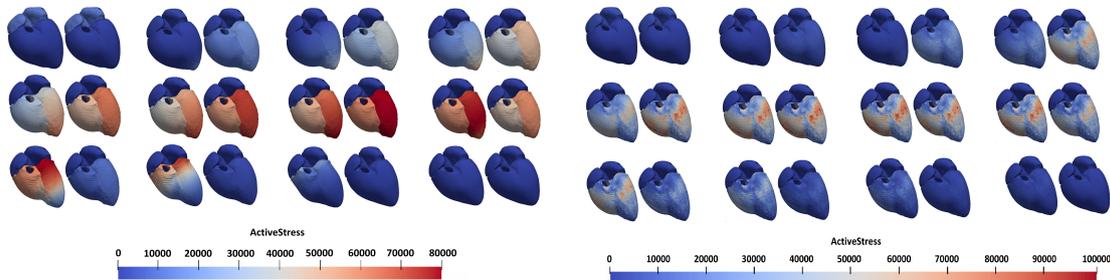


Figure 6.17.: Effect of activation times on the patterns of developed tensions starting at 650 ms in 50 ms steps (left to right, then top to bottom) for DoubleHill (left half) and Lumens (right half) with each pair containing homogeneous (left) and inhomogeneous activation (right), adapted from Müller (2017).

6. Improving the Physiological Behavior

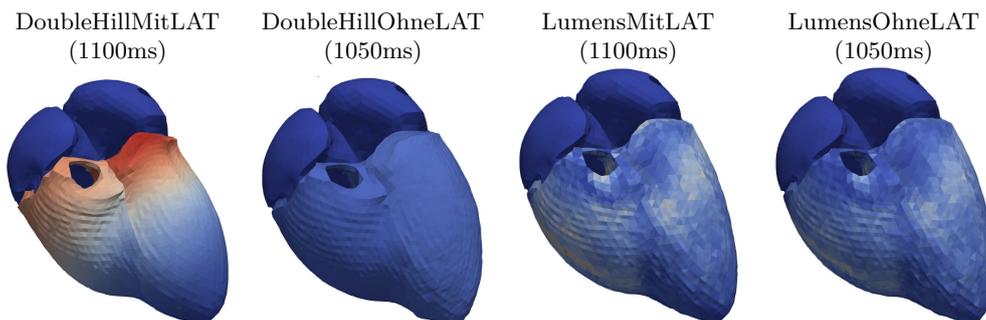


Figure 6.18.: Comparison of tension distribution during relaxation of all four setups. For a legend see Figure 6.17.

activation, both patterns are visually identical besides the aforementioned temporal delay of 50 ms between the two visualizations.

Figure 6.19 shows stretch λ_x in fiber direction after initial inflation (before first systole, $t = 500$ ms) and developed tension with the Lumens model with homogeneous activation times (end of first systole, $t = 960$ ms). Certain similarities between the patterns of initial pre-stretch and of developed peak tension are visible. The right ventricle has a rather uniform pre-stretch up to 1.24 and shows an even tension distribution during contraction. In the upper left ventricle, there is a distinct area with pre-stretch around 1.16, which can be re-identified in the tension distribution as an area with developed tension of above 60 kPa. In the remainder region closer to the apex, the pre-stretch is at most 1.1 and develops tensions below 40 kPa. Besides the initial stretch, in that simulation all cells have the exact same properties including activation time, tension model parameters and material properties.

The plots in Figure 6.20 show pressure and chamber volume over time for the left ventricle of the five heart beats. Both values do not start in circulatory system's equilibrium state and change in subsequent beats for all four simulations. The volume curves of the simulations using Lumens tension are lower compared to the curves from the simulations with DoubleHill tension model. The same behavior is observed in pressure curves and peak pressure, which are consistently lower in the setups with Lumens than with DoubleHill. Moreover, both tension models reach their pressure peaks at a later point in time when using inhomogeneous activation times compared to homogeneous ones. The pV loops of all four chambers are shown in Figure 6.21. Developed energy of the left ventricle (area of pV loop diagram) is larger for the DoubleHill tension model than it is for the Lumens tension model, indicating that the same amount of tension might be used more efficiently when distributed homogeneously. The atrial pV diagrams behave similar for all four simulations.

Finally, Table 6.4 shows key values evaluated from the last beat of all simulation setups. Maximum and ejection volumes decrease in both ventricles when locally delayed activation is used, the same holds true for the total heart volume. This is consistently observed for both tension models, and the decrease is in a similar range for both tension

6. Improving the Physiological Behavior

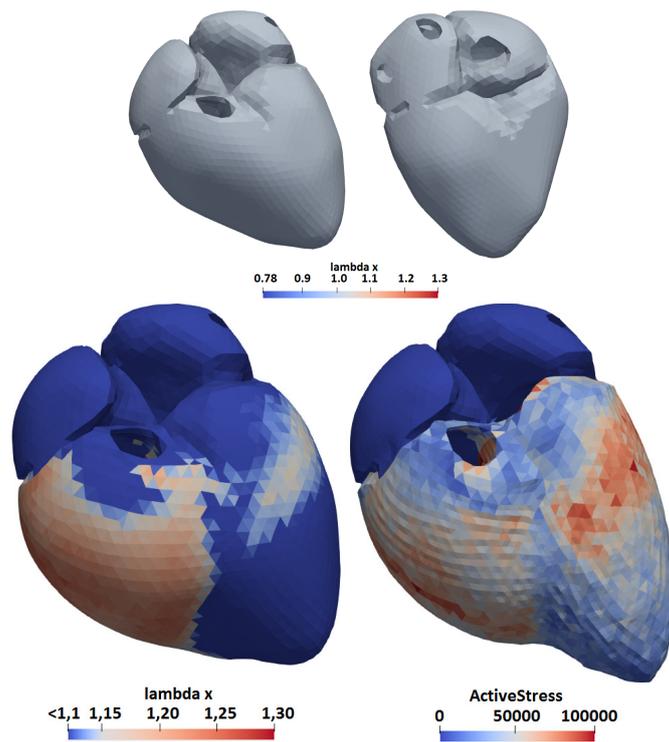


Figure 6.19.: Patterns of stretch after inflation of the initial geometry to end-diastolic pressure (500 ms) and developed peak tension for the simulation with Lumens tension model and homogeneous activation (960 ms), from Müller (2017).

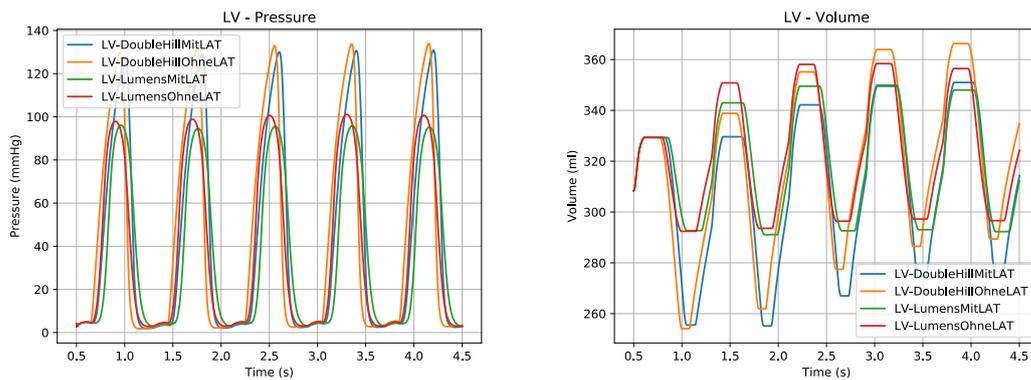


Figure 6.20.: Left ventricular pressure and volume time courses comparing the four simulation setups, adapted from Müller (2017).

6. Improving the Physiological Behavior

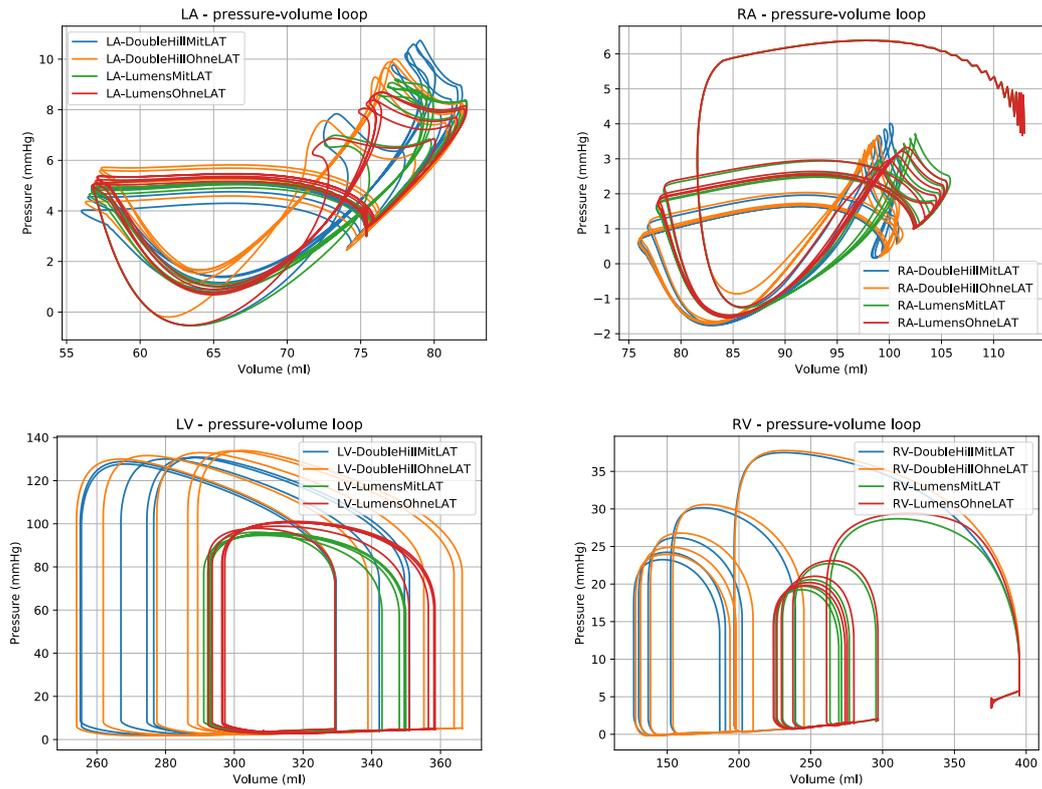


Figure 6.21.: Pressure-volume loops of all four chambers comparing the four simulation setups, from Müller (2017).

6. Improving the Physiological Behavior

Table 6.4.: Overview of the results from the fifth heart beat (last beat) for all four simulations, from Müller (2017).

	DoubleHillMitLAT	DoubleHillOhneLAT	LumensMitLAT	LumensOhneLAT
LV-Vol Max	350,99	366,32	348,02	356,49
LV-Vol Min	276,55	289,34	292,27	296,61
LV EjecVol	74,44	76,98	55,75	59,88
LV EjecFrac	21,21%	21,01%	16,02%	16,80%
RV-Vol Max	186,59	193,76	269,55	273,96
RV-Vol Min	126,68	127,95	224,34	223,95
RV EjecVol	59,91	65,81	45,21	50,01
RV EjecFrac	32,11%	33,96%	16,77%	18,25%
Tot-Vol Max	537,59	560,08	617,57	630,46
Tot-Vol Min	403,23	417,29	516,61	520,56
Tot EjecVol	134,36	142,79	100,96	109,89
Tot EjecFrac	24,99%	25,49%	16,65%	17,43%
LV-Press Max	130,86	133,97	95,27	100,69
LV-Press Min	2,46	2,69	2,87	2,92
RV-Press Max	23,25	23,96	19,25	19,80
RV-Press Min	-0,18	-0,19	0,73	0,71
LA-Press Max	9,79	9,29	8,59	8,06
LA-Press Min	1,16	1,53	0,80	0,69
RA-Press Max	4,02	3,66	3,72	3,33
RA-Press Min	-1,70	-1,67	-1,47	-1,47

models (DoubleHill: 2.6ml, Lumens: 4.1ml for LVEjecVol). The only exception is an increase in the minimal volume of the right ventricle with Lumens model, that increases slightly from 224.0 ml to 224.3 ml. With inhomogeneous activation, the ventricular peak pressures decrease in both tension models, by 3.1 / 0.7 mmHg (DoubleHill LV/RV) and 5.4 / 0.5 mmHg (Lumens LV/RV). The peak pressures in the atria consistently increase slightly, but by at most 0.5 mmHg, which is observed in the LA for both tension models. The simulations with DoubleHill show in the left ventricle significantly higher peak pressures and ejection volumes than the simulations with Lumens. Although the Lumens model leads to smaller ejection volumes and smaller peak pressures in both ventricles, it can be observed that the absolute volume of the right ventricle is significantly larger than with the DoubleHill model. Changing both tension models to inhomogeneous activation times lets the LV peak pressures (LV-PressMax) reduce by 2.321 % and 5.383 % and LV ejection volume (LVEjecVol) lowers by 3.300 % and 6.897 %, for DoubleHill and Lumens tension model respectively. The differences of using Lumens instead of DoubleHill as tension model are larger, and consist of a reduction of LV-PressMax and LVEjecVol by 27.197 % and 25.107 % in the simulation with activation times, and by 24.841 % and 22.214 % in the simulations without.

6.2.5. Discussion

The phenomenologically modeled, stretch-dependent Lumens model is able to reflect the Starling effect on tension development on cell level. The cells with a high pre-stretch were the ones that developed a high amount of peak tension and on the other hand cells with a low pre-stretch were not able to develop a considerable amount of tension. This can be observed even without solving additional systems of differential equation that describe electrical potentials or ion concentrations inside biological cells. An inhomogeneous distribution of pre-stretch leads to an inhomogeneous pattern of developed tension maximum, and more pre-stretch means more developed tension. Due to the link between stretch and developed tension, the local stretch values itself remained in all cells in the physiological range between 0.7 and 1.3 throughout the whole heart beat, which is not the case for DoubleHill tension model.

Respecting local activation times, inhomogeneous activation lead with the DoubleHill tension model to a jellyfish-like motion of the left ventricle, with the contraction visibly starting near the apex and then moving upwards towards the valve plane. In combination with the circulatory system, this motion generated lower peak pressure and ejection volumes that are reached in both ventricles at a later time than with homogeneous case. Furthermore, that jellyfish-like motion possibly affects the opening diameter of the AV and aortic valves and might facilitate blood flow from atria to the ventricles as well as outflow to the aorta in a circulatory system model that respects e. g. aortic valves with varying diameter (which is not the case for most of them).

With the Lumens tension model in contrast, homogeneous or inhomogeneous activation showed not much difference in the motion pattern. However, the patterns from the inhomogeneous activation resembled the ones from 50ms earlier more than from the same time step and both ventricles reached their peak pressure at a later time than with homogeneous case. By the modeling of the activation time map, a ventricular cell was activated on average roughly 60ms later than in the homogeneous case, which could explain differences in pressure and volume curves. The jellyfish-like motion observed with the DoubleHill model was not present with the Lumens model at all, even with the inhomogeneous activation, and an explanation is a smoothing phenomenon by the length-dependency: The natural interaction between neighboring cells lets neighboring cells which are not activated yet already experience a higher passive pre-stretch which results in more and faster developing tension. Overall, homogeneous activation resulted in a higher pressure difference and output volume in both ventricles for both tension models with a slightly more pronounced effect for the Lumens model.

In the simulations with the DoubleHill tension model, each cell develops exactly the same force over time. The heart consumes in both simulations exactly the same amount of energy, making DoubleHill ideal for the evaluation of mechanically inhomogeneous contraction. DoubleHill shows with homogeneous activation higher peak pressures and ejection volumes in both ventricles than with the inhomogeneous activation. Under the assumption of the same amount of consumed energy, this translates to a higher efficiency of the heart with homogeneous activation. Besides the local delay, tension model parameters between the two setups are absolutely the same.

6. Improving the Physiological Behavior

The number of five simulated heart beats turned out to be sufficient for evaluating the pressure and volume curves. For all four setups, the pV loops of the fifth beat reached a periodical quasi-steady state and were identical to the ones from the fourth beat by visual comparison.

Areas with larger pre-stretch develop significantly higher tension than areas with low to medium pre-stretch, which emphasizes the importance of a reasonable pre-stretch for a sufficient tension development with the Lumens model. The DoubleHill tension model has no length-dependency and behaves indifferent. Pre-stretch however can not be directly generated, in the simulations it rather is a result of the initial inflation instead.

The observed ventricular end-diastolic volumes were in all four simulations larger than the patient ones due to the model initialization by inflation and resulted in rather high absolute volume changes. The patients volumes are the ones present before initialization (see Figure 6.14), since the geometry was segmented from end-diastolic state at which pre-stretch already is present. Since all four setups experienced the exact same initialization, the qualitative conclusion/outcome of this study does not change. Moreover, the thereby obtained homogeneously large pre-stretch was essential for the Lumens model to develop a considerable amount of tension (despite still lower than with DoubleHill), a point at which all other of the available unloading procedures from Section 3.10 failed. A general – not necessarily only tension model related – problem of the current state of this heart simulation model is that ventricular pressures have a tendency to be on the rather small side. The one single reason for that could not yet be found and might be a combination of segmentation error, wall thickness, fiber orientation, material parameters, boundary conditions and modeling of the hearts surroundings. All of the currently known unloading procedures listed in Section 3.10 however failed to produce a homogeneous large pre-stretch and contained even cells with negative pre-stretch, which is a clear no-go in combination with the Lumens tension model (for DoubleHill this does not matter).

The use of findings from other chapters for material parameters (Section 6.1) and circulatory system model (Section 6.3) lead to the benefits of fitting ventricular ejection volumes, a large V_8/V_0 ratio and a good fit to the Klotz curve.

The DoubleHill simulations have larger overall pressures in the left ventricle than the simulations with Lumens. This is not an indication of superiority of the DoubleHill model, in fact the overall developed tension and resulting compliance can be easily reduced while maintaining the same overall shape of the compliance curve via the single parameter T_{max} . The DoubleHill model had a larger area covered by the pV loop, denoting it produced more pumping energy. The larger overall values seen with the DoubleHill model might indicate that chosen scaling factors T_{max} of 60 kPa (left) and 80 kPa (right) might be slightly too large for direct comparability between the results of Lumens and DoubleHill. This makes the simulations with DoubleHill model hard to compare to the ones with Lumens model, since the amount of force put into the system by the two models is different. Even when restricting to one model, the results between homogeneous and inhomogeneous case are difficult to compare, since the inhomogeneous case has a larger “effective AV delay” averaged over all cells and the area near the valve plane contracts significantly later. In fact, a different AV delay alone could already explain the observed differences in pressure, volume and efficiency, leaving conclusions drawn from

6. Improving the Physiological Behavior

homogeneous vs inhomogeneous doubtful until more about that influence is known. In sum, this makes a reasonable comparison between the four simulations questionable.

Simulations of better comparability would require to take the simulation with Lumens and tune DoubleHill via T_{max} until both show the same ‘produced pump energy’ (i. e. area covered by pV loop). Regarding simulations with and without inhomogeneous activation, the inhomogeneous cases need to start activation earlier until both show the same AV delay at least in average. Otherwise, the observed effects might very well be caused by the different AV delays alone.

The mechanisms behind DoubleHill and Lumens are different, and so differ the possibilities to influence the tension development. Simply increasing tension scaling of the Lumens model – whose parameters were directly chosen from a publication – does not have the intended effect of producing more tension. The morphology of the tension over time for a single cell changes, with longer peak duration and shorter increase and decrease time. Peak tension does not increase, instead tension develops faster, the shape of tension over time steepens and overall chamber compliance alters. Chamber compliance however is important to fit for a circulatory system model and easily leads to failing simulations otherwise. DoubleHill has a prescribed “normalized tension” curve and reacts directly to modification of T_{max} by generating more tension while maintaining the shape of the tension curve and overall compliance time course. The Lumens model in contrast has a prescribed curve describing the “rise of mechanical activation” $F_{rise}(t)$, which is used together with current sarcomere, fiber length and other parameters to dynamically compute a value C for the “mechanical level of activation” (Lumens et al., 2009). Modifying the scaling factor σ_{act} in the Lumens model (equivalent to T_{max} in DoubleHill) however has a different effect. The shape of the tension curve changes, the peak is reached earlier and holds on for a longer time duration, which changes the overall compliance time course of the ventricles, but not the ejection volume. Instead, the Lumens model needs more stretch to generate a larger output. Values of 1.3 however should not be exceeded, so the only option is to increase pre-stretch in low stretch regions like near the apex. Lumens therefore needs a homogeneously large pre-stretch in order to compare results obtained on a segmentation-based geometry and draw the conclusion that using length-dependency leads to generally better or worse results.

6.2.6. Conclusions and Outlook

In this study, simulations were performed to compare the effects of switching from a stretch-independent tension model to a stretch dependent one and the effect of using individual activation times in the ventricles instead of one global value for each chamber.

The results showed no clear winning approach, neither a stretch-dependent tension model, nor the inclusion of activation times necessarily lead to a more efficient function heart. DoubleHill is good for cases in which a good overall compliance is important e. g. when coupling and evaluating a circulatory system or pericardium model. Lumens is good for use cases where local stretch values are important, for questions like “Which is the area with the largest stretch or the most load on the tissue?”, since these values permanently stay in physiological range.

6. Improving the Physiological Behavior

At the current state, a tension model without stretch-dependency still is a viable choice for a large number of applications on global level. Double Hill produced higher ejection volumes and peak pressures, and allows a direct control over the chamber compliance curve (that is a simple task to measure in reality). The compliance-follows-tension principle makes it easy to adjust shape and amplitude of the normalized tension to generate a specific (measured) compliance suitable as input for coupled models (e. g. to circulatory system, pericardium modeling). The only factor unreliable for evaluation is local stretch, that might still be out of physiological range due the stretch-unawareness of the DoubleHill model.

Switching from a simple function based model to a length-dependent model is not a straight forward step. Indeed, the Lumens model in its current state resulted in significantly lower ejection volume and peak pressure than the DoubleHill model and needs reasonable pre-stretch to develop tension at all. The Lumens model however showed some potentially interesting peculiarities that are not possible with the DoubleHill model. First, an inhomogeneous activation had almost no effect on the patterns of contraction and peak tension, such that length-dependency can be interpreted as a mechanism to revert the electrically necessary delay of activation. Second, a strong peak-tension-follows-initial-stretch principle was observed, making the method of unloading decisive for the applicability of stretch-dependent models. It turned out that the pattern of developed tension maximum follows the distribution of pre-stretch and more leads to more, which is a resemblance of the Starling effect on organ level. Simulations with the Lumens model even showed a smoothing effect of the stretch-dependency, the simulation with inhomogeneous activation lead to a visually very similar motion pattern as the ones with homogeneous activation, but only with lower key values. Physiologically, Lumens was able to keep the local stretch values in the physiological range, however global values like volume change and peak pressures were significantly smaller than with the DoubleHill model. Areas with small pre-stretch were unable to develop a significant amount of active tension and thereby could not contribute to the pumping function of the heart. The Lumens model has stricter requirements than the DoubleHill model to be able to generate reasonable tension at all. Main requirement of the Lumens model thereby is a sufficient pre-stretch that is present in a large number of cells. The pre-stretch however still needs to stay in physiological range of at most 1.3 and low-pre-stretch-regions reduce the efficiency drastically. Even small areas of low pre-stretch impact the global mechanical functioning of the heart and impact simulation results, the heart does not produce enough pressure. Due to the stretch-dependency, the initial pre-stretch and thereby the method used for unloading becomes a deciding factor for the applicability of the Lumens tension model. Unfortunately, this is difficult to generate for segmentation based patient geometries, even by a sophisticated unloading method (other than simple inflation, which increases the original geometry volumes) from Section 3.10. With the amount of pre-stretch and its spatial distribution (overall homogeneously large pre-stretch), an important prerequisite for the Lumens model could be identified that might serve as quality measure for the evaluation of an unloading method.

Regarding activation times, both tension models showed larger pressure and volume changes with simultaneous activation of ventricular cells, which can be interpreted as a

6. Improving the Physiological Behavior

sign of a higher efficiency or better utilization of available forces. With inhomogeneous activation times, all key values were lower compared to the simulations with global activation. Interestingly, it is known that there is a delay present on the electrical level, which is not necessarily the case on mechanical level. With the length-dependent tension model, an inhomogeneous start of the contraction had no visual effect on mechanical motion and tension patterns, only on pressure and volume amplitude. With the stretch-independent model, the motion pattern did change and resembled a motion similar to a moving jellyfish. The impact on pressure and volume amplitudes was slightly higher than with the length-dependent tension model. In fact, electro-physiological peculiarities in the human heart like the fast-leading Purkinje tree try to reduce the overall activation time of the ventricles. Overall, the effect of local activation times was rather limited and smaller than the difference between two different tension models.

To restate the first hypothesis, local activation times are definitely not needed for an efficient mechanical motion of the heart. While a locally delayed activation is a valid and proven assumption on the electrical level, a smoothing effect makes this almost impossible to measure on the mechanical level. Moreover, the simulation results in this study clearly indicate that a homogeneous activation on the mechanical level is advantageous for the heart's efficiency, and addition of local activation times decreases the efficiency of the heart. To restate the second hypothesis from the introductory section, the length-dependent tension model is capable of smoothing out inhomogeneities due to local activation time. Partially thinner walls are effectively weaker and are exposed to a larger passive deformation, which results in a higher developed tension. Regions with lower initial stretch develop a low amount of tension. Differences due to inhomogeneous pre-stretch however stayed present and that pattern recurred in the maximum developed tension as well. To restate the third hypothesis, the Starling effect on cell level was observed to lead to a Starling effect on organ level. The length-dependent Lumens model produced a larger peak tension with an increase of initial stretch. Using the geometry without any preloading mechanism, all cells start at an initial stretch of $\lambda_x = 1$, which develops only a negligible amount of tension not sufficient to produce any motion of the organ at all. To restate the fourth hypothesis from the introduction, length dependency is more valuable than respecting inhomogeneous local activation times. The addition of locally inhomogeneous activation times – while being correct and present on the electrical level – turned out to be rather disadvantageous for tension development and mechanical functioning of the heart and might not necessarily be present on the mechanical level due to the second hypothesis. Using a length-dependent tension model however succeeded in showing a number of qualitative effects like the Starling effect (larger stretch leads to more tension), the maintaining of local fiber strain values to physiological limits and the aforementioned smoothing effect of inhomogeneous activation.

As future steps, a number of improvements to the current study are possible. In order to improve the comparability between the four setups, additional measures could be introduced that allow tuning towards a similar overall energy consumption. Assuming a cell time consumes more energy when it produces a higher amount of tension or the same amount over a longer period of time, a space-time integrated force over one heartbeat would allow a better relative “efficiency-rating” by the amount of force put in the system

6. Improving the Physiological Behavior

rather than simply comparing peak pressures or volumes changes. For that, reducing DoubleHill scaling factor (T_{max}) could produce a comparable “energy consumption” between the two tension models and make peak pressures better comparable. An obvious improvement is the use of a different, inverse unloading algorithm that allows to generate pre-stretch while maintaining to get chamber volumes and end-diastolic shape close to the patient. Main challenges for a good functioning of the Lumens tension model thereby are the generation of homogeneously large pre-stretch within physiological limits, a point where even the simple inflation approach massively struggled. Another possible improvement might be the image acquisition of the end-systolic instead of end-diastolic state. This contracted state is closer to the actual pressure-free state and naturally provides end-diastolic volumes closer to the patient ones while circumventing the need for an inverse unloading. Further investigation of the delay between mechanical (not electrical) contraction of atria and ventricles is needed. The delay between atrial contraction and ventricular contraction in the case of inhomogeneous activation times is not fully understood. It is unclear to what extent the lowering of ejection volumes and pressure peaks might be reproduced with homogeneous activation pattern and 50 ms additional delay. In that case, a comparison of AV delays with respect to “average activation time” instead of earliest activation might serve well. Moreover, the mentioned compliance curve should be evaluated to see how suited the Lumens model is for applications where an accurate compliance is needed. Finally, a better initialization or unloading method needs to be found that reduces low stretch regions and increases the pre-stretch homogeneously to the range between 1.2 and 1.3 in order to improve the applicability of the Lumens tension model.

6.3. Application III – A Fully Coupled Circulatory System

To fulfill its main purpose of sustain metabolism, the human heart needs to pump blood through a human’s body arterial and venous systems. A model of the circulatory system has two main purposes. First, it has the numerical purpose of providing time-varying boundary conditions on the endocardial surfaces of each of the four compartments, pressures on the valve planes and pre- and afterloads values at the orifices. Second, it allows evaluation of diagnostic values often used in clinical applications: end-systolic and end-diastolic pressure values, ejection volumes under “real load” (i. e. contraction against a resistance) and furthermore time-continuous data of the four chambers like pressure and volume over time as well as flow rates through the open valves. For a credible solution, the circulatory system should conserve blood volume and be able to reach a quasi-static periodic state by computing multiple heart beats in a fully coupled model, even when ignoring mass inertia of the myocardial walls.

Aim of this project is to develop a closed-loop model of the human vascular system that allows to run multiple heart beats and couple it to the existing finite element framework for solving the elasticity equation. The model should moreover reflect the conservation of flow between adjacent chambers, conservation of total blood volume and provide equal ejection volumes of both ventricles in quasi-periodic static state after computing multiple

beats. A closed-loop model is needed for blood conservation and equal ejection volume of the two ventricles in contrast to open-loop models that are most applicable to provide simple boundary conditions during a specific phase of the cardiac cycle.

This chapter presents the hierarchical development of a lumped parameter closed-loop circulatory system models (one ventricle, two ventricles, four chambers) that are subsequently applied to the biomechanics model in a fully-coupled manner. Different from the circulatory system model used by Fritz et al. (2014), the implementation of this framework uses pressure-difference-dependent valve states that allow seamless computation of multiple heart beats. The second difference is that the strong coupling is realized in an iterative manner instead of a monolithic approach with the idea in mind to allow easier experimentation with preconditioners and solvers specialized to biomechanical problems. The investigations in this section were conducted before the previous two sections and thereby do not contain the findings for passive and active material properties from Section 6.1 and 6.2 yet. Instead, their modeling is the same as in Fritz et al. (2014).

6.3.1. Closed Circulation Modeling

Circulatory systems can be described via the analogy between hydraulic networks and electrical networks. Resistances and elasticity of blood vessels correspond to electrical resistances and capacitors, pressure gradients to voltage differences, and flow rates to electrical current. The arising differential equations are the same and allow to apply schematics and solution strategies borrowed from electrical engineering.

The hierarchy of circulatory system models was designed in a student supervision of Steffen Schuler's master thesis (Schuler, 2016) by the author of this thesis. It is greatly inspired by the publication from Lumens et al. (2009) about modeling a circulatory system including the four heart chambers as representative electrical circuit and Kerckhoffs et al. (2007b) for the coupling part for a two-chamber FEM model.

Circulation with One Ventricle

First, a circulatory system model "CircOneVentricle" was created that contains one ventricle with its outflow connected to its input (Figure 6.22). Despite this is not a physiological case, it however serves well as a simple model for development of the coupling algorithm to elasticity simulation and acts as a base implementation for the more complicated circuits.

The ventricle is given by a time-varying capacitance to model the active part, and diodes represent the unidirectional property of AV (mitral/tricuspid) and arterial valve (aortic/pulmonary). The arterial system is modeled as three-element Windkessel with arterial compliance C_{Art} , arterial impedance R_{Art} , and peripheral resistance R_{Per} . The venous system is given by compliance C_{Ven} . R_{Ven} closes the system by connecting arterial output with ventricular input, and a separate resistance for the arterial valve $R_{ArtValve}$ allows to get the arterial pressure P_{Art} , while the AV valve resistance is included in the peripheral resistance R_{Per} .

6. Improving the Physiological Behavior

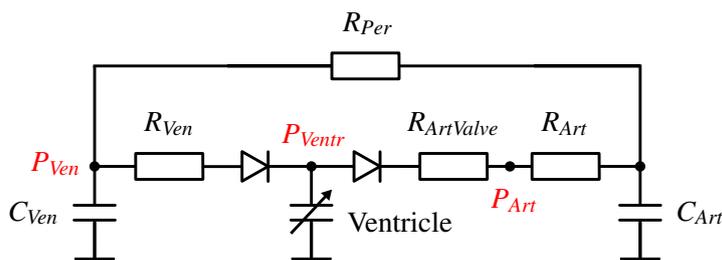


Figure 6.22.: Schematic of the circulation model “CircOneVentricle”, from Schuler (2016).

The governing equations of the “CircOneVentricle” model are given as follows:

Algebraic equations (one ventricle)

$$\begin{aligned}
 P_{CArt} &= \frac{V_{Art}}{C_{Art}} \\
 Q_{Art} &= \max\left\{\frac{P_{Ventr} - P_{CArt}}{R_{ArtValve} + R_{Art}}, 0\right\} \\
 P_{Art} &= P_{Ventr} - R_{ArtValve} \cdot Q_{Art} \\
 P_{Ven} &= \frac{V_{Ven}}{C_{Ven}} \\
 Q_{Per} &= \frac{P_{CArt} - P_{Ven}}{R_{Per}} \\
 Q_{Ven} &= \max\left\{\frac{P_{Ven} - P_{Ventr}}{R_{Ven}}, 0\right\}
 \end{aligned}$$

Differential equations (one ventricle)

$$\begin{aligned}
 \dot{V}_{Ventr} &= Q_{Ven} - Q_{Art} \\
 \dot{V}_{Art} &= Q_{Art} - Q_{Per} \\
 \dot{V}_{Ven} &= Q_{Per} - Q_{Ven}
 \end{aligned}$$

Instead of solving each differential equation separately, a vector-valued state variable object \mathbf{u} with $\dot{\mathbf{u}} = [\dot{V}_{Ventr}, \dot{V}_{Art}, \dot{V}_{Ven}]^T$ is used.

Circulation with Two Ventricles

The second model “CircTwoVentricles” contains both ventricles by adding a second time-varying elastance for the right ventricle, using “CircOneVentricle” twice (Figure 6.23). It is very similar to the model by Santamore and Burkhoff (1991) depicted in Figure 6.24, which is probably the first Windkessel-based closed-loop model of the circulatory system.

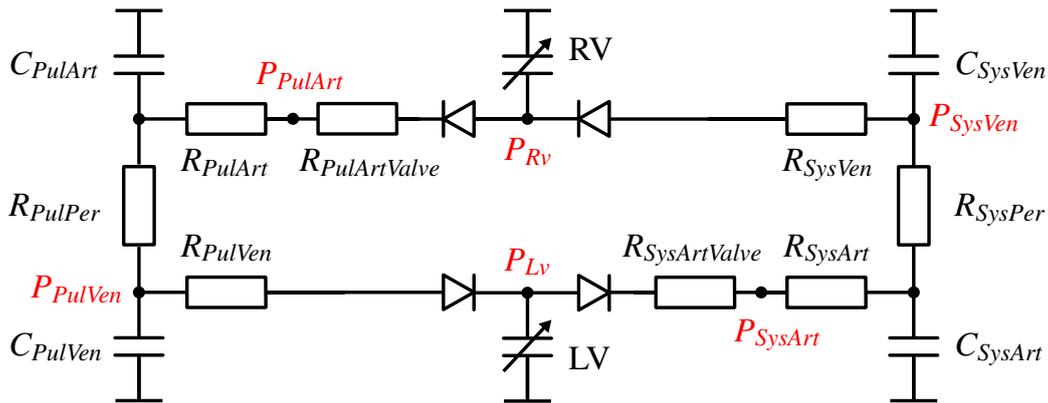


Figure 6.23.: Schematic of the circulation model “CircTwoVentricles”, from Schuler (2016).

As given by physiology, the output of the left ventricle is connected via the systemic circulation to the input of the right ventricle (right half of the circuit), and vice versa via the pulmonary circulation (left half of the circuit).

Each ventricle gives its active contribution to the system via a variable capacitor, that represents the varying elastance usually created by contraction. Both, systemic circulation and pulmonary circulation are given by a standard three-element Windkessel model (R_c, C_a, R_a) connected to venous resistance and compliance (R_v, C_v), respectively. Different from Santamore and Burkhoff (1991), this model includes valve resistances $R_{PulArtValve}$ and $R_{SysArtValve}$. Parameters for the human circulation are available by Burkhoff and Tyberg (1993).

Circulation with the Whole Heart

The “CircWholeHeart” model is an extension of “CircTwoVentricles” to include both atria for a complete model of the circulatory system (Figure 6.25).

Both atria as well as both ventricles are now given as time-varying elastance, with separate resistances $R_{LavValve}$ and $R_{RavValve}$ for the two AV valves. R_{SysVen} and R_{PulVen} thereby become characteristic venous resistances, creating own three-element Windkessel models of the venous systems together with peripheral resistance and venous compliance as in Wang et al. (2006). Physiologically, there are no valves present between atria and the venous systems.

Governing equations of the “CircWholeHeart” model read as follows:

6. Improving the Physiological Behavior

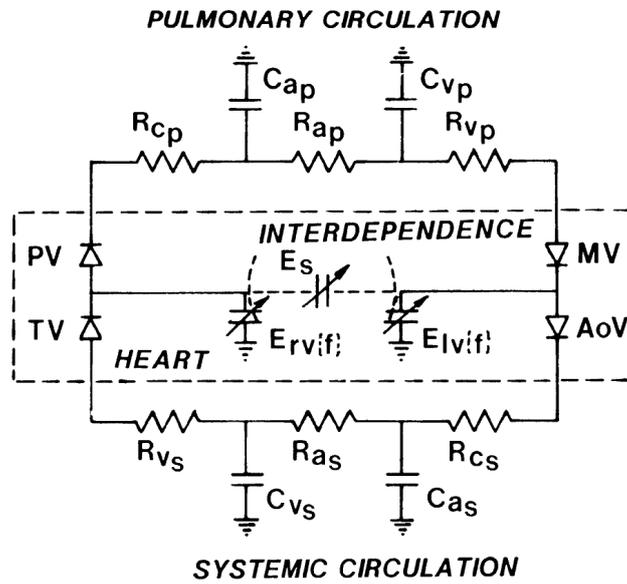


Figure 6.24.: Schematic of the circulation model from Santamore and Burkhoff (1991).

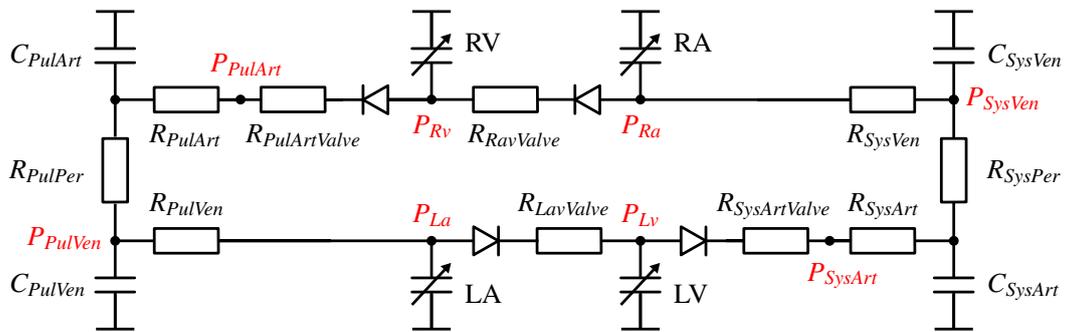


Figure 6.25.: Schematic of the circulation model "CircWholeHeart", from Schuler (2016).

6. Improving the Physiological Behavior

Algebraic equations (whole heart)

$$\begin{aligned}
 P_{CSysArt} &= \frac{V_{SysArt}}{C_{SysArt}} \\
 Q_{SysArt} &= \max\left\{\frac{P_{Lv} - P_{CSysArt}}{R_{SysArtValve} + R_{SysArt}}, 0\right\} \\
 P_{SysArt} &= P_{Lv} - R_{SysArtValve} \cdot Q_{SysArtValve} \\
 P_{SysVen} &= \frac{V_{SysVen}}{C_{SysVen}} \\
 Q_{SysPer} &= \frac{P_{CSysArt} - P_{SysVen}}{R_{SysPer}} \\
 Q_{SysVen} &= \frac{P_{SysVen} - P_{Ra}}{R_{SysVen}} \\
 Q_{Rav} &= \max\left\{\frac{P_{Ra} - P_{Rv}}{R_{RavValve}}, 0\right\} \\
 P_{CPulArt} &= \frac{V_{PulArt}}{C_{PulArt}} \\
 Q_{PulArt} &= \max\left\{\frac{P_{Rv} - P_{CPulArt}}{R_{PulArtValve} + R_{PulArt}}, 0\right\} \\
 P_{PulArt} &= P_{Rv} - R_{PulArtValve} \cdot Q_{PulArt} \\
 P_{PulVen} &= \frac{V_{PulVen}}{C_{PulVen}} \\
 Q_{PulPer} &= \frac{P_{CPulArt} - P_{PulVen}}{R_{PulPer}} \\
 Q_{PulVen} &= \frac{P_{PulVen} - P_{La}}{R_{PulVen}} \\
 Q_{Lav} &= \max\left\{\frac{P_{La} - P_{Lv}}{R_{LavValve}}, 0\right\}
 \end{aligned}$$

Differential equations (whole heart)

$$\begin{aligned}
 \dot{V}_{Lv} &= Q_{Lav} - Q_{SysArt} \\
 \dot{V}_{SysArt} &= Q_{SysArt} - Q_{SysPer} \\
 \dot{V}_{SysVen} &= Q_{SysPer} - Q_{SysVen} \\
 \dot{V}_{Ra} &= Q_{SysVen} - Q_{Rav} \\
 \dot{V}_{Rv} &= Q_{Rav} - Q_{PulArt} \\
 \dot{V}_{PulArt} &= Q_{PulArt} - Q_{PulPer} \\
 \dot{V}_{PulVen} &= Q_{PulPer} - Q_{PulVen} \\
 \dot{V}_{La} &= Q_{PulVen} - Q_{Lav}
 \end{aligned}$$

6. Improving the Physiological Behavior

$$\dot{\mathbf{u}} = [\dot{V}_{Lv}, \dot{V}_{SysArt}, \dot{V}_{SysVen}, \dot{V}_{Ra}, \dot{V}_{Rv}, \dot{V}_{PulArt}, \dot{V}_{PulVen}, \dot{V}_{La}]^T$$

Time Integration

These systems of ordinary differential equations $\dot{\mathbf{u}} = \mathbf{g}(t, \mathbf{u}) = \mathbf{f}(\mathbf{p}_c(t), \mathbf{u})$ were initially solved using a standard explicit Runge-Kutta method of fourth order (RK4):

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \cdot \frac{1}{6}(\mathbf{k}_1 + 2\mathbf{k}_2 + 2\mathbf{k}_3 + \mathbf{k}_4)$$

Since the heart chambers are represented by variable compliances, the whole system $\mathbf{p}_c(t)$ depends explicitly on time. A coupling algorithm however yields chamber pressures $\mathbf{p}_{c,estim}^n$ and $\mathbf{p}_{c,estim}^{n+1}$ estimated by the FEM model only at whole numerical time steps, not at $t^n + \frac{\Delta t}{2}$. Values at the intermediate steps $\mathbf{k}_1, \dots, \mathbf{k}_4$ are not directly available from a coupling algorithm. Using only $p_c(t) \equiv \mathbf{p}_{c,estim}^n$ for $\mathbf{k}_1, \dots, \mathbf{k}_4$, the computed chamber pressures turned out to be susceptible to numerical oscillations. Instead, an improvement of numerical stability could be observed when calculating \mathbf{k}_4 from $\mathbf{p}_{c,estim}^{n+1}$ and the pressures needed for \mathbf{k}_2 and \mathbf{k}_3 via linear interpolation $\mathbf{p}_c(t^n + \frac{\Delta t}{2}) := \frac{1}{2}(\mathbf{p}_{c,estim}^n + \mathbf{p}_{c,estim}^{n+1})$. The oscillations further vanished completely when using only $\mathbf{p}_{c,estim}^{n+1}$ at all four intermediate rates $\mathbf{k}_1, \dots, \mathbf{k}_4$, therefore this is the algorithm of choice. While this modification makes the algorithm more robust it negatively impacts the accuracy of the time integration.

For comparison of accuracy, a ground-truth was generated with RK4 and an explicitly prescribed varying chamber elastance that can be written directly into the equations, instead of one estimated from FEM model. The results for two beats are shown in Figure 6.26 together with a similarly modified explicit Euler method ($\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \cdot f(\mathbf{p}_{c,estim}^{n+1}, \mathbf{u}^n)$) as well. Additionally, a smaller time step vastly increases accuracy as well. As the computational costs of the circulatory system are small compared to the number of unknowns of the FEM model, a separate maximal step length for the circulatory system was implemented, that allows to run the circulatory system independently at a smaller step size (usually ≤ 0.1 ms) than the time integration of the FEM model. One step computed by the FEM solver gets then complemented by multiple smaller time integration steps of the circulatory system.

Parameters for the Circulation Model (Whole Heart)

Despite being modeled by simple components, the ‘‘CircWholeHeart’’ already depends on a total of 19 model parameters (10 resistances, 4 capacitances, 4 unstressed volumes and the total blood volume). Credibility of its results is only as good as the choice of the used parameters. The mesh of the FEM model was generated from a 28 year old healthy male, which was used to select appropriate parameters from literature (assuming 75 kg body weight, $1.9 m^2$ body surface area). Being constructed by Windkessel models for the single parts of the circulation, data can be used from measurements representing only a part of the circulation (only systemic or pulmonary Windkessel) as well. Table 6.5 lists models sharing some of their structure, together with the respective parameters,

6. Improving the Physiological Behavior

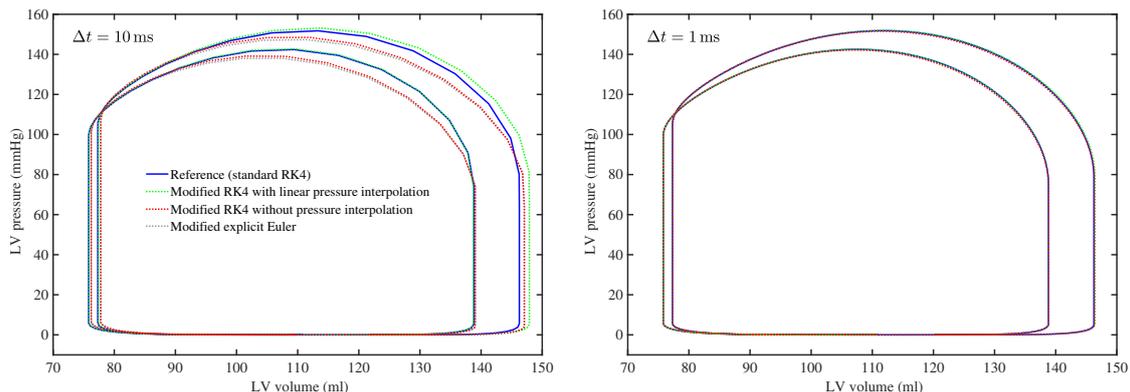


Figure 6.26.: Comparison of different time integration schemes for the circulatory system, adapted from Schuler (2016).

and finally the parameters chosen for the simulations with the “CircWholeHeart” model, denoted as “baseline parameters”.

Additionally, all parameters were modified by 50% around the baseline parameters to check the model for its sensitivity with respect to the single parameters (Figure 6.27).

Solving the Coupled Problem

Besides the direct interaction of neighboring chambers through the motion of common walls, there is also indirect interaction of the volumes through exchange of blood via circulatory system. Interaction between circulatory system and the solver framework was realized as iterative strong coupling (see Section 4.3). Thereby, changes in cavity volumes of the FEM model are guaranteed to fit the time integrated flow rates of the circulation model $|V_{FEM} - V_{circ}| < \varepsilon$. This is achieved by computing a specific time interval multiple times with different pressures, until the volumetric residuum between the two systems is minimal and pressures and volumes at the end of the interval are in equilibrium:

$$\min_p r(p) \quad \text{with} \quad r(p) := V_{FEM}(p) - V_{circ}(p)$$

where V_{FEM} and V_{circ} are n -dimensional vectors containing computed chamber volumes corresponding to specific chamber pressures p for each of the n heart chambers by solving the respective elasticity FEM systems. The initial pressures were determined by an Adams-Bashforth extrapolation of the previous five solutions in time. Coupling (or state of convergence) was defined to be achieved at a cavity volume error of less than 1 ml between volume from one step FEM simulation with prescribed pressure and predicted volume from circulatory system.

The coupling algorithm is an extension of Newton’s method described by Kerckhoffs et al. (2007b) for a bi-ventricular 3d finite-element model, where the atria were represented

6. Improving the Physiological Behavior

Table 6.5.: Parameters of circulatory system models that share some of their structure with the “CircWholeHeart” model and the parameters chosen as “baseline” parameter set. Units are R in $mmHg \cdot s \cdot ml^{-1}$, C in $ml \cdot mmHg^{-1}$ and V in ml . From Schuler (2016).

	[Santamore & Burkhoff, 1991] 20 kg dog	[Burkhoff & Tyberg, 1993] 70-75 kg healthy man, 1.9 m ²	[Smith et al., 2007] Healthy human	[Hann et al., 2010] Healthy human	[Kerckhoffs et al., 2007] 20 kg dog	[Aguado-Sierra et al., 2011] 65 year old male with severe cardiac diseases	[Bovendeerd et al., 2009] Human	Calculated from [Arts et al., 2012] Normal human at rest	Chosen baseline values
$R_{SysArtValve}$	–	–	0.0105	0.018	–	–	–	–	0.006
R_{SysArt}	0.2	0.03	–	–	0.0525	0.0008	0.075	0.0882	0.07
C_{SysArt}	0.4	1.32	1.42	1.45	4.35	0.67	3.33	1.42	2
$V_{SysArtUnstr}$	–	?	800	–	–	–	500	–	800
R_{SysPer}	2.5	0.9	1.05	1.0889	1.8527	0.6	0.9	–	0.9
R_{SysVen}	0.015	0.015	–	–	0.3855	0.045	0.0375	0.0108	0.03
C_{SysVen}	17	70	88.88	169.49	57.73	46.66	80	91.25	100
$V_{SysVenUnstr}$	–	?	2830	–	–	–	3000	–	2850
$R_{RavValve}$	–	–	0.0014	0.0237	0.0038	–	–	–	0.003
$R_{PulArtValve}$	–	–	0.0036	0.0055	–	–	–	–	0.003
R_{PulArt}	0.06	0.02	–	–	0.03	0.0015	–	0.0367	0.02
C_{PulArt}	2	13	2.96	2.71	5.56	2	–	2.36	10
$V_{PulArtUnstr}$	–	?	160	–	–	–	–	–	150
R_{PulPer}	0.3	0.03	0.1425	0.1552	0.0379	0.15	–	–	0.07
R_{PulVen}	0.015	0.015	–	–	0.0379	0.045	–	0.0196	0.03
C_{PulVen}	3	8	166.65	136.99	6.67	2.67	–	7.87	15
$V_{PulVenUnstr}$	–	?	200	–	–	–	–	–	200
$R_{LavValve}$	–	–	0.0005	0.0158	0.0038	–	–	–	0.003
V_{Total}	250	5500	5500	1500	1722.3	?	5000	–	5500
$V_{TotalStr}$	250	750	1510	1500	1722.3	?	1500	–	1500
$V_{TotalUnstr}$	0	4750	3990	0	0	?	3500	–	4000
$V_{TotalCirc}$	175	5334	–	1276	1569.9	–	–	–	–

6. Improving the Physiological Behavior

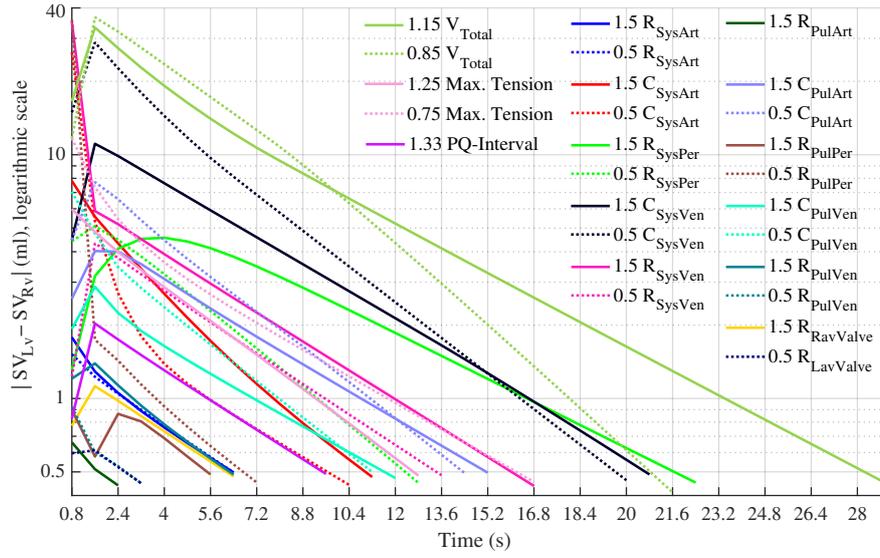


Figure 6.27.: Sensitivity of the “CircWholeHeart” model with baseline parameter set to perturbation of each parameter by 50% by means of needed simulated time until reaching converged state is again. From Schuler (2016).

in the circulatory system model as simple time-varying elastance instead of an elastance based on the FEM representation.

The Newton algorithm successively computes pressure guesses p_i and Jacobians $C_i = \frac{dr(p_i)}{dp}$ by pressure perturbation (finite differences) in order to find the root of dr/dp :

$$p_{i+1} = p_i - C_i^{-1}r(p_i) \quad \text{with} \quad C_i = \begin{pmatrix} \frac{\partial V_{Lv}}{\partial p_{Lv}} & \frac{\partial V_{Lv}}{\partial p_{Ra}} & \frac{\partial V_{Lv}}{\partial p_{Rv}} & \frac{\partial V_{Lv}}{\partial p_{La}} \\ \frac{\partial V_{Ra}}{\partial p_{Lv}} & \cdot & \cdot & \cdot \\ \frac{\partial V_{Rv}}{\partial p_{Lv}} & \cdot & \cdot & \cdot \\ \frac{\partial V_{La}}{\partial p_{Lv}} & \cdot & \cdot & \cdot \end{pmatrix}$$

Diagonal entries of the Jacobian reflect chamber compliances ($\Delta V/\Delta P$), off-diagonal entries interdependencies of the volumes. Four additional solutions of the FEM system have to be computed by separate pressure disturbance of each chamber before one Newton step can take place. In order to reduce the computational effort from computing the required expensive FEM solutions, Kerckhoffs et al. (2007b) proposed to use a quasi-Newton method instead, which updates only an approximation of the Jacobian in each iteration.

Due to the specific structure of the Jacobian

$$C = dV_{FEM}/dp - dV_{circ}/dp = C_{FEM} - C_{circ}$$

6. Improving the Physiological Behavior

an update-formula gives the possibility for performance improvements by updating only the circulatory system part C_{circ} of the Jacobian in each step and the FE part C_{FEM} less frequently, which can be done comparatively inexpensive compared to a full derivative \mathbf{C} . This allows to update the derivative with the cheap-to-compute circulatory part, while the expensive-to-compute mechanical part of the derivative is recomputed only once in a while, e.g. in the first coupling iteration or when a certain number of iterations is succeeded. Hence, the algorithm was modified to a Quasi-Newton method, first with a derivative approximation by the “good-Broyden” formula (Broyden, 1965):

$$C_i = C_{i-1} + (\Delta r_i - C_{i-1} \Delta p_i) \frac{\Delta p_i^T}{\Delta p_i^T \Delta p_i}$$

The Sherman-Morrison formula goes one step further and approximates the inverse of the pressure Jacobian C^{-1} directly and saves one matrix inversion by using only simple and inexpensive matrix operations (addition, multiplication). This improves the approximation of C^{-1} by only using the current pressure and information already computed in previous iterations:

$$C_i^{-1} = C_{i-1}^{-1} + (\Delta p_i - C_{i-1}^{-1} \Delta r_i) \frac{\Delta p_i^T C_{i-1}^{-1}}{\Delta p_i^T C_{i-1}^{-1} \Delta r_i}$$

In the very first time step, a full Jacobian needs to be computed by pressure perturbation. As the Jacobian usually does not change much between successive FEM time steps, it is often approximated already sufficiently accurate by the previous Jacobian plus update steps. However this does not always pay off, and a more reliable convergence could be obtained when the first iteration of each time step uses a full Jacobian with Newton, followed by quasi-Newton updates in the following iterations.

A comparison of this quasi-Newton method and a simple secant method by means of the needed number of iterations per step was done over two heart beats and is shown in Figure 6.28. The secant method took on average 51.86 iterations, while the quasi-Newton averaged at only 3.58 iterations. As the relaxation phase begins, the number of iterations increases drastically since the heart tissue is exposed to large accelerations due to opening mitral and tricuspid valves and rapid inflow of blood. The initial derivative approximation from previous time steps is at this point in time naturally a bad guess.

6.3.2. Integration in the Biomechanics Framework

For implementation of the circulatory system models, a new framework for circulation models was created. That framework inherits from `CBSolverPlugin` and integrates as iterative solver plugin in `CardioMechanics` (see Section 4.1). Figure 6.29 gives a simplified overview of the circulation model framework, its classes, data types, functions, and the relations between them. These are:

`CBCirculationCavity` represents exactly one heart chamber given by a closed surface mesh and provides functions to apply a specific pressure to it and to compute its current volume. `CBCircModel` reflects an abstract circulation model and implements time in-

6. Improving the Physiological Behavior

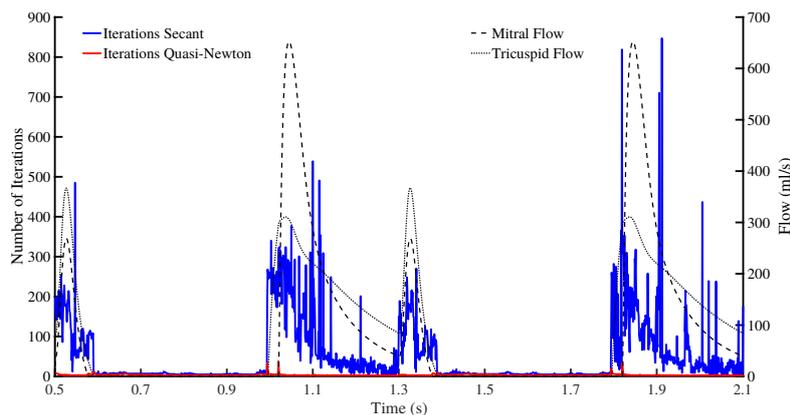


Figure 6.28.: Comparison of coupling iterations needed per time step between quasi-Newton and the secant method (T4 elements, without pericardium), from Schuler (2016).

tegration, check of the coupling condition and the steady-state check using an abstract `stateVars` object. Finally, `CBCirculation` holds the `CBCircModel` and a list of `CBCirculationCavity`, and it provides the interface functions for a `CBSolverPlugin` to the main program, namely the `Apply()` function that implements the general coupling algorithm and handles the necessary return codes. `CBCircOneVentricle`, `CBCircTwoVentricles` and `CBCircWholeHeart` all are realizations of `CBCircModel` and implement the respective differential equations in its `Algebraics()` function as well as a map between cavity indices and the respective heart chambers necessary for that model.

The coupling iterations are realized by a “REPEAT” return code, that advises the global solver part not to progress in time and instead recompute the same step again, without modifying any of the other possibly used solver plugins. A custom data type containing all state-variables was implemented as well as a copy operator to facilitate the necessary handling of progress and step back in time.

The iterative coupling algorithm implemented in `CBCirculation` solves the minimization problem of the coupling condition. By tracking previous states of the optimization variables (chamber pressures), the algorithm computes derivatives and new estimates in order to minimize the volume difference between circulatory system and FEM model of a specific time interval. Once the cavity volume error is small enough, `CBCirculation` sends a “SUCCESS” return code and the biomechanical framework is allowed to progress in time. Strong coupling of more than one iterative `CBSolverPlugin` is possible as well and can be realized without additional implementation efforts as depicted in Figure 4.5 using a nested loop in the main program.

A closed-surface check was implemented in initialization phase to detect logical errors in the input data that would otherwise lead to position-dependent and incorrect volumes. According to Gauss law, the cavity volume (volume integral with density 1) is computed

6. Improving the Physiological Behavior

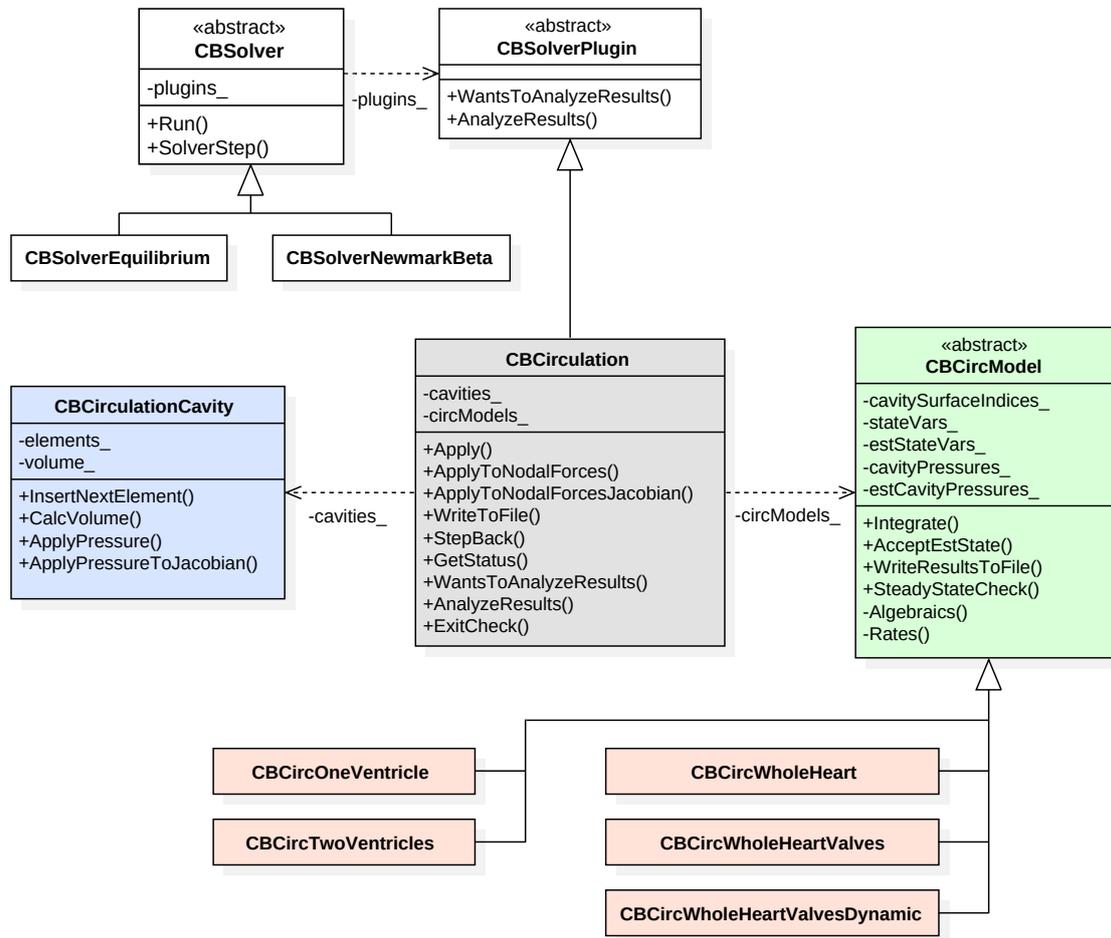


Figure 6.29.: Diagram showing the most important components of the circulation model framework, from Schuler (2016).

6. Improving the Physiological Behavior

as surface integral using all surface triangle normals:

$$\begin{aligned} \text{vol}(V) &= \int \int \int_V 1 \, dx \, dy \, dz = \int_V \nabla \cdot \frac{1}{3}(\mathbf{111})^T dV = \int_{\partial V} \frac{1}{3}(\mathbf{111})^T \cdot \vec{n} dS \\ &= \sum_{T \in \text{Triangles}} \frac{1}{3}(\mathbf{111})^T \cdot n_{T_k} = \sum_{T_k \in \text{Triangles}} \Delta V_{T_k} \end{aligned}$$

The last part sums up the volume parts spanned by the tetrahedron between an origin \vec{r} and the respective surface triangle points \vec{a} , \vec{b} and \vec{c} :

$$\Delta V_{T_k} = \frac{1}{6}(\vec{a} - r) \cdot ((b - a) \times (c - a))$$

The volume gets computed with two different origins and the surface is regarded as closed when the two values are the same.

A steady-state check was implemented that allows to stop the simulation once the blood distribution in the circulatory system does not change between cardiac cycles. For that, the stroke volume difference (SVD) between the systemic and pulmonary part of the circulation is added as another ODE to be integrated over time:

$$\dot{\text{SVD}} = Q_{\text{SysArt}} - Q_{\text{PulArt}}$$

Once the absolute SVD drops below a defined threshold of 0.5 ml for the whole duration of one cardiac cycle (usually 0.8 s), the steady-state is regarded as reached.

A useful application of the circulatory system is as a very simple mechanism to generate preload on the geometry. Therefore, a certain negative pressure gets applied to the chambers, which usually causes a shrinking of the geometry, followed by a reset of the reference coordinates and a subsequent inflation to reobtain approximately the original volumes. This method was used to generate preload for all simulations in this study.

6.3.3. Simulations and Results of the Coupled Model

This chapter presents results obtained from the newly developed circulatory system model coupled to the FEM model of the heart. Simulations were done on two different geometries and circulation models containing either only the left ventricle (with ‘‘CircOneVentricle’’ as circulation model), both ventricles (with ‘‘CircTwoVentricles’’), all four chambers without pericardium (with ‘‘CircWholeHeart’’), or all four chambers including pericardium (also with ‘‘CircWholeHeart’’, but adding the Fritz pericardium modeling). Additionally, a sensitivity analysis was done as guide for further parameter modification and estimation of certain cardiovascular illnesses.

The simulations use the same four-chamber geometry and passive tissue parameters for the heart and surrounding tissue as used by Fritz et al. (2014), extended with truncated representations of the aorta and pulmonary veins (parameters given in Table 6.6). A second mesh was generated with linear elements and without trunks or pericardium for less accurate but faster computation of the sensitivity analysis and development of the coupling algorithm. Pressures were applied only to the heart chambers included in

6. Improving the Physiological Behavior

Table 6.6.: Material parameters used in the simulations, adopted from Schuler (2016).

Guccione	C (Pa)	b_1	b_2	b_3	K (Pa)
Ventricles	1200	26.7	2.0	14.7	2×10^5

Mooney-Rivlin	C_1 (Pa)	C_2 (Pa)	B (Pa)
Atria	7450	0	2×10^5
Pericardium	1000	0	1×10^4
Chimneys	14900	0	2×10^5

the respective circulation model. The sensitivity analysis and all simulations without pericardium were conducted with first-order elements, to concentrate on the effects of the circulatory system. The simulation of all four chambers including pericardium uses elements of second order and the contact pericardium model by Fritz et al. (2014) together with the respective mesh. The force model however was changed in all simulations to a prescribed time-dependent function, given by the DoubleHill model as described in Section 6.2 with respective parameters for contracting atria and contracting ventricles. A length-dependent model was explicitly not chosen to prevent positive side effects that otherwise might reduce or even hide effects of a possibly malfunctioning circulatory system model or coupling algorithm (a larger chamber pressure would lead to more strain, a larger developed force and thereby still to a similar volume). The cycle length was set to 0.8 s, corresponding to 75 beats per minute, and all simulations were allowed to compute multiple beats until convergence. Unless stated otherwise, all simulations were evaluated in a converged quasi-periodic steady-state with a stroke volume difference below 0.5 ml. Results are evaluated for pressure curves, volume curves, pV loops and in case of the whole heart setups flow rates through the valves as well.

6.3.4. Results Sensitivity Analysis

Figure 6.27 shows the sensitivity of the CircWholeHeart model with baseline parameters in quasi-static state. Each parameter was modified by $\pm 50\%$ in both directions and the simulated time was measured until the system reaches equal stroke volume again, with exceptions for V_{total} (15%), max tension (25%) and PQ interval (33%). These were the parameters that showed the largest effect, and where simulations failed with the initial modification by 50%. The circulatory system still shows its largest sensitivity when modifying the total blood volume V_{Total} with up to 40 ml difference in stroke volume between left and right ventricle. In that case, the system takes more than 28 s to reach a converged state again. The plot shows a linear reduction for most of the parameters on the logarithmic scale, and the slope can be used to estimate time to convergence in future modifications.

A sensitivity quotient S of a parameter x with respect to a certain output parameter Y was defined as a central differences approximation, normalized relative to the value at

6. Improving the Physiological Behavior

baseline parameter:

$$S := \frac{Y(x_{base} + \Delta x) - Y(x_{base} - \Delta x)}{2 \Delta x} \cdot \frac{x_{base}}{Y(x_{base})} \cdot 100 \%, \quad \Delta x = 0.5 x_{base}$$

Hence, S describes the change in percent to a modification of that parameter by 50%. Sensitivity S_{max} of all parameters is given in Table 6.7, evaluated with peak values Y_{max} obtained over one beat at new steady state. Values that change by more than 5% are marked in bold. It is notable that there is a diagonal structure visible, which describes that values measured close to the parameter are more sensitive to a change than values measured further away.

Furthermore, pressure time courses of both ventricles and arteries are given in Figure 6.30 for a selection of parameters. One parameter modification of clinical relevance is the compliance of the arteries and the aortic arch C_{SysArt} , which is known to decrease with age. The most notable effect of a change in peripheral resistance R_{SysPer} is a global shift of pressure levels in the left ventricle and the aortic pressure. A change of C_{SysArt} keeps the left ventricular peak pressure the same but changes the shape between round and symmetric to a pronounced peak tilted to the left. Decreasing C_{SysArt} lowers the minimal aortic pressure while keeping the peak pressure similar. The effect of changing parameters in pulmonary circulation is similar.

It however has to be noted, that these data do not directly transfer to clinical environment and a human organism might react different. The described sensitivity shows purely effects of the circulatory system and does not contain physiological control mechanisms, e. g. no baroreceptor feedback.

6.3.5. Results One Ventricle

Figure 6.31 shows the results of the left ventricle coupled to “CircOneVentricle”, where the outflow is connected to the inflow of the same chamber via a closed-loop Windkessel model describing systemic circulation. The model was configured with the previously described baseline parameters and a total blood volume of 4700 ml. Simulations were run with negative pressure preloading (solid lines) as used by Fritz et al. (2014) and without any preloading method (dotted lines) for comparison. The numbers (1) - (4) describe different characteristic points in the cardiac cycle: (1) closing of the mitral valve and beginning ventricular contraction, (2) opening of the arterial valve, (3) closing of the arterial valve and (4) opening of mitral valve and beginning of diastole.

Largest effect of preloading is a shift of ventricular and venous volumes, while the volume difference between end-systole and end-diastole stays the same. Preloading helps with keeping the end-diastolic left ventricular volume lower and closer to original volume that was obtained from a segmentation at end-diastolic state. Peak pressure is slightly lower for the simulation with preloading as well, and the shape is slightly more tilted to the left. Overall, preloading reduces end-systolic and end-diastolic pressure-volume ratio.

The large venous compliance greatly smooths the venous pressure, which is almost constant throughout the whole cardiac cycle (gray in top left diagram). As the venous

6. Improving the Physiological Behavior

Table 6.7.: Sensitivity of output maxima S_{max} , from Schuler (2016).

	R-SysArtValve	R-SysArt	C-SysArt	R-SysPer	C-SysVen	R-SysVen	R-RayValve	R-PulArtValve	R-PulArt	C-PulArt	R-PulPer	C-PulVen	R-PulVen	R-LavValve
V _{Lv}	0.0	0.5	-2.5	3.7	-8.0	-2.0	-0.1	-0.0	-0.1	-1.2	-1.8	-0.9	-0.4	-0.2
V _{SysArt}	-0.0	-0.2	8.5	6.3	-2.3	-0.4	-0.0	-0.0	-0.4	-0.4	-0.3	-0.1	-0.0	
V _{SysVen}	-0.0	-0.2	-1.3	-3.1	4.4	1.1	0.1	0.0	0.0	-1.0	-0.1	-0.6	-0.1	-0.0
V _{Ra}	-0.0	-0.1	-0.6	-1.6	-3.6	-1.6	-0.0	0.0	0.1	-0.5	0.1	-0.3	0.1	-0.0
V _{Rv}	-0.0	-0.3	-1.4	-3.9	-8.2	-2.6	-0.5	0.0	0.0	-1.1	0.5	-0.6	0.2	0.0
V _{PulArt}	0.0	0.7	-4.8	3.6	-18.1	-5.0	-0.5	-0.0	-0.2	18.4	7.0	-2.8	0.7	0.1
V _{PulVen}	0.0	1.1	-6.4	7.6	-22.0	-5.9	-0.5	-0.0	-0.3	-3.7	-4.8	11.6	0.4	0.2
V _{La}	0.0	0.5	-2.5	3.3	-8.4	-2.2	-0.2	-0.0	-0.1	-1.4	-1.9	-2.6	-0.1	0.1
P _{Lv}	0.7	8.9	-2.4	23.2	-10.3	-2.1	-0.2	0.0	-0.0	-1.7	-2.0	-1.3	-0.4	-0.1
P _{SysArt}	-0.3	9.2	-2.9	23.9	-10.5	-2.1	-0.2	-0.0	-0.0	-1.7	-2.0	-1.3	-0.5	-0.1
P _{SysVen}	-0.0	-0.8	-5.8	-14.1	-35.8	5.0	0.5	0.0	0.1	-4.8	-0.2	-2.7	-0.2	-0.1
P _{Ra}	-0.0	-0.7	-5.9	-12.7	-34.5	-8.2	-1.5	0.0	0.1	-4.6	0.9	-2.7	0.5	-0.0
P _{Rv}	0.0	0.6	-6.0	1.9	-23.6	-6.9	-0.8	2.2	15.0	-3.2	7.5	-3.3	1.0	0.1
P _{PulArt}	0.0	0.6	-6.4	2.4	-24.9	-7.2	-0.8	-0.8	16.7	-3.9	8.0	-3.5	1.0	0.1
P _{PulVen}	0.1	3.1	-17.5	21.0	-60.8	-16.3	-1.4	-0.1	-0.9	-10.3	-13.3	-20.4	1.0	0.5
P _{La}	0.1	3.4	-17.6	24.5	-59.5	-14.9	-1.2	-0.1	-0.5	-8.9	-12.1	-6.7	-1.9	-0.4
Q _{SysArt}	-1.2	-14.8	6.8	-10.5	-4.2	-1.6	-0.1	0.0	0.0	-0.7	-1.2	-0.5	-0.3	-0.1
Q _{SysPer}	-0.1	-1.0	-9.5	-16.4	-9.4	-2.8	-0.2	-0.0	-0.1	-1.7	-2.3	-1.3	-0.5	-0.1
Q _{SysVen}	0.0	-0.4	-2.1	-7.6	-14.6	-35.3	-2.2	-0.0	-0.2	-1.4	-1.2	-0.7	0.8	0.1
Q _{Rav}	0.0	-0.2	-0.6	-2.6	-3.4	-1.6	-3.7	-0.0	-0.0	-0.5	0.0	-0.3	0.1	0.0
Q _{PulArt}	-0.0	-0.5	-0.3	-5.7	-4.3	-1.9	-0.5	-1.9	-13.1	5.2	-0.3	-0.0	0.1	-0.0
Q _{PulPer}	-0.0	-0.7	-1.2	-7.5	-9.1	-3.2	-0.6	-0.3	-1.7	-19.0	-19.1	2.0	1.8	-0.0
Q _{PulVen}	-0.1	-1.1	-0.5	-8.1	-8.1	-2.3	-0.2	-0.0	-0.1	-1.6	-1.9	-1.2	-31.7	-1.6
Q _{Lav}	-0.0	-0.0	-1.5	-1.2	-7.6	-1.9	-0.1	-0.0	-0.1	-1.3	-1.7	-2.6	-8.0	-2.5

system is connected directly to the ventricle without atria, opening of the mitral valve leads to a very pulsatile increase of venous flow at beginning of diastole (gray in bottom left diagram at (4)), with a negative ventricular pressure that supports the ventricular refilling with blood as well. This is different in case of the arterial flow, where the large arterial compliance leads to a smoother increase of the flow at the beginning of systole (orange in bottom left diagram at (2)) by storing blood in the systemic volume and the aortic arch (orange in second diagram from the bottom).

6.3.6. Results Two Ventricles

Figure 6.32 shows the same simulation setup, but with the “CircTwoVentricles” model that now includes left and right ventricle. As in the previous simulation, the circulatory system parameters are set to the baseline parameters. However, the total blood volume does not include the atria here and was reduced from 5500 ml to 5370 ml. Negative-pressure preloading was used.

The results in the left ventricle are very similar to the simulations with the one ventricle model. This is a hint that direct interaction between the ventricles does not occur, which might be caused by the fact that there is no modeling of the enclosing pericardium.

The left ventricular pressure shows a small difference during diastole. Being now connected to pulmonary instead of systemic venous system, it has a more pulsatile pressure (black in top right diagram at (4)) due to the smaller venous compliance.

6. Improving the Physiological Behavior

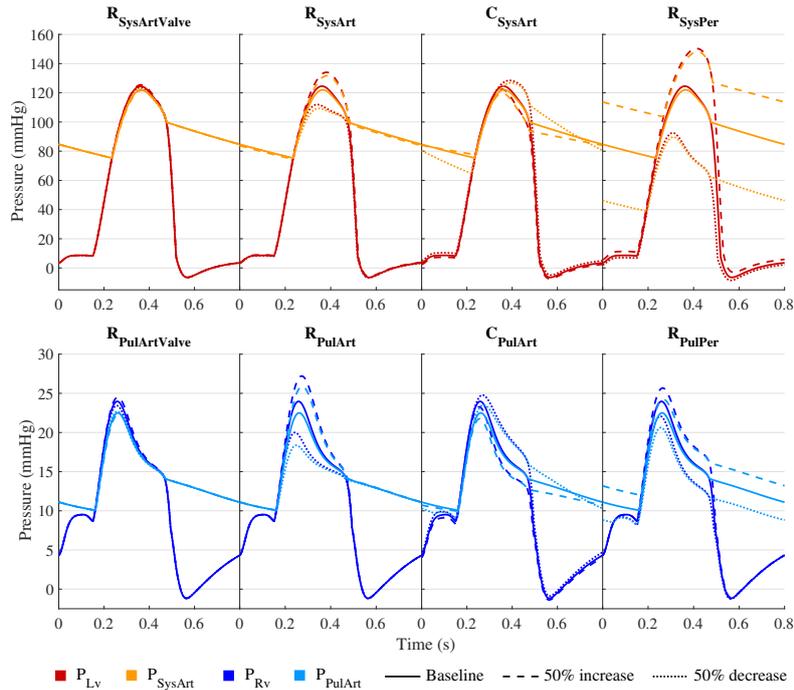


Figure 6.30.: Ventricular and arterial pressure time courses under modification of selected parameters, from Schuler (2016).

The right ventricle has a thinner myocardial wall than the left ventricle and generates less force. That means its force acts less dominant and the shape of its pressure time course during ejection is more influenced by its afterload and more tilted to the left than in the left ventricle (dark blue in top left diagram, between (2) and (3)). Similarly, the right ventricle has a smaller passive compliance, resulting in a slower volume increase during the relaxation phase (dark blue in mid-left diagram, at (4)) and a continuous pressure increase (no plateau) until the mitral valve opens again (dark blue in top left diagram, before (1)). As result, the isovolumetric phases in the pV loop are shorter (bottom left diagram, between (1) and (2), and between (3) and (4)) than in the left ventricle. The pV diagram of the right ventricle shows the typical triangular shape as described by Redington et al. (1988), Bishop et al. (1997).

6.3.7. Results Whole Heart

Figure 6.33 show the simulation with the “CircWholeHeart” model where all four chambers are connected to systemic and pulmonary circulation and pressure is applied to the atria as well.

Modeling of the atria adds some additional features. Due to the contracting atria, time courses show the “a” wave in atrial pressure and an increase of ventricular volume towards end of ventricular diastole. In the flow rates, this is visible by an “A” peak in the flows through the left (Lav) and right (Rav) atrioventricular valves (corresponding

to venous flows in the previous models without atria). In the pV loops of the left and right ventricle, the onset of atrial contraction causes a bump in the bottom right corner of the plots. This however appears only with mass inertia in the FEM simulation, which lets the ventricular volume increase lag behind the small but rapid pressure increase created by atrial contraction. Similarly, the rapid blood flow into the ventricles at the beginning of ventricular relaxation is shown by a typical “E” peak in the flow rates. Atrial pV diagrams normally show two loops, a counterclockwise “A” loop from atrial contraction and relaxation, and a clockwise “V” loop from passive atrial deformation during ventricular contraction and ventricular filling as described in Figure 2.8. In the simulation, the “A” loop is visible for both atria, but only the left atrium shows a correct clockwise “V” loop and a volume increase over the onset of atrial contraction. The counterclockwise “V” loop in the right atrium has the wrong direction, possibly due to specific fixation of the atria and absence of a pericardium. Presence of the pericardium would probably help the atria to better reproduce correct “V” loops by keeping their cross-section area in the upper part constant when the ventricles pull the lower part downwards.

6.3.8. Results Whole Heart with Pericardium

Figure 6.34, Figure 6.35 and Figure 6.36 show the results of the final simulation that includes contact pericardium modeling. Different from the previous simulations, the mesh now consists of second-order tetrahedra and contains truncated representations of the veins, allowing to move Dirichlet fixations completely away from the myocardium (atrial orifices and apex) to the outer end of the trunks and the outside of the pericardial mesh. The Rayleigh damping parameter had to be increased from $\beta = 0.005$ to $\beta = 0.02$ to improve convergence of the FEM solver.

Figure 6.34 shows the pV loops of the four chambers. Most notably, the atrial loops are different and now have a correctly rotating “v” loop as well as a volume increase during ventricular systole similar to measurements by Ferguson et al. (1989); Matsuda et al. (1983). An explanation is that the contact handling restricts the outer shape of the atria to the inner shape of the rather rigid pericardium, causing a volume increase when the contracting ventricles pull the valve plane towards the apex. Maximal atrial volumes are thereby much larger, while the atrial minimal volumes stay the same. Ventricular loops also look different, especially the effect of mass inertia on the lower right corners is more pronounced. Ventricular stroke volumes are both larger, as the pericardial restriction of transversal diameter is here present as well. This effect actually “reshapes” ventricular motion such that the valve plane moves further towards the apex during contraction as described previously by Fritz et al. (2014).

Figure 6.35 gives time courses of pressure and volume. Presence of the pericardium produces a more linear ventricular volume increase during first half of relaxation instead of an inverse exponential one. Ventricular pressure peaks are less pronounced than in the previous simulation. The pericardium causes more interaction between the ventricles, which shows the fact that minimal pressures in left and right ventricle are now on the same level, and the ventricular volumes increase synchronously during early filling.

6. Improving the Physiological Behavior

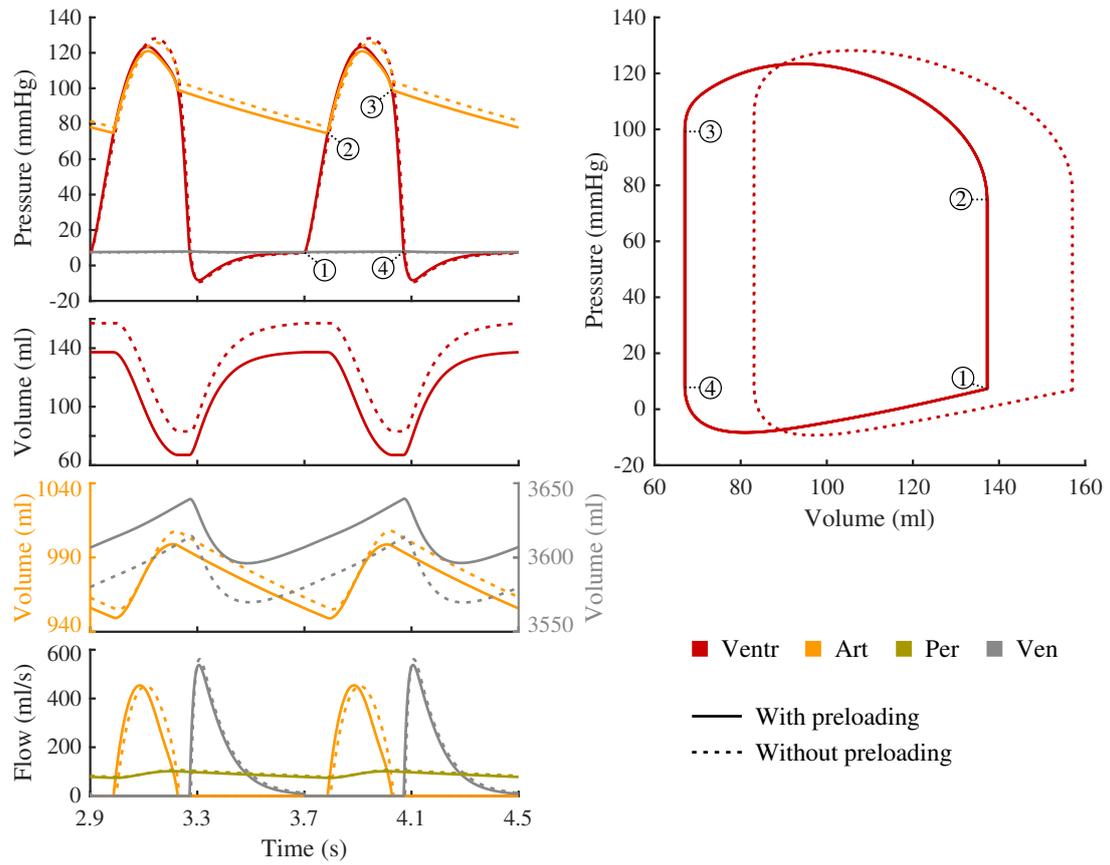


Figure 6.31.: Results with the “CircOneVentricle” model as representation of the systemic circulation, from Schuler (2016).

6. Improving the Physiological Behavior

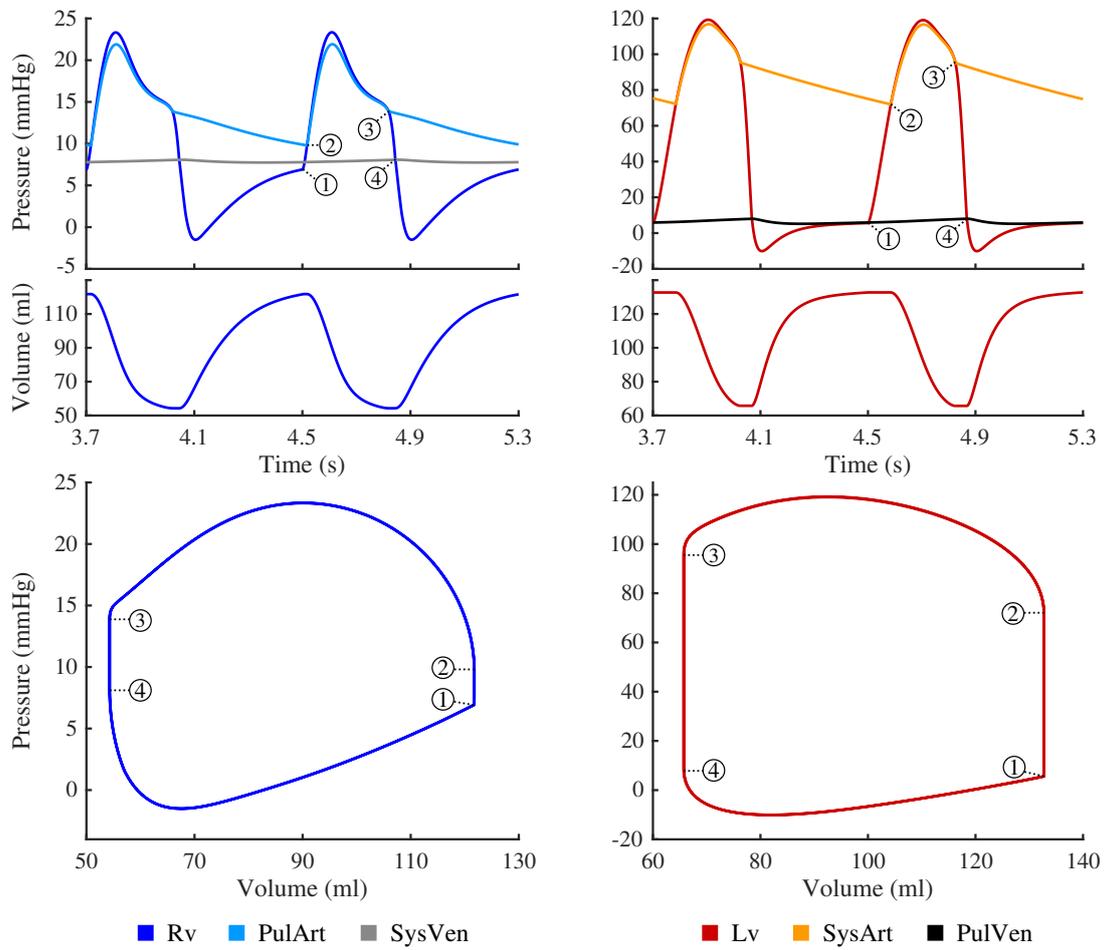


Figure 6.32.: Results with the “CircTwoVentricles” model, from Schuler (2016).

6. Improving the Physiological Behavior

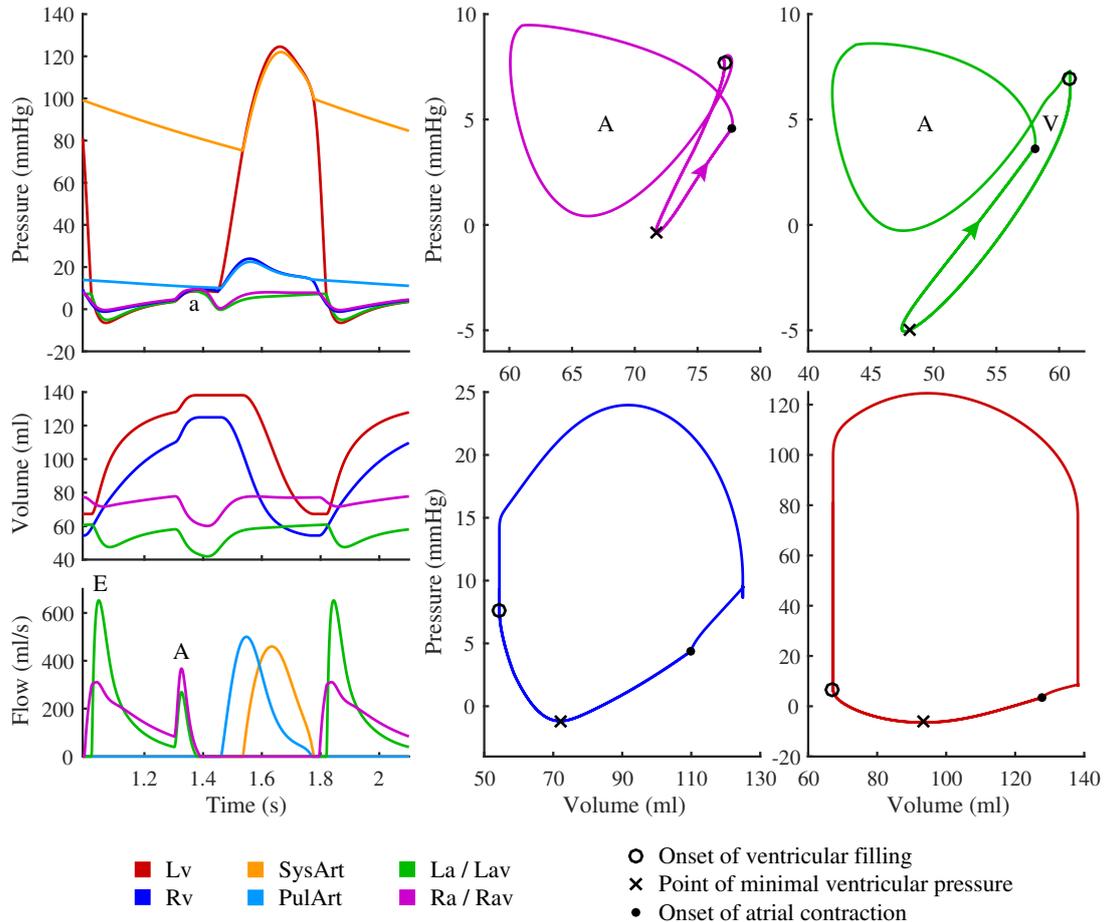


Figure 6.33.: Results with the "CircWholeHeart" model, from Schuler (2016).

6. Improving the Physiological Behavior

Atrial pressures show the typical “a” wave, “x” wave, “v” wave and “y” wave similar to measurements of normal humans by Matsuda et al. (1983); Ferguson et al. (1989).

The time courses of flow rates are given in Figure 6.36. Flows through the valve plane between atria and ventricles (L_{av} and R_{av} in pink and green at the bottom) show two positive peaks, a smaller one during atrial contraction (“A” wave), and a larger one during early ventricular refilling (“E” wave), similar to measurements at a healthy 28 year old male by Zoghbi et al. (1990). The “E/A ratio” between the peaks is a diagnostic marker of diastolic dysfunction (Galderisi, 2005), with a value > 1 describing the normal case, a value < 1 describing abnormal relaxation (too slow filling), and a value > 1.5 denoting restrictive patterns (too fast filling, this is here the case). These two peaks are also found in the pulmonary vein flow, but there is also an additional, negative peak. As there are no valves between pulmonary vein and the left atrium, Doppler measurements by Firstenberg et al. (2000) indeed show an atrial reversal flow (“AR”) into the vein, together with the “S” wave during atrial systole and the “D” wave during ventricular diastole. Hence, the simulation reproduced all major waves found in measurements of flow.

As the geometric model used for these simulations did not have a volumetric representation of the valve plane, the simulations still contain a small modeling shortcoming. The entirety of surface forces due to pressure of a closed cavity usually sum up to zero. Since there are no volume elements the ventricular pressures that normally act onto the valve could be applied to, these forces were simply ignored. That way however, they do not sum up to zero anymore and slightly push the ventricular myocardium towards the apex, similar to wind blowing into a ship’s sail. This effect however only gets visible with weaker passive parameters and during specific phases of the cardiac cycle.

6.3.9. Summary, Conclusion and Outlook

In this application, a hierarchy of closed-loop circulatory systems was created on the basis of lumped parameter modeling for one ventricle, two ventricles and the whole heart, and baseline parameters provided from a literature research. It fulfills blood conservation by design, and the approach with separate valve states allows to compute multiple beats until convergence. The model was integrated into the Biomechanics framework and coupling was realized by an efficient quasi-Newton algorithm for strong iterative coupling with full control of the coupling error. The circulatory systems framework runs on a smaller timescale allowing for a more accurate time integration, and performs closed-surface and steady-state check to detect logical input errors and convergence.

Summarizing, the results in the left ventricle extend well from `CircOneVentricle` to `CircTwoVentricle` and in both ventricles from `CircTwoVentricle` to `CircWholeHeart`. This is a good indication that approaches chosen for time integration and coupling and implementation are suitable. Especially the simulation using whole heart with pericardium is able to reproduce major diagnostic factors in pressure, volume and flow. These are in particular:

1. diastolic decay of arterial pressures

6. Improving the Physiological Behavior

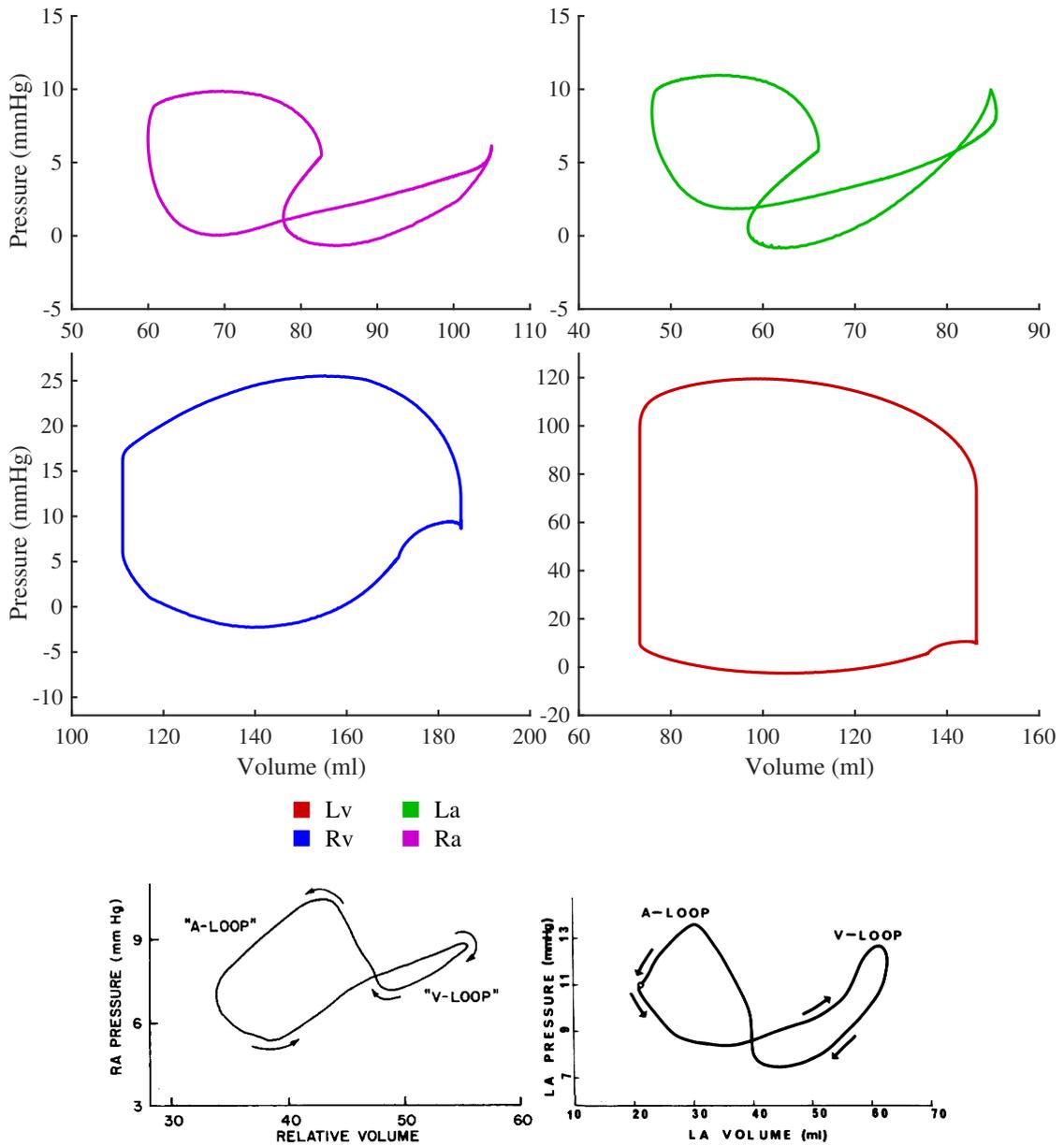


Figure 6.34.: Resulting pressure volume loops with the “CircWholeHeart” model including contact pericardium (top) and from atrial measurements by Ferguson et al. (1989); Matsuda et al. (1983), adapted from Schuler (2016).

6. Improving the Physiological Behavior

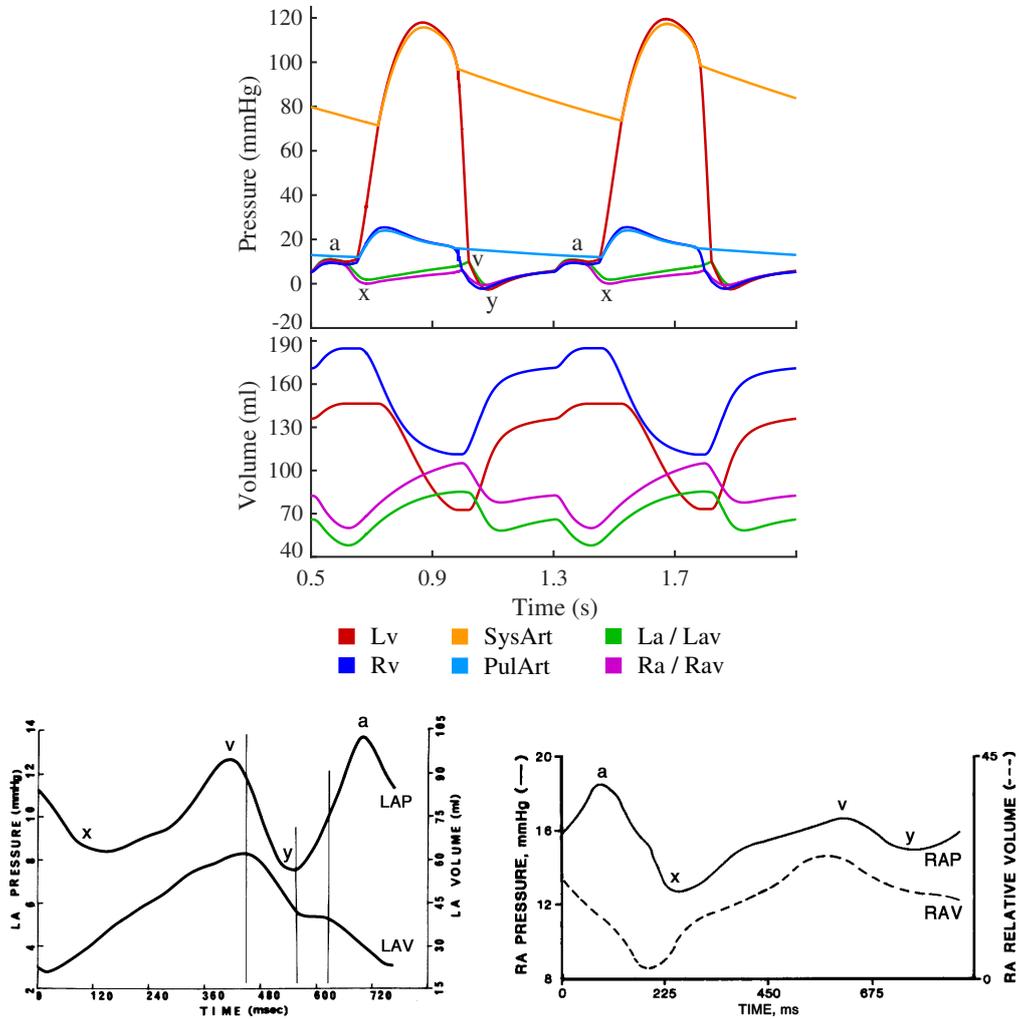


Figure 6.35.: Resulting pressure and volume time courses with the “CircWholeHeart” model including contact pericardium (top) adapted from Schuler (2016), and left and right atrial pressures measured in humans (bottom left and bottom right) adapted from Matsuda et al. (1983); Ferguson et al. (1989) by Schuler (2016).

6. Improving the Physiological Behavior

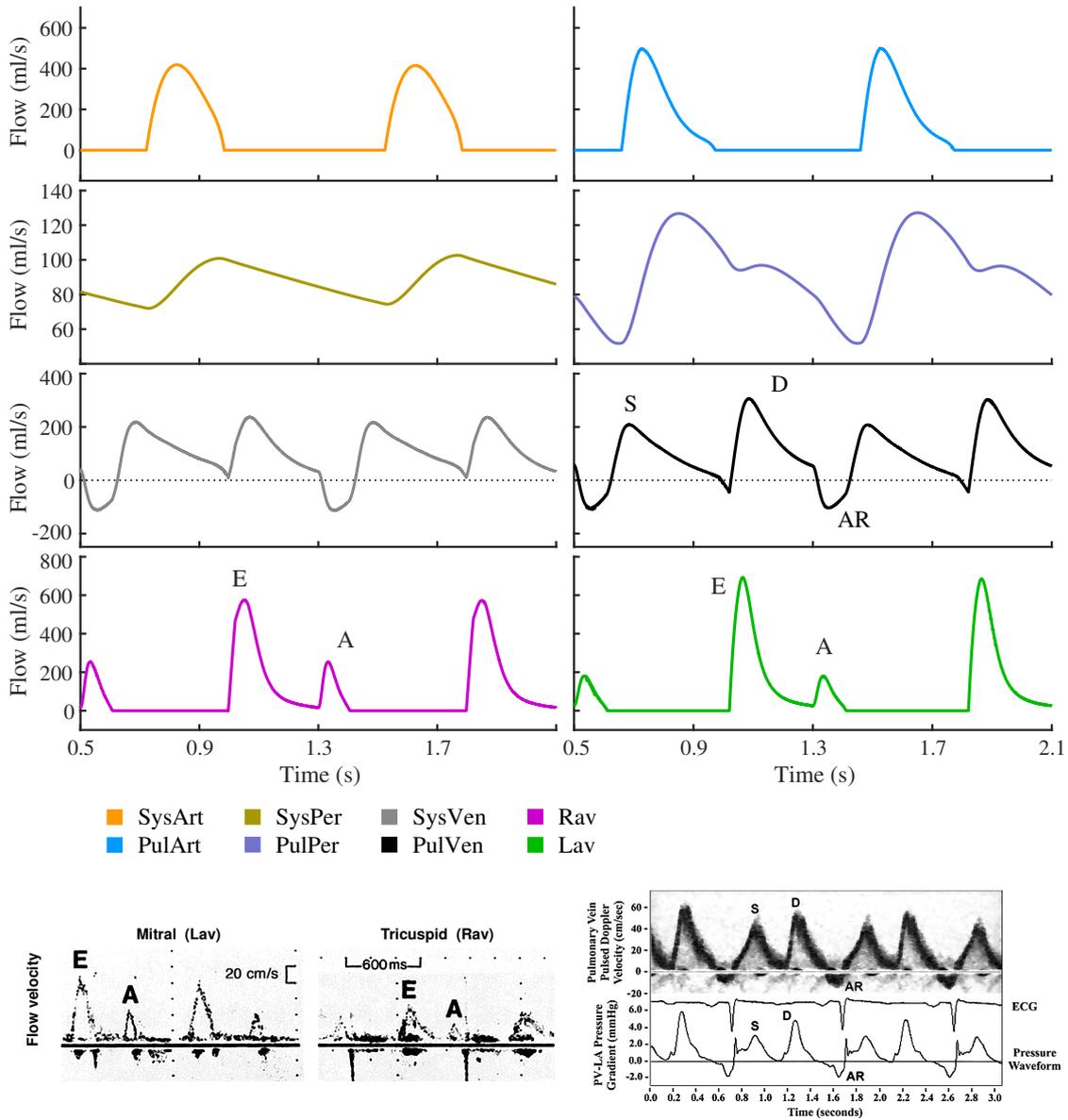


Figure 6.36.: Resulting flow rates through the various valves with the “CircWholeHeart” model including contact pericardium (top) and flow rates from measurements Zoghbi et al. (1990) (bottom left) and by Firstenberg et al. (2000) (bottom right), adapted from Schuler (2016).

6. Improving the Physiological Behavior

2. increase of ventricular pressure and volume during atrial contraction
3. “a”, “x”, “v”, “y” waves in atrial pressures
4. pV loops split into A loop and V loop in the atria
5. “E” and “A” waves in the atrio-ventricular flows, but not the “L” wave
6. “S”, “D” and “AR” waves in the venous flows

Interestingly, these plots already contain hints that the passive parameters are too stiff during ventricular relaxation. The ventricular pressures drop more than observed in measurements during early ventricular filling and drop even below zero, which expresses a “sucking” effect of the ventricles. Due to a too large end-systolic passive influence, possible solutions are to use more preloading (smaller pressure-free volume reduces the local strain there) and a generally weaker ventricular passive behavior (requires different parameters). Furthermore, there are two phenomena related to inertia of blood that were not reproduced by the circulatory system model in the plots. These are an additional “L” wave between “E” and “A” wave in mitral valve flow sometimes visible at higher heart rates (valve bulging from ventricle to atrium), and the “dicrotic notch” in aortic pressure (backflow into the ventricle from aortic valve while closing non-instantly, reflection of aortic pressure from the bifurcations) present in any textbook illustration and often used as marker for end-systole in clinical data.

In future works of the circulation modeling, blood inertia and a more sophisticated modeling of the heart valves should be included. Such a dynamic opening and closing of the valves instead of instantaneous events can dramatically impact simulated pressure and flow curves. This has the potential to reproduce further properties of the pressure and flow curves: dicrotic notch, an increase of arterial pressure after closing of the valve was not seen even in any of the simulations shown here. With regards to application in a coupled multi-component system, development of the single components in a common sense would be beneficial. Due to strong coupling, it actually is necessary to choose circulation parameters together with passive ventricular parameters as the latter will significantly influence ventricular relaxation phase. A first step in this direction is the inclusion of passive parameters from Section 6.1 and use a different method to generate preload, however here a ‘ground truth’ is difficult to obtain. The same holds true for the behavior of the modeled pericardium, that gets more dominant when active forces are absent, but it is still very unclear in which phases of the cardiac cycle it supports or impedes the cardiac motion.

6.4. Application IV – A zero-dimensional Pericardium Modeling

Goal of this project is to create and evaluate a reduced-order version of the pericardium model presented by Fritz et al. (2014). Their existing model adds a significant amount of complexity at various stages: during mesh generation (two fitting meshes needed), during simulation setup by choice of its non-physics-related parameters for the surrounding

6. Improving the Physiological Behavior

tissue material (arbitrary thickness, stiffness, no volume conservation), and even during simulation time (majority of simulations initially failed due to non-perfect choices in the previous tasks). With the pericardium model by Kerckhoffs et al. (2007a), a zero-dimensional model was already presented in literature but not yet used in a four-chamber setup, or even compared against the Fritz pericardium. Such a reduced pericardium has a number of advantages over the 3d formulation and could be better suited to flatten the road towards clinical applications due to the possibility for a drastically easier mesh generation workflow and a lower number of more consistent choosable model parameters. The characteristics of such a model is very similar in structure to models of the circulatory system (takes volume changes, yields pressures), and can be implemented without additional effort in already existing frameworks of circulatory systems e.g. as the one described in Section 6.3.2.

6.4.1. Modeling

In this study, two different approaches to pericardium modeling were used. The first one is the *contact pericardium* model by Fritz et al. (2014); Fritz (2015) which was already implemented in an earlier version of the framework due to these publications. It uses an additional separate three-dimensional mesh around the myocardial mesh and computes surface forces to create a frictionless pericardium with permanent contact (Fritz et al., 2014). The second one is a lower dimensional model from Kerckhoffs et al. (2007a) which was originally used for 3D finite element simulations of canine ventricles where the atria were represented only as part of the circulatory model. It uses an exponential relationship based on total heart volume (calculated from blood, myocardium and atrial contributions) to compute an external pressure acting on the epicardial surface due to the pericardial sac. It was implemented for this study in the circulatory system framework from Section 6.3.2. Both models are described in detail in Section 3.8.

Available Measurement Parameters

For validating a correct pericardial influence, suitable measurement data are crucial.

Carlsson et al. (2004) observed the variation of total heart volume of a heart cycle in eight human subjects. The data was obtained using cine MRI and comprises 13 slices and 15-21 images per cycle. Their data includes ventricular, atrial and myocardial volume. The measurements show a relative total heart volume change of $8.2 \pm 0.8\%$ (Mean \pm SE) when using image segmentation (“volumetric measurements”) and $8.8 \pm 1.0\%$ when using flow integration (“flow measurements”).

Fritz et al. (2014) used cine MRI images from the same patient as used for simulation to compute the displacement of the atrio-ventricular valve plane over time. Their publication struggled to reach the measured data of 1.7 cm in the simulations (only 1.04 cm with and 0.63 cm without contact pericardium). Later investigations with the original mesh and model parameters showed an unusual stiff passive behavior of the isolated ventricles under passive inflation (see Section 3.11). The total heart volume variation however was still around 8% and within the measurements by Carlsson et al. (2004), indicating an

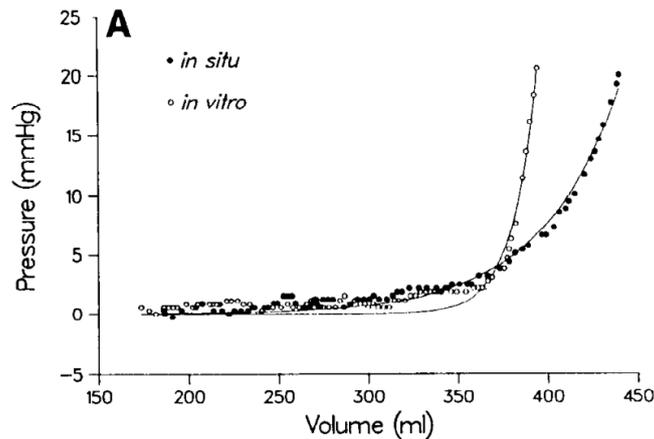


Figure 6.37.: Measurements of the pressure-volume response of the pericardium embedded in surrounding tissue (*in situ*, black dots) and without (*in vitro*, white dots), from Freeman and Little (1986).

unfortunate set of parameters that “fixes” or hides controversial behavior of the single modeling components.

Freeman and LeWinter (1984); Freeman and Little (1986) measured pericardial pressure volume curves from six dogs. Measurements were done *in situ* embedded in the chest, as well as *in vitro* testing only the pericardial sac, the heart was removed and not present in both cases (see Figure 6.37). They found an exponential relationship and derived parameters for α and β . Kerckhoffs et al. (2007a) used these model parameters α and β unmodified from a later study by Freeman and Little (1986) for the pressure pericardium. One strange observation from Figure 6.37 is, that above 3 mmHg the pericardium alone (*in vitro*) seems to behave stiffer than pericardium including surrounding tissue (*in situ*). Unfortunately they did not point out the normal volumes during a healthy heart cycle in these dogs, which could be used to approximately derive a volume relationship between normal dog hearts and pressure-free pericardium. Together with the measurements by Carlsson et al. (2004), this would give a hint of the typical in-vivo working range of pericardial pressure.

Modifications applied to the Kerckhoffs model

Due to a lack of an atrial representation in their original 3D mesh, Kerckhoffs et al. (2007a) applied pericardial pressure only to the ventricular epicardial surface, and subtracted it from the atrial pressure in the circulatory system instead. In the mesh used in this study however, a volumetric representations of the atria exist. So, the pericardial pressure could be applied directly to both, the surface of ventricular and surface of atrial epicardium. Hence, a special consideration of atrial pressure in the circulatory system was not needed.

Due to the formulation of the pressure curve, the Kerckhoffs model generates a permanently positive pressure over the whole volume range. Since such a permanently

6. Improving the Physiological Behavior

positive pressure acting on the epicardial surface conflicts with the required equilibrium of “zero forces” in the initial state, all simulations failed to start. To overcome this issue, a *pressure growth duration* was implemented, that linearly scales the parameter α in order to smoothly activate the pericardium model over a certain number of time steps. This retains the initial equilibrium condition.

The modified Kerckhoffs model will be called *pressure pericardium* through the remainder of this study.

Modifications applied to the Fritz model

The Fritz model is a complex model and existed in an even more complex, undocumented implementation in a style typical of research code. A number of crucial bugs and flaws existed, that needed to be fixed: Simulations often became unreliable and failed non-reproducibly as soon as the pericardium was activated. By theoretical considerations, the normals of a master element and its corresponding slave element should align in parallel to create no unwanted friction force (“frictionless pericardium”). However, choosing different passive myocardial material parameters (e.g. weaker behaving ones with a more realistic pV response) repeatedly lead to mostly parallel surfaces where few of the triangles were not oriented “in-plane”. In consecutive time steps these created problems with non-matching master-slave pairs being pulled in unintended directions, which continuously worsens the situation and finally leads the simulation to fail within a few steps. Moreover, the master surface used by Fritz et al. (2014) contained holes in it that emerged from removing these “problematic triangles”, which is a lot of manual work and changes with each new simulation setup. The potentially problematic triangles have no physical or physiological meaning and therefore cannot be identified other than by many iterations over unsuccessful simulations and deletions of triangles. This point seems to be a huge compromise needed for getting the complex pericardium modeling initially to work.

To overcome these issues, a completely new mesh was generated with slightly more elements. Since the original epicardial surface was neither closed nor convex, volumetric fat tissue was added to the space in-between myocardium and its convex hull, tightly connected to the myocardium. The surface of master elements could then be defined as the closed and now convex surface of the two materials. Fat was modeled as a non-contracting, incompressible and isotropic Mooney-Rivlin material (parameters: $C_1 = 3725$ Pa, $C_2 = 0$ Pa, $B=2E5$ Pa). Especially near the valve plane, this creates a closed “ring made of fat” which surrounds and stabilizes the valve region.

A second improvement covers the algorithm used for search of corresponding slave element candidates that intersect the normal of the master element. That search function runs in each simulated step for each master element, and was found to be unreliable and slow – especially with the larger number of master elements – as it always checked every single slave element ($\mathcal{O}(n^2)$). For speedup, a neighborhood map was added for each slave element, which gets initialized once during start of the simulation. The time spent in the search function now could be vastly improved by starting at the previous found slave element and iterating first through its neighbors, instead of going through all slave

6. Improving the Physiological Behavior

elements in fixed order. Since the displacements between time steps are generally small, the number of tested slave element candidates reduces in average to around 10 elements, instead of half the slave elements vector. The algorithmic complexity reduced from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ (with the number of slave elements n), which is important when using larger mesh sizes (the mesh used in this study has now twice as many slave elements).

The modified Fritz model will be called *contact pericardium* through the remainder of this study.

6.4.2. Simulation Setups

All simulations in this study base on the new heart model that includes truncated veins and the volumetric valve plane as depicted in Figure 4.9 in Section 4.5. Passive material parameters were chosen according to the results from Section 6.1. Unloading was done once by a fixed number of Bols iterations, since the Bols error increased after the first few iterations again. As model for active contraction, the Lumens model was chosen in the ventricles and the DoubleHill model in the atria. The circulatory system model as presented in Section 6.3 was used to obtain realistic chamber pressures. Parameters for the contact pericardium were taken unmodified from Fritz et al. (2014).

Parameters for the pressure pericardium are more difficult to obtain. Unfortunately, Freeman and Little (1986) did not give values for the parameter $V_{tot,0}$, so a simulation study was carried out to test different values for that parameter. The total zero volume $V_{tot,0}$ is mandatory in their formula of the pericardial pressure:

$$p(V_{total}) = \alpha \left(\exp\left(\beta \left(\frac{V_{total}}{V_{tot,0}} - 1 \right)\right) - 1 \right)$$

Therefore, aim of the first simulation study is to find an appropriate set of parameters for the pressure pericardium to get a modeling setup comparable to the contact model.

Aim of the second simulation study is to compare the two available pericardium approaches and the different effects introduced by them. Hence, nine different simulation setups were created and compared with the following three variants of pericardium modeling:

1. no pericardium
2. contact pericardium (Fritz et al., 2014) with various values for $V_{tot,0}$
3. pressure pericardium (Kerckhoffs et al., 2007a)

6.4.3. Results

In the first parameter study, the pressure pericardium model was adapted to produce the same relative total heart volume change as the contact pericardium using the zero volume as free parameter. Similar to the contact pericardium, the model should represent the situation including pericardium and the surrounding areas, hence reflecting the *in situ* measurements from Freeman and Little (1986). Since the pericardium effect should

6. Improving the Physiological Behavior

be fairly pronounced, the parameter set “dog 2” was chosen as it produced the highest pericardium pressure in the relevant range of relative volume increase $V_{tot}/V_{tot,0}$ between 1.0 and 2.0 ($\alpha = 0.2330$ mmHg, $\beta = 6.76$). The only parameter missing is $V_{tot,0}$.

The time courses of total heart volume with different parameters of $V_{tot,0}$ are shown in Figure 6.38, together with the contact pericardium for comparison. A smaller value for the zero volume influenced not only the lowest value obtained during the time course, but especially the maximum volume showed a great dependency. The shape of the curve shows two peaks with a small negative “hump” during atrial contraction around 0.9s, and with larger $V_{tot,0}$ the second peak goes higher than the first one. Compared to the contact pericardium, the shape of the peaks is generally more similar with lower $V_{tot,0}$, however the volume amplitude over the whole beat decreases. The exact volumes and volume differences for the last beat are listed in Table 6.8. It is noticeable that even the largest volume obtained with the pressure pericardium is in all cases clearly below the minimal volume reached with the contact pericardium. Further, the pericardial pressure stays positive (inward pointing, pushing effect) throughout the whole heart beat. This is in strong contrast to the contact pericardium, where the “pressure” (surface force due to contact handling) starts with zero at the beginning of the atrial contraction and becomes negative (outward pointing, pulling effect) during the heart beat. At $V_{tot,0} = 557$ ml, the pressure pericardium produced the same relative volume change as the contact pericardium, which was chosen as “pressure pericardium” in the following comparisons.

Results of the comparison study between “pressure pericardium”, “contact pericardium” and “no pericardium” are shown for the enclosed pericardial volume over time in Figure 6.41, with data of absolute volumes and relative volume change given in Table 6.9. Additionally, volumes over time of the four chambers and the pV loops of the four chambers are shown in Figures 6.39 and 6.40. Similar to the previous comparison with the contact pericardium, the total volume variation in Figure 6.39 shows two distinct inverted peaks at atrial and at ventricular contraction. The negative “hump” during atrial contraction is most pronounced in the simulation with contact pericardium, and this is the only simulation with a real volume reduction during that phase. The simulations with pressure pericardium and without pericardium still increase their volume during atrial contraction, but the increase slows down with beginning of atrial contraction. After atrial contraction, the total heart volume without pericardium is much higher than before atrial contraction, which is less pronounced with pressure pericardium and almost not present with contact pericardium. Interestingly, the volume increase during ventricular relaxation are very different as well. Here, it is the simulation without pericardium that has an almost linear increase, while the pressure pericardium starts fast in the beginning and slows down towards the end, and the contact pericardium has an s-shaped relaxation. All simulations show a double loop structure in the left as well as in the right atrium. However, the larger volume difference is obtained in the setup with the contact pericardium compared to the other two setups.

Figure 6.42 gives a qualitative comparison of the total volume time course between appropriately scaled and shifted simulation to measurement data from Carlsson et al. (2004). While both time courses show the large negative bump due to ventricular contraction (around 300ms in the measurements), the bump due to atrial contraction is

6. Improving the Physiological Behavior

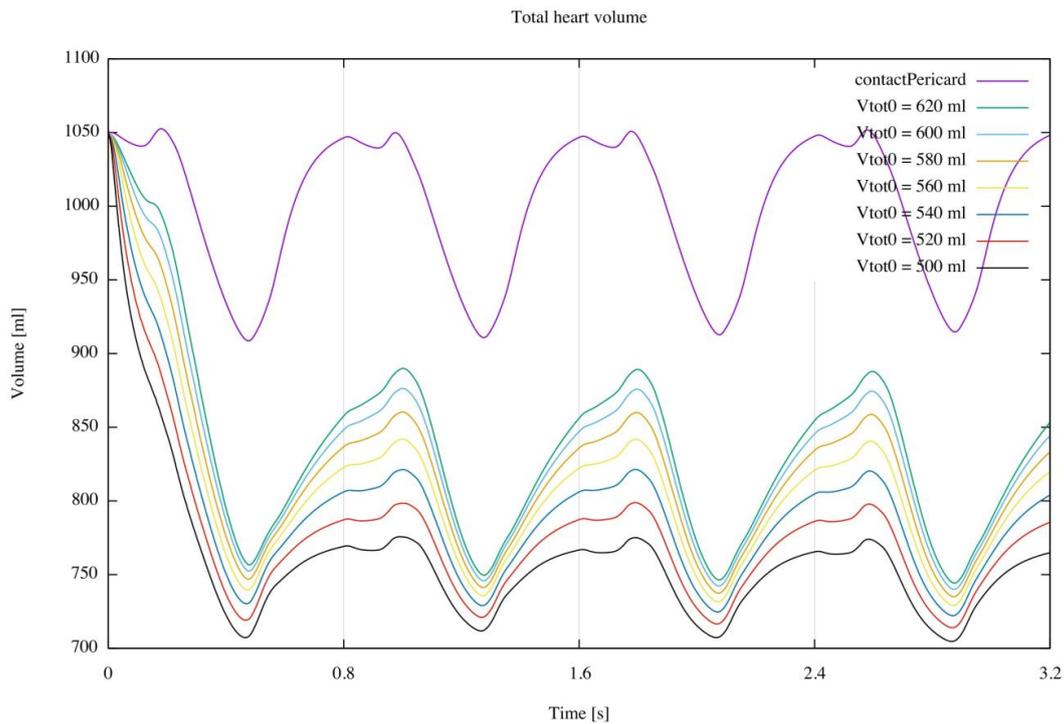


Figure 6.38.: Volume over time plot for different values of the zero volume parameter $V_{tot,0}$, from Fensterseifer Schmidt (2017). That value is an essentially needed parameter for simulation setup, but not given by Freeman and Little (1986).

only visible in the simulation data (blue arrow). The behavior of that “atrial contraction bump” is the same for the other seven subjects from Carlsson et al. (2004) as well.

6.4.4. Discussion

Regarding change of total heart volume, Table 6.8 gives 19.2% for the simulation without pericardium model, which reduces to 13.1% and 13.0% when using either of pressure or contact pericardium models. This is close to the measured $8.8\% \pm 1.0\%$ given by Carlsson et al. (2004). In the zero-volume study, the pressure pericardium was tuned to produce the same total heart volume change as the contact pericardium by an adequate choice of the unknown modeling factor $V_{tot,0}$. According to Fritz et al. (2014), the parameters for the original contact pericardium were chosen to reproduce the outer shape of the heart during the cardiac cycle from cine MRI of the very same patient, however without giving an exact quantification. According to Kerckhoffs et al. (2007a), the original pressure pericardium was designed around pressure volume measurements of the isolated pericardium from Freeman and Little (1986). Due to the relatively small total amount of change, a heart that produces a total volume change in the correct range implicitly shows

6. Improving the Physiological Behavior

Table 6.8.: Evaluated volumes obtained from simulations with the pressure pericardium for different zero volumes $V_{tot,0}$, as well with the contact pericardium and without pericardium as reference, from Fensterseifer Schmidt (2017).

	VOLUME AND PRESSURE RANGES						
	Vtot0	Vmin	Vmax	% ΔV	pmin	pmax	Δp
noPericard	***	761.384	942.011	19.17462 %	0.00000	0.00000	0.00000
contactPericard	***	914.666	1051.75	13.03390 %	0.00000	0.00000	0.00000
pressurePericard	620	744.251	887.859	16.17464 %	0.67005	4.08940	3.41935
pressurePericard	600	740.003	874.485	15.37842 %	0.89523	4.90064	4.00541
pressurePericard	580	734.968	858.761	14.41530 %	1.18530	5.77019	4.58488
pressurePericard	560	729.047	840.673	13.27817 %	1.56006	6.66621	5.10616
pressurePericard	557	728.072	837.76	13.09301 %	1.62498	6.80069	5.17571
pressurePericard	540	722.098	820.312	11.97276 %	2.04405	7.55337	5.50932
pressurePericard	520	713.94	797.886	10.52105 %	2.66640	8.40181	5.73541
pressurePericard	500	704.623	774.005	8.96402 %	3.47238	9.23418	5.76180

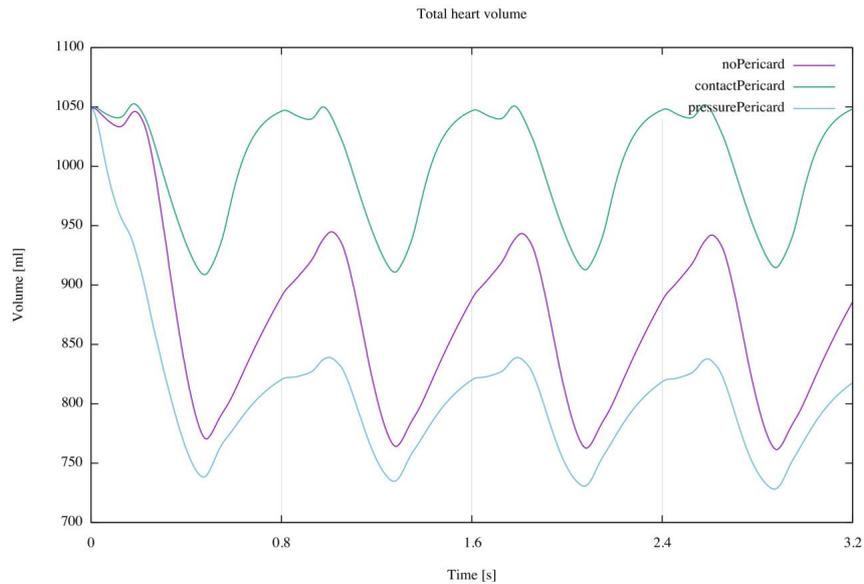


Figure 6.39.: Total heart volume of the three setups, from Fensterseifer Schmidt (2017). The hard-to-read legend labels the curves as noPericard (lilac, middle curve), contactPericard (green, upper curve), and pressurePericard (gray, bottom curve) with the axes showing volume over time.

6. Improving the Physiological Behavior

Table 6.9.: Computed total heart volumes of the three simulation setups. From Fensterseifer Schmidt (2017).

	V_{\min}	V_{\max}	ΔV
noPericard	761	942	19%
contactPericard	914	1051	13%
pressurePericard	728	837	13%

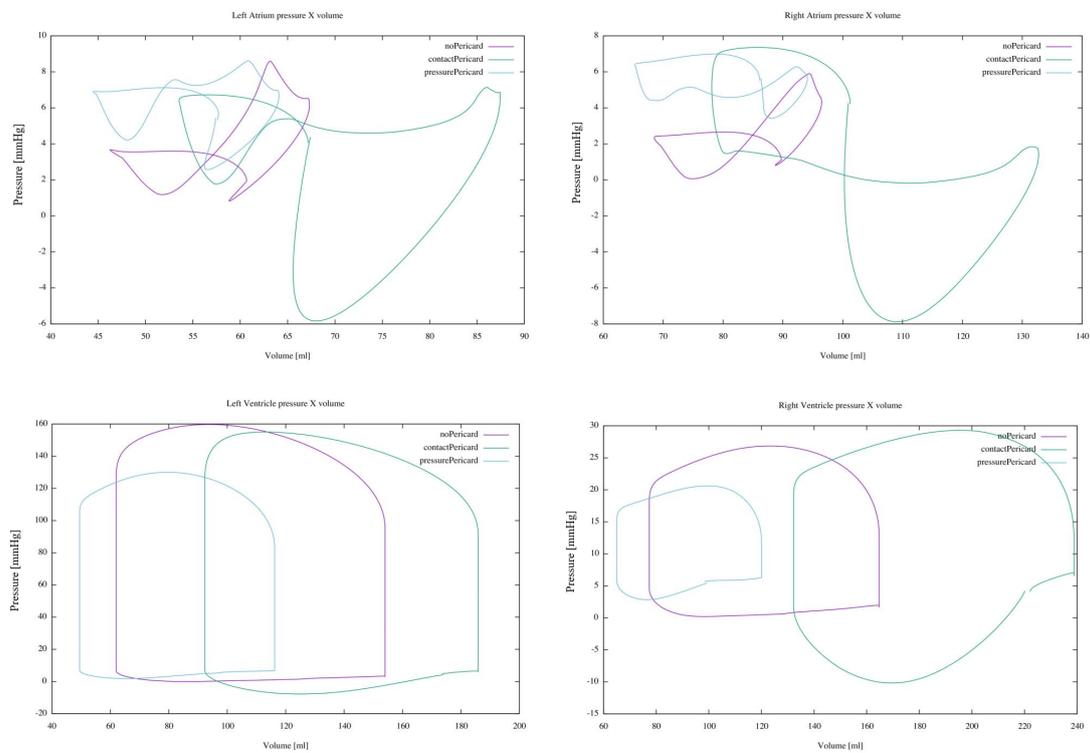


Figure 6.40.: Pressure volume loops of the three setups for all four chambers, from (Fensterseifer Schmidt, 2017). The plots show pV loops in the left atrium (top left), right atrium (top right), left ventricle (bottom left) and right ventricle (bottom right) with pressure over volume on the axes. Color labels are the same as in Figure 6.39.

6. Improving the Physiological Behavior

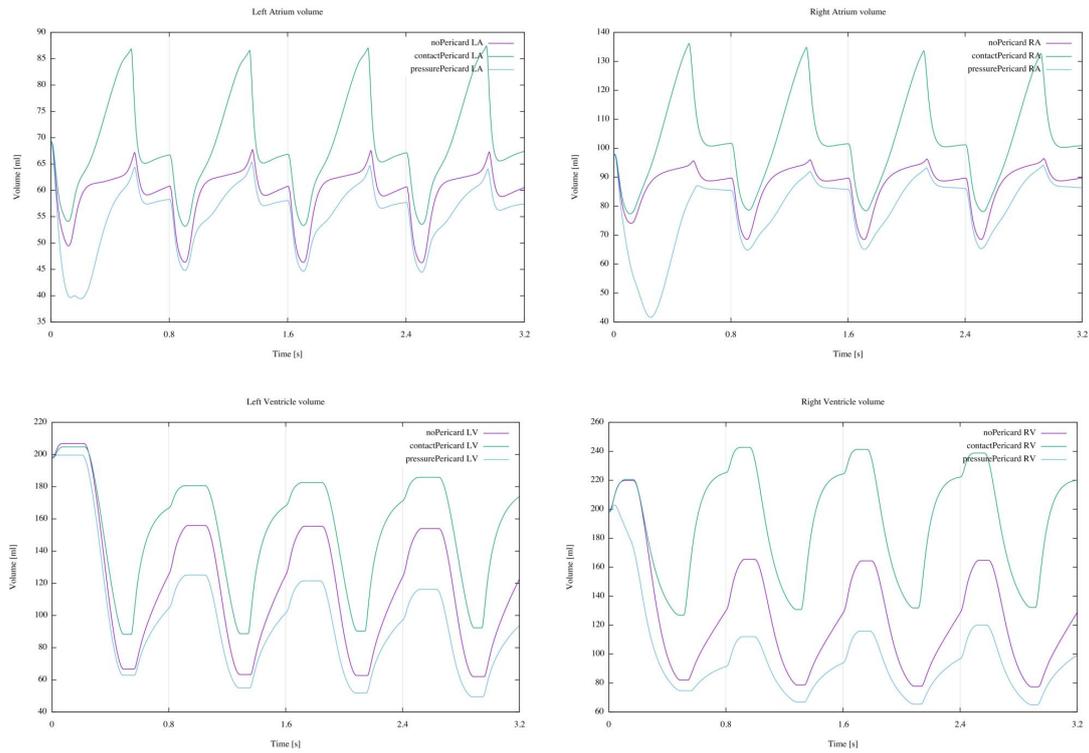


Figure 6.41.: Volume time courses for all four chambers comparing the three simulated setups, from Fensterseifer Schmidt (2017). The plots show left atrium (top left), right atrium (top right), left ventricle (bottom left) and right ventricle (bottom right) with pressure over time on the axes. Color labels are the same as in Figure 6.39.

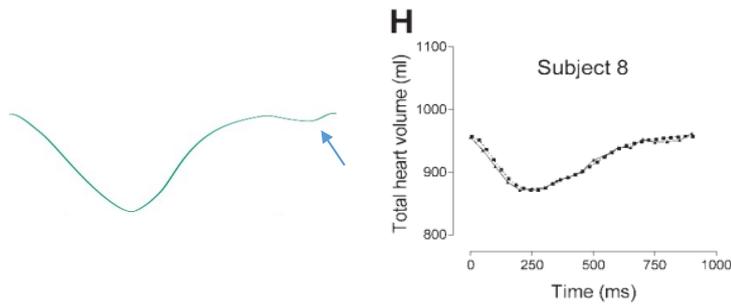


Figure 6.42.: Qualitative comparison between simulated total volume variation (pressure pericardium) and one from experimental data (Carlsson et al., 2004), matching the visually best subject. The peak due to atrial contraction is not visible in measurements (blue arrow). From Fensterseifer Schmidt (2017).

6. Improving the Physiological Behavior

an outer shape that is visually almost indistinguishable from a cine MRI measurement and vice versa. Both models reproduced the absolute change of total heart volume according to measurements given by Carlsson et al. (2004).

In the volume over time diagram for the zero-volume study from Figure 6.38, all simulations showed a distinct negative bump due to ventricular contraction at 0.5 s, 1.3 s, and so on, which also indicates the minimal achieved total heart volume as well. The simulation using the contact pericardium however additionally showed a second, smaller negative bump due to the atrial contraction, which was only visible with the pressure pericardium and only for the smaller values of $V_{tot,0}$. In case of using a larger $V_{tot,0}$, the total heart volume actually increased during atrial contraction. For the pressure pericardium, the total heart volume at beginning of atrial contraction is approximately the same as at the beginning of ventricular contraction. In the available measurement data from Carlsson et al. (2004) however, the atrial contraction can not be seen at all in any of the 8 human subjects (see Figure 6.42).

Looking at the time course of ventricular relaxation in Figure 6.39, the contact pericardium shows its steepest section in mid-relaxation which flattens out towards ending of ventricular relaxation. The pressure pericardium shows a similar decreasing steepness from mid to end of ventricular relaxation as well, except that the steepest part is at the beginning. In case of the simulation without pericardium modeling, the behavior is different, the time course is an almost linear one and extends even over the atrial contraction. An explanation for this different behavior is as follows. During contraction, the dominant forces are the active forces, which are the same in all three cases. During relaxation, active forces are absent, and the time course is dominated by passive forces releasing previously stored energy, the circulatory system, and forces due to the pericardium system. At beginning of relaxation, the mesh representing the surrounding tissue in the contact pericardium is considerably deformed, which pulls the epicardium outwards, hence resulting in an initially quick increase of ventricular volume. Towards the ending of relaxation, the deformation of surrounding tissue gets negligible and its influence becomes small. With the pressure pericardium, the formula yields a strictly positive pressure throughout the whole cardiac cycle for any chosen value of $V_{tot,0}$. This creates a permanently pushing compression effect due to the pericardium, which is increasing with ongoing relaxation. That effect is visible by an overall smaller total heart volume of the simulations with pressure pericardium and a quicker flattening of the volume curve during relaxation. The simulation without pericardium modeling are lacking any epicardial forces and expose fully the interplay between releasing stored passive energy and the circulatory system. Since the myocardium underwent a preloading procedure that reduces the overall volume during initialization, and this simulation misses the outward pulling effect of the contact pericardium, the overall volume is situated in-between contact pericardium and pressure pericardium.

Another effect of a pericardium model that was already observed in the zero-volume study is the restriction of the total heart volume change in both pericardium models. Although the ventricular volumes change still by more than 50%, the total heart volume change reduces from 19% to around 13%. As consequence, the atrial volumes increase considerably during ventricular contraction for both models. This becomes especially

6. Improving the Physiological Behavior

visible for the contact pericardium in the pV loops in Figure 6.40 and the chamber volumes in Figure 6.41: The volume change in both ventricles is about the same as without pericardium, but both atrial volumes change by about twice the amplitude. In case of the pressure pericardium, the situation is similar: Although atrial volume change looks about the same as without pericardium, the relationship between atrial and ventricular volume change is increased, since both ventricular and overall volume are generally smaller due to the models permanently applied positive pressure. In Figure 6.40, both atria show a larger volume change in the simulations with a pericardium model compared to the simulation without pericardium, thereby the pericardium supports the refilling of the atria with blood during ventricular contraction.

Due to previous investigations during development of circulatory system model in Section 6.3, one expected effect was to see a double loop structure in the atrial pV loops with the pericardium models but not without. Figure 6.40 shows such a double loop structure for the contact pericardium and the pressure pericardium, that is observed in measurements as well (compare e. g. Figure 6.34 and references therein). Surprisingly, all three simulations showed such a figure-eight loop, which is different from the results obtained during development of the circulatory system model in Section 6.3. Possible reasons are purely speculative: For this study, the myocardium mesh was replaced with a different one with slightly more tetrahedra and a better mesh quality. This was necessary to obtain more robust contact pericardium, although both meshes represent the same heart segmented from the same DICOM data set from the very same patient. Other differences are different passive material parameters and a different, now length-dependent active contraction model. This is an unfortunate point that seems to indicate an unwanted strong dependency of the results on the meshing strategy or some part of the modeling and should be considered in further investigations.

Regarding the behavior of isolated pericardium and surrounding tissue (without heart), Freeman and LeWinter (1984); Freeman and Little (1986) observed an exponential relationship between applied pressure and volume response, with a permanently positive pressure during the whole cardiac cycle. A technical limitation for the three-dimensional contact pericardium is the necessity to start the simulations in equilibrium-of-(zero)-forces state. As direct consequence, a pressure of zero already yields the end-diastolic pericardial volume, and even needs a negative pressure to obtain the smaller end-systolic volume. That means that the surrounding pericardium mesh of the contact pericardium pulls the myocardial wall outwards, with increasing amplitude during contraction. Due to its design, the pressure pericardium in contrast shows an exponential pressure-volume relationship similar to the one observed in the measurements by Freeman and Little (1986), which it is based on.

It is worth noting, that the expected pressures due to the pericardium (≈ 10 mmHg) are about in the same range as the lung pressure (≈ 10 mmHg) and atrial pressure (≈ 15 mmHg), but a magnitude of order smaller than ventricular blood pressure (≈ 120 mmHg). So effects of a pericardium modeling are expected to be mostly seen in the atria and less so in the ventricles. The similarity to lung pressure might open the possibility to gain insights from in-vivo measurements, by investigating the pericardium behavior at the same patient in inhaled and in exhaled lung state (“Valsalva maneuver”).

6.4.5. Conclusion

During realization of the project, it turned out that the two models showed a somehow opposing behavior. The results from the first study to find the zero-volume value for the pressure pericardium already showed in the different volume levels that the two models would be difficult to compare. Due to modeling implications, the two simulations start with different initial volumes and the force directions on the epicardial surface are different over the heart cycle. Even though the contact model with its three-dimensional design has the potential of showing a “locally more correct” behavior, there should not be such a great contradicting qualitative behavior. At least on the global level, averaging effects should allow the three-dimensional contact pericardium model to reduce to something that is at least qualitatively capable to be similar to the one-dimensional pressure pericardium model.

With regards to application in a fully coupled heart model, the pressure pericardium currently seems to be the better choice. The usual circulatory system models interact only via the chambers’ pressures and volumes with the FEM model, so these are the important parameters that need to fit. Unless the contact pericardium is able to reproduce isolated pressure-volume relation from measurements, it is difficult to recommend it. If only looking at the outcome of the combined model, it is easily possible to get a wrong combination of parameters that seem to fit. Missing preloading, a non-obvious implementation, and difficult-to-set-up simulations make it not a good choice for coupled heart simulations. One big advantage of the contact model is that it allows to look at local areas of interest. Possible applications are given by observing local stretch values, e. g. to identify regions of large deformation that might result in abnormal trigger points on the electrophysiological level. Another application is to simulate effects of stiffer regions in the hearts surroundings, e. g. partially stiffer tissue near the spine or the liver. In its current state however, the contact pericardium lacks validation on the local level, does not reproduce the pressure-volume responses of the isolated pericardium, and even inverts the force directions from a model that does so (pressure pericardium). In sum, these points add a big question mark behind any conclusions obtained from simulations with such model.

6.4.6. Outlook

In future works, two major tasks need to be addressed. First, more investigations on the experimental side are necessary to clarify the physiological behavior by measurements. The two modeling approaches showed a completely different behavior. Alone from the currently available literature (Freeman and LeWinter, 1984; Freeman and Little, 1986) it is not absolutely clarified which behavior is correct. The two modeling approaches are not contradicting by nature, however the used parameters imposed such a behavior. Better measurement data is needed to get a clear knowledge of the global behavior, both of the isolated and then the non-isolated pericardium, to decide if the pericardium has a pulling or pushing effect during ventricular contraction, and to develop the two models towards such a behavior. Ex-vivo experiments that are limited to animal experiments, could

6. Improving the Physiological Behavior

measure forces/pressures exposed by the implanted and explanted pericardium, including the important total heart volume of the same subject to decide if the pericardium is “pushing or pulling”. In-vivo experiments with human beings are mostly limited to non-invasive observation of the heart motion embedded in the pericardial sac and surrounding tissue. These would necessarily involve inverse methods to estimate parameters for the two pericardium models. However, even in the remainder part of heart modeling (passive tissue, contractile behavior, circulatory system) a large number of model parameters exist with considerable uncertainties. At their current state of accuracy this would make most approaches non-practical by a hard-to-estimate credibility of such a studies outcome. A promising approach involving human subjects is to investigate and recreate the behavior of patients with congenital absence of the pericardium Topilsky et al. (2010), which could give more insight in the interaction with the surrounding tissues with and without the pericardium. A second approach with human subjects is to investigate the behavior e. g. of atrial pressures during a “Valsalva maneuver” in inhaled and exhaled lung state, which would involve modeling of the heart embedded in a torso including lungs.

The second task is to improve the comparability between the two models, which could be done independently from the first task. The two models imply very different pericardial pressures at the beginning of the systole, thereby the initial volume is smaller for the pressure pericardium than for the contact pericardium. The strictly positive exponential formulation of the pressure pericardium however does not allow a zero or even negative pressure at all. Since the total heart volume only changes by around 10% throughout a cardiac cycle (Carlsson et al., 2004), the relevant part of volumes of the exponential function is small and can be linearized, even with error quantification (Taylor remainder formula). The most straightforward approach therefore would be to introduce a reduced linearized pericardium model. The task is to modify the pressure pericardium to allow a zero pressure at end-diastole and use a “slope” factor to describe the pressure-volume relationship throughout the cardiac cycle. The physiological meaning of that slope factor is directly related to the compliance of the pericardium and its surroundings. When restricting focus to that specific part of the pressure-volume plots from Freeman and Little (1986), the compliance is reasonably constant and a linear fit approximates the data points as good as an exponential fit. The difference between such a fit and the current approach is well below the fitting accuracy of the originally used exponential approximation to measurement data, but would greatly facilitates parameter estimation and it would allow the zero-pressure. By choosing a positive or negative slope, such a model is able to adapt to both behaviors and that factor could even be chosen to exactly reproduce the pressure volume response from simulations of the isolated contact pericardium (“surrogate model”). A study with same initial volume and same change of total heart volume would allow to decide which local benefits can be expected from a three-dimensional model over a zero-dimensional one, as differences in simulation outcome could be related clearly to local versus non-local modeling.

The most difficult part and the one with the most uncertain benefit is to improve the contact pericardium model. In its present state, it shows a reasonable change of total heart volume, but lacks to recreate the behavior of an isolated pericardium with surrounding tissue from Freeman and Little (1986). The most critical part of its modeling

6. *Improving the Physiological Behavior*

is the lack of preloading, which however is mandatory to create simulations with a pushing effect of the three-dimensional pericardium during contraction. In case of a preloaded pericardium, no globally stress-free state would exist, which makes a whole new framework design necessary that allows independent model initialization of parts of the mesh (for pericardium/surrounding tissue and for the actual heart). Stress of the individual parts still must fit together to start the simulations in equilibrium of forces. Besides the preloading part, work is needed on the modeling factors thickness of the pericardium mesh and passive material properties in order to get a better understanding of their influence on and relationship to physiological parameters. This on the other hand would facilitate future model parametrization towards a specific patient, e. g. by using similar methods as for ventricular passive parameter estimation done in Section 6.1. The outcome would be a universally applicable contact pericardium model that is able to fulfill all available global measurement data while showing possible local effects. Once better suitable measurement data is available, these could serve as a better modeling target for the two approaches and once their comparability is given, a purposeful development towards the measured behavior in terms of surface force direction, total volume change and isolated/non-isolated pressure-volume response becomes an easy task.

While the pressure pericardium actually was created to reproduce measurement data, it can also be interpreted differently, as a penalty formulation like it is used e. g. in mathematical optimization: A pressure onto the outer surface is used to influence the amount of volume change during the cardiac cycle. In case of the proposed linearized pressure pericardium, the penalty term is a simple (linear) distance function, and the penalty weight a compliance that can be determined from a simulation study to create a certain amount of volume change.

7. Summary and Outlook

In this thesis, a multi-component computational model of the human heart was created. The first part of this thesis starts with the general fundamentals of physiology and mathematical modeling and finishes in a description of the implementation of the developed software framework. In the second part, all the novelties introduced by this thesis are presented. The heart comprises of a high resolution finite element mesh, contains modeling of passive and active tissue behavior, a blood pressure model and a model of the surrounding pericardium and the area it is embedded in. Each model was verified to comply with widely available measurement data and suitable parameters are presented.

The passive model of myocardial tissue is the most essential part to fit since its parameters can be investigated independently but greatly influence the experiments to determine parameters for the other modeling components. The passive model parameters were verified to recreate a specific pressure-volume curve under inflation of the left ventricle. For that, an optimization procedure was used together with parameter reduction to obtain the intended passive behavior. As side effect, the fiber strain increased compared to previous works and now is closer to the physically expected strain.

Two different active models for the development of myocardial contraction force were implemented. One uses a prescribed time course of the developed force that can be delayed according to a map of local activation time and targets at creating a specific varying compliance of the ventricles. The other one uses a prescribed time course of the calcium transient, that is used together with the current fiber strain to dynamically compute the currently developed force in realtime. In a previous work, a precomputed time course was used, that was obtained from a complicated procedure that was not reproducible at the time of the current work.

A closed-loop model of the cardiovascular system was developed to compute blood pressure as boundary conditions in the four chambers. A representative electrical circuit contains systemic and pulmonary circulation and the elastic and resistive properties of the respective parts. In contrast to previous implementations, this approach ensures blood conservation, allows multiple heart beats, interaction between the chambers and overall a different quality of investigations. With suitable parameters obtained from an extensive literature research, the model is able to produce realistic time courses for blood pressures and ejection volumes, that converge to quasi-static states over multiple heart beats.

Concerning the heart-surrounding, two different models for the pericardium were implemented and compared. The complex contact pericardium uses a second, additional finite element mesh and minimizes the surface distance between both meshes. The simpler pressure pericardium uses a prescribed exponential pressure-volume relationship based on measurements and acts as epicardial surface force, too. Its source code is very similar in

7. Summary and Outlook

structure to that of a blood pressure model and therefore straightforward to implement and adds no additional overhead regarding mesh generation and computation time. While the contact pericardium shows the rather static outer shape during the heart beat, that can be observed in cine MRI sequences as well, the pressure pericardium produces the typical figure-8 loops that are characteristic for atrial pressure-volume loop diagrams. The isolated behavior of the two models however seems to contradict each other and needs further clarification.

One important factor for the success of a finite element simulation is the mesh choice. Due to the high computing time requirements of such a model, most models in literature are limited to a rather coarse mesh with only one to two elements transmurally, throughout the left ventricular wall. A mesh resolution study was conducted by inflating an ellipsoidal left ventricle, with resolutions up to 16 elements transmurally obtained from a fully-automated mesh generation. The results indicate that two second-order elements transmurally are at most justifiable when restricting the evaluation to volume-integral-type values like the enclosed endocardial volume (e.g. chamber volume). Regarding surface and apex positions however, two transmural elements are not sufficient since its locations still change significantly between four and eight elements (e.g. location of a catheter tip, or contact pericardium).

The other important factor to successfully conduct such a simulation study is performance of the software implementation. An efficient parallelization was realized which showed its scalability on a high performance computing cluster with up to 256 computing cores. The limiting factor is the number of elements per computing core, and memory bandwidth. A high resolution mesh scales almost linearly as long as there are roughly above 10,000 elements per core, and computation time profits significantly by distributing a certain number of computing threads to multiple nodes, that have their own, separate memory – compared to using multiple cores on one node.

In future works, the decision between two main directions will be necessary. The first one is to scale up the number of simulated patients. That requires a robust modeling setup, which naturally favors simpler models with easy to determine patient-specific parameters. The biggest hurdle in that direction certainly is the mesh generation, which currently demands numerous hours of work and a precious amount of knowledge to circumnavigate numerous pitfalls (mesh resolution, fitting surfaces) before obtaining a suitable simulation-ready mesh. Depending on the research question, abandoning parts of the modeling features might provide one viable solution, deforming a template mesh another. The second main direction is to improve the precision of the simulations. Room for improvements can be found almost everywhere: significantly higher resolution meshes, electrophysiology-based electrical excitation of the myofibers, or even fluid structure interaction inside the ventricles and the atria. An important step towards the simulation of highly complex systems with high resolution meshes was achieved in this thesis with a reliable parallelization of the simulation code that is able to run on a high performance computing cluster. Besides increasing modeling complexity, one rather simple but important aspect is the quality of the used parameters values. Part of that are models that have robustly determinable parameters, but also experiments that clarify the right question. It is still not well understood why the pericardial sac exist, another

7. Summary and Outlook

question is the presence of fiber twist. Since experiments on human subjects are difficult to realize in this area, experiments on animals with similar physiology in the specific area of question will be necessary. Another aspect to be kept in mind is, that with increasing modeling complexity the implementation gets more extensive and error prone. Possible errors in the modeling setup are becoming increasingly more difficult to detect, and it is even harder to locate and solve them. Finally, the best advice is to identify the relevant effects and not to use more complexity than necessary to answer the specific research question.

List of Publications with Contributions of the Author

Robin Andlauer, Gunnar Seemann, Lukas Baron, Olaf Dössel, Peter Kohl, Pyotr Platonov, and Axel Loewe. Influence of left atrial size on p-wave morphology: differential effects of dilation and hypertrophy. *Europace*, 20:iii36–iii44, 11 2018. doi: 10.1093/europace/euy231.

Lukas Baron, Thomas Fritz, Gunnar Seemann, and Olaf Dössel. Sensitivity study of fiber orientation on stroke volume in the human left ventricle. *Computing in Cardiology*, 41: 681–684, 01 2014.

Ekaterina Kovacheva, Lukas Baron, Olaf Doessel, and Axel Loewe. Electro-mechanical delay in the human heart: A study on a simple geometry. 12 2018. doi: 10.22489/CinC.2018.199.

Sander Land, Viatcheslav Gurev, Sander Arens, Christoph Augustin, Lukas Baron, Robert Blake, Chris Bradley, Sebastián Castro, Andrew Crozier, Marco Favino, Thomas Fastl, Thomas Fritz, Hao Gao, Alessio Gizzi, Boyce Griffith, Daniel Hurtado, Rolf Krause, Xiaoyu Luo, Martyn Nash, and Steven Niederer. Verification of cardiac mechanics software: Benchmark problems and solutions for testing active and passive material behaviour. *Proceedings of The Royal Society A Mathematical Physical and Engineering Sciences*, 471, 12 2015. doi: 10.1098/rspa.2015.0641.

Armin Müller, Ekaterina Kovacheva, Steffen Schuler, Olaf Dössel, and Lukas Baron. Effects of local activation times on the tension development of human cardiomyocytes in a computational model. *Current Directions in Biomedical Engineering*, 4:101–104, 09 2018. doi: 10.1515/cdbme-2018-0026.

Bibliography

- H. Ashikaga, B. A. Coppola, K. G. Yamazaki, F. J. Villarreal, J. H. Omens, and J. W. Covell. Changes in regional myocardial volume during the cardiac cycle: implications for transmural blood flow and cardiac structure. *American Journal of Physiology-Heart and Circulatory Physiology*, 295(2):H610–H618, Aug. 2008. ISSN 0363-6135, 1522-1539. doi: 10.1152/ajpheart.00107.2008. URL <https://www.physiology.org/doi/10.1152/ajpheart.00107.2008>.
- C. M. Augustin, A. Neic, M. Liebmann, A. J. Prassl, S. A. Niederer, G. Haase, and G. Plank. Anatomically accurate high resolution modeling of human whole heart electromechanics: A strongly scalable algebraic multigrid solver method for nonlinear deformation. *Journal of Computational Physics*, 305:622–646, Jan. 2016. ISSN 00219991. doi: 10.1016/j.jcp.2015.10.045. URL <http://linkinghub.elsevier.com/retrieve/pii/S0021999115007226>.
- T. Belytschko, W. K. Liu, B. Moran, and K. I. Elkhodary. *Nonlinear Finite Elements for Continua and Structures*. John Wiley & Sons, 2014.
- A. Bishop, P. White, P. Oldershaw, R. Chaturvedi, C. Brookes, and A. Redington. Clinical application of the conductance catheter technique in the adult human right ventricle. *International Journal of Cardiology*, 58(3):211–221, Feb. 1997. ISSN 0167-5273, 1874-1754. doi: 10.1016/S0167-5273(96)02880-X. URL [http://www.internationaljournalofcardiology.com/article/S0167-5273\(96\)02880-X/abstract](http://www.internationaljournalofcardiology.com/article/S0167-5273(96)02880-X/abstract).
- Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam, 2017. URL <http://www.blender.org>.
- J. Bols, J. Degroote, B. Trachet, B. Verheghe, P. Segers, and J. Vierendeels. A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels. *Journal of Computational and Applied Mathematics*, 246:10–17, July 2013. ISSN 03770427. doi: 10.1016/j.cam.2012.10.034. URL <http://linkinghub.elsevier.com/retrieve/pii/S0377042712004815>.
- C. G. Broyden. A class of methods for solving nonlinear simultaneous equations. *Mathematics of Computation*, 19(92):577–593, 1965. ISSN 0025-5718. doi: 10.2307/2003941. URL <https://www.jstor.org/stable/2003941>.
- D. Burkhoff and J. V. Tyberg. Why does pulmonary venous pressure rise after onset of LV dysfunction: a theoretical analysis. *American Journal of Physiology-Heart and*

Bibliography

- Circulatory Physiology*, 265(5):H1819–H1828, Nov. 1993. ISSN 0363-6135, 1522-1539. doi: 10.1152/ajpheart.1993.265.5.H1819. URL <http://www.physiology.org/doi/10.1152/ajpheart.1993.265.5.H1819>.
- D. Burkhoff, I. Mirsky, and H. Suga. Assessment of systolic and diastolic ventricular properties via pressure-volume analysis: a guide for clinical, translational, and basic researchers. *American Journal of Physiology-Heart and Circulatory Physiology*, 289(2):H501–H512, Aug. 2005. ISSN 0363-6135. doi: 10.1152/ajpheart.00138.2005. URL <https://www.physiology.org/doi/abs/10.1152/ajpheart.00138.2005>.
- M. Carlsson, P. Cain, C. Holmqvist, F. Stahlberg, S. Lundback, and H. Arheden. Total heart volume variation throughout the cardiac cycle in humans. *American Journal of Physiology. Heart and Circulatory Physiology*, 287(1):H243–250, July 2004. ISSN 0363-6135. doi: 10.1152/ajpheart.01125.2003.
- M. L. Chuang, P. Gona, G. L. Hautvast, C. J. Salton, S. J. Blease, S. B. Yeon, M. Breeuwer, C. J. O'Donnell, and W. J. Manning. Left ventricular trabeculae and papillary muscles: Correlation with clinical and cardiac characteristics and impact on cardiovascular magnetic resonance measures of left ventricular anatomy and function. *JACC. Cardiovascular imaging*, 5(11):1115–1123, Nov. 2012. ISSN 1936-878X. doi: 10.1016/j.jcmg.2012.05.015. URL <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3502069/>.
- J. Chung and G. M. Hulbert. A Time Integration Algorithm for Structural Dynamics With Improved Numerical Dissipation: The Generalized-alpha Method. *Journal of Applied Mechanics*, 60(2):371–375, June 1993. ISSN 0021-8936. doi: 10.1115/1.2900803. URL <http://appliedmechanics.asmedigitalcollection.asme.org/article.aspx?articleid=1410995>.
- K. D. Costa, K. May-Newman, D. Farr, W. G. O'Dell, A. D. McCulloch, and J. H. Omens. Three-dimensional residual strain in midanterior canine left ventricle. *American Journal of Physiology-Heart and Circulatory Physiology*, 273(4):H1968–H1976, 1997.
- K. D. Costa, J. W. Holmes, and A. D. McCulloch. Modelling cardiac mechanical properties in three dimensions. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 359(1783):1233–1250, June 2001. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.2001.0828. URL <http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2001.0828>.
- M. Courtemanche, R. J. Ramirez, and S. Nattel. Ionic mechanisms underlying human atrial action potential properties: insights from a mathematical model. *American Journal of Physiology - Heart and Circulatory Physiology*, 275(1):H301–H321, July 1998. ISSN 0363-6135, 1522-1539. URL <http://ajpheart.physiology.org/content/275/1/H301>.
- E. S. Di Martino, C. Bellini, and D. S. Schwartzman. In vivo porcine left atrial wall stress: Computational model. *Journal of Biomechanics*, 44(15):2589–2594, 2011. ISSN 0021-9290.

Bibliography

- S. Dokos, B. H. Smaill, A. A. Young, and I. J. LeGrice. Shear properties of passive ventricular myocardium. *American Journal of Physiology - Heart and Circulatory Physiology*, 283(6):H2650–H2659, Dec. 2002. ISSN 0363-6135, 1522-1539. doi: 10.1152/ajpheart.00111.2002. URL <http://ajpheart.physiology.org/lookup/doi/10.1152/ajpheart.00111.2002>.
- A. Fensterseifer Schmidt. Implementation of a zero-dimensional pericardium model of the human heart. Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2017.
- J. J. Ferguson, M. J. Miller, J. M. Aroesty, P. Sahagian, W. Grossman, and R. G. McKay. Assessment of right atrial pressure-volume relations in patients with and without an atrial septal defect. *Journal of the American College of Cardiology*, 13(3):630–636, 1989.
- M. S. Firstenberg, N. L. Greenberg, N. G. Smedira, D. L. Prior, G. M. Scalia, J. D. Thomas, and M. J. Garcia. Doppler echo evaluation of pulmonary venous-left atrial pressure gradients: human and numerical model studies. *American Journal of Physiology-Heart and Circulatory Physiology*, 279(2):H594–H600, 2000.
- G. L. Freeman and M. M. LeWinter. Pericardial adaptations during chronic cardiac dilation in dogs. *Circulation Research*, 54(3):294–300, Mar. 1984. ISSN 0009-7330, 1524-4571. doi: 10.1161/01.RES.54.3.294. URL <http://circres.ahajournals.org/content/54/3/294>.
- G. L. Freeman and W. C. Little. Comparison of in situ and in vitro studies of pericardial pressure-volume relation in dogs. *American Journal of Physiology-Heart and Circulatory Physiology*, Aug. 1986. doi: 10.1152/ajpheart.1986.251.2.H421. URL <http://www.physiology.org/doi/abs/10.1152/ajpheart.1986.251.2.H421>.
- T. Fritz. *Biomechanical Modeling of the Human Heart - Modeling of the Ventricles, the Atria and the Pericardium and the Inverse Problem of Cardiac Mechanics*, volume 23 of *Karlsruhe Transactions on Biomedical Engineering*. KIT Scientific Publishing, Karlsruhe, 2015.
- T. Fritz, C. Wieners, G. Seemann, H. Steen, and O. Dössel. Simulation of the contraction of the ventricles in a human heart model including atria and pericardium: Finite element analysis of a frictionless contact problem. *Biomechanics and Modeling in Mechanobiology*, 13(3):627–641, June 2014. ISSN 1617-7959, 1617-7940. doi: 10.1007/s10237-013-0523-y. URL <http://link.springer.com/10.1007/s10237-013-0523-y>.
- Y.-c. Fung. *Foundations of solid mechanics*. Prentice-Hall, 1965.
- Y.-C. Fung and others. *Mechanical properties of living tissues*, volume 547. Springer, 1993.

Bibliography

- M. Galderisi. Diastolic dysfunction and diastolic heart failure: diagnostic, prognostic and therapeutic aspects. *Cardiovascular ultrasound*, 3(1):9, 2005.
- M. Genet, L. C. Lee, R. Nguyen, H. Haraldsson, G. Acevedo-Bolton, Z. Zhang, L. Ge, K. Ordovas, S. Kozerke, and J. M. Guccione. Distribution of normal human left ventricular myofiber stress at end diastole and end systole: a target for in silico design of heart failure treatments. *Journal of Applied Physiology*, 117(2):142–152, July 2014. ISSN 8750-7587, 1522-1601. doi: 10.1152/jappphysiol.00255.2014. URL <http://jap.physiology.org/cgi/doi/10.1152/jappphysiol.00255.2014>.
- T. Gerach, S. Schuler, J. Fröhlich, L. Lindner, E. Kovacheva, R. Moss, E. M. Wülfers, G. Seemann, C. Wieners, and A. Loewe. Electro-mechanical whole-heart digital twins: A fully coupled multi-physics approach. *Mathematics*, 9(11):1247, 2021.
- C. Geuzaine and J.-F. Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009. URL <http://onlinelibrary.wiley.com/doi/10.1002/nme.2579/full>.
- J. M. Guccione, A. D. McCulloch, and L. K. Waldman. Passive Material Properties of Intact Ventricular Myocardium Determined From a Cylindrical Model. *Journal of Biomechanical Engineering*, 113(1):42, 1991. ISSN 01480731. doi: 10.1115/1.2894084. URL <http://Biomechanical.asmedigitalcollection.asme.org/article.aspx?articleid=1398519>.
- J. M. Guccione, K. D. Costa, and A. D. McCulloch. Finite element stress analysis of left ventricular mechanics in the beating dog heart. *Journal of Biomechanics*, 28(10):1167–1177, Oct. 1995. ISSN 0021-9290. doi: 10.1016/0021-9290(94)00174-3. URL <http://www.sciencedirect.com/science/article/pii/0021929094001743>.
- V. Gurev, P. Pathmanathan, J.-L. Fattebert, H.-F. Wen, J. Magerlein, R. A. Gray, D. F. Richards, and J. J. Rice. A high-resolution computational model of the deforming human heart. *Biomechanics and Modeling in Mechanobiology*, 14(4):829–849, Aug. 2015. ISSN 1617-7959, 1617-7940. doi: 10.1007/s10237-014-0639-8. URL <http://link.springer.com/10.1007/s10237-014-0639-8>.
- M. Hadjicharalambous, R. Chabiniok, L. Asner, E. Sammut, J. Wong, G. Carr-White, J. Lee, R. Razavi, N. Smith, and D. Nordsletten. Analysis of passive cardiac constitutive laws for parameter estimation using 3d tagged mri. *Biomechanics and modeling in mechanobiology*, 14(4):807–828, 2015.
- J. E. Hall. *Guyton and Hall textbook of medical physiology e-Book*. Elsevier Health Sciences, 2015.
- R. Hauptmann, S. Doll, M. Harnau, and K. Schweizerhof. Solid-shell’elements with linear and quadratic shape functions at large deformations with nearly incompressible materials. *Computers & Structures*, 79(18):1671–1685, 2001.

Bibliography

- M. Heron. Deaths: Leading Causes for 2017. 68(6):77, June 2019.
- B. S. Hertzberg, B. S. Mahony, and J. D. Bowie. First trimester fetal cardiac activity. Sonographic documentation of a progressive early rise in heart rate. *Journal of Ultrasound in Medicine*, 7(10):573–575, 1988. ISSN 1550-9613. doi: 10.7863/jum.1988.7.10.573. URL <https://onlinelibrary.wiley.com/doi/abs/10.7863/jum.1988.7.10.573>.
- G. A. Holzapfel and R. W. Ogden. Constitutive modelling of passive myocardium: a structurally based framework for material characterization. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 367(1902): 3445–3475, Sept. 2009. ISSN 1364-503X, 1471-2962. doi: 10.1098/rsta.2009.0091. URL <http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2009.0091>.
- T. J. R. Hughes, K. S. Pister, and R. L. Taylor. Implicit-explicit finite elements in nonlinear transient analysis. *Computer Methods in Applied Mechanics and Engineering*, 17-18:159–182, Jan. 1979. ISSN 0045-7825. doi: 10.1016/0045-7825(79)90086-0. URL <http://www.sciencedirect.com/science/article/pii/0045782579900860>.
- W. Jakob, M. Tarini, D. Panozzo, and O. Sorkine-Hornung. Instant field-aligned meshes. *ACM Trans. Graph.*, 34(6):189–1, 2015.
- R. Kerckhoffs, A. McCulloch, J. Omens, and L. Mulligan. Effect of pacing site and infarct location on regional mechanics and global hemodynamics in a model based study of heart failure. *Functional Imaging and Modeling of the Heart*, pages 350–360, 2007a. URL <http://www.springerlink.com/index/01n10x42k7524721.pdf>.
- R. C. P. Kerckhoffs, M. L. Neal, Q. Gu, J. B. Bassingthwaighte, J. H. Omens, and A. D. McCulloch. Coupling of a 3d Finite Element Model of Cardiac Ventricular Mechanics to Lumped Systems Models of the Systemic and Pulmonic Circulation. *Annals of Biomedical Engineering*, 35(1):1–18, Jan. 2007b. ISSN 0090-6964, 1573-9686. doi: 10.1007/s10439-006-9212-7. URL <https://link.springer.com/article/10.1007/s10439-006-9212-7>.
- R. Klabunde. *Cardiovascular physiology concepts*. Lippincott Williams & Wilkins, 2011.
- S. Klotz, I. Hay, M. L. Dickstein, G.-H. Yi, J. Wang, M. S. Maurer, D. A. Kass, and D. Burkhoff. Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application. *American Journal of Physiology - Heart and Circulatory Physiology*, 291(1):H403–H412, July 2006. ISSN 0363-6135, 1522-1539. doi: 10.1152/ajpheart.01240.2005. URL <http://ajpheart.physiology.org/content/291/1/H403>.
- S. Klotz, M. L. Dickstein, and D. Burkhoff. A computational method of prediction of the end-diastolic pressure–volume relationship by single beat. *Nature Protocols*, 2(9): 2152–2158, Sept. 2007. doi: 10.1038/nprot.2007.270.

Bibliography

- M. Kotzeva. *Eurostat regional yearbook 2019 edition*. Statistical books. Luxembourg: Publications Office of the European Union, June 2019. ISBN 978-92-76-03505-3.
- S. Land and S. A. Niederer. Influence of atrial contraction dynamics on cardiac function: Influence of atrial contraction dynamics on cardiac function. *International Journal for Numerical Methods in Biomedical Engineering*, page e2931, Nov. 2017. ISSN 20407939. doi: 10.1002/cnm.2931. URL <http://doi.wiley.com/10.1002/cnm.2931>.
- S. Land, V. Gurev, S. Arens, C. M. Augustin, L. Baron, R. Blake, C. Bradley, S. Castro, A. Crozier, M. Favino, T. E. Fastl, T. Fritz, H. Gao, A. Gizzi, B. E. Griffith, D. E. Hurtado, R. Krause, X. Luo, M. P. Nash, S. Pezzuto, G. Plank, S. Rossi, D. Ruprecht, G. Seemann, N. P. Smith, J. Sundnes, J. J. Rice, N. Trayanova, D. Wang, Z. Jenny Wang, and S. A. Niederer. Verification of cardiac mechanics software: benchmark problems and solutions for testing active and passive material behaviour. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 471(2184):20150641, Dec. 2015. ISSN 1364-5021, 1471-2946. doi: 10.1098/rspa.2015.0641. URL <http://rspa.royalsocietypublishing.org/lookup/doi/10.1098/rspa.2015.0641>.
- S. Land, S.-J. Park-Holohan, N. P. Smith, C. G. dos Remedios, J. C. Kentish, and S. A. Niederer. A model of cardiac contraction based on novel measurements of tension development in human cardiomyocytes. *Journal of Molecular and Cellular Cardiology*, 106:68–83, May 2017. ISSN 00222828. doi: 10.1016/j.yjmcc.2017.03.008. URL <http://linkinghub.elsevier.com/retrieve/pii/S0022282817300639>.
- A. Lassoued and O. Boubaker. Modeling and control in physiology. In *Control Theory in Biomedical Engineering*, pages 3–42. Elsevier, 2020.
- I. J. LeGrice, B. H. Smaill, L. Z. Chai, S. G. Edgar, J. B. Gavin, and P. J. Hunter. Laminar structure of the heart: ventricular myocyte arrangement and connective tissue architecture in the dog. *American Journal of Physiology - Heart and Circulatory Physiology*, 269(2):H571–H582, Aug. 1995. ISSN 0363-6135, 1522-1539. URL <http://ajpheart.physiology.org/content/269/2/H571>.
- J. Lumens, T. Delhaas, B. Kirn, and T. Arts. Three-Wall Segment (TriSeg) Model Describing Mechanics and Hemodynamics of Ventricular Interaction. *Annals of Biomedical Engineering*, 37(11):2234–2255, Nov. 2009. ISSN 0090-6964, 1573-9686. doi: 10.1007/s10439-009-9774-2. URL <http://link.springer.com/10.1007/s10439-009-9774-2>.
- Y. Matsuda, Y. Toma, H. Ogawa, M. Matsuzaki, K. Katayama, T. Fujii, F. Yoshino, K. Moritani, T. Kumada, and R. Kusukawa. Importance of left atrial function in patients with myocardial infarction. *Circulation*, 67(3):566–571, 1983.
- J. R. Mitchell and J.-J. Wang. Expanding application of the Wiggers diagram to teach cardiovascular physiology. *Advances in Physiology Education*, 38(2):170–175, June 2014. ISSN 1043-4046. doi: 10.1152/advan.00123.2013. URL <https://www.physiology.org/doi/abs/10.1152/advan.00123.2013>.

Bibliography

- C. Moler et al. *MATLAB users' guide*. University of New Mexico, 1982.
- A. Müller. Parallelisierung und implizite Kopplung von dehnungsabhängigen Modellen der aktiven Kraft zur Anwendung auf menschlichen Herzmuskelzellen. Master's thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2017.
- J. P. Mynard. *Computer modelling and wave intensity analysis of perinatal cardiovascular function and dysfunction*. PhD thesis, University of Melbourne, 2011. URL <http://jupiter.its.unimelb.edu.au/handle/11343/36318>.
- J. P. Mynard and J. J. Smolich. One-Dimensional Haemodynamic Modeling and Wave Dynamics in the Entire Adult Circulation. *Annals of Biomedical Engineering*, 43(6): 1443–1460, June 2015. ISSN 0090-6964, 1573-9686. doi: 10.1007/s10439-015-1313-8. URL <http://link.springer.com/10.1007/s10439-015-1313-8>.
- A. Nikou, S. M. Dorsey, J. R. McGarvey, J. H. Gorman, J. A. Burdick, J. J. Pilla, R. C. Gorman, and J. F. Wenk. Effects of Using the Unloaded Configuration in Predicting the In Vivo Diastolic Properties of the Heart. *Computer methods in biomechanics and biomedical engineering*, 19(16):1714–1720, Dec. 2016. ISSN 1025-5842. doi: 10.1080/10255842.2016.1183122. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5278778/>.
- J.-i. Okada, T. Washio, M. Nakagawa, M. Watanabe, Y. Kadooka, T. Kariya, H. Yamashita, Y. Yamada, S.-i. Momomura, R. Nagai, et al. Multi-scale, tailor-made heart simulation can predict the effect of cardiac resynchronization therapy. *Journal of molecular and cellular cardiology*, 108:17–23, 2017.
- J. H. Omens, D. A. MacKenna, and A. D. McCulloch. Measurement of strain and analysis of stress in resting rat left ventricular myocardium. *Journal of Biomechanics*, 26(6): 665–676, 1993.
- OpenStax. *Anatomy and Physiology*. OpenStax, Houston, TX, 2017.
- V. Rajagopal, J.-H. Chung, D. Bullivant, P. M. F. Nielsen, and M. P. Nash. Determining the finite elasticity reference state from a loaded configuration. *International Journal for Numerical Methods in Engineering*, 72(12):1434–1451, Dec. 2007. ISSN 00295981, 10970207. doi: 10.1002/nme.2045. URL <http://doi.wiley.com/10.1002/nme.2045>.
- A. N. Redington, H. H. Gray, M. E. Hodson, M. L. Rigby, and P. J. Oldershaw. Characterisation of the normal right ventricular pressure-volume relation by biplane angiography and simultaneous micromanometer pressure measurements. *Heart*, 59(1): 23–30, Jan. 1988. ISSN 1355-6037, 1468-201X. doi: 10.1136/hrt.59.1.23. URL <http://heart.bmj.com/content/59/1/23>.
- L. M. Rios and N. V. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):

Bibliography

- 1247–1293, July 2013. ISSN 0925-5001, 1573-2916. doi: 10.1007/s10898-012-9951-y. URL <http://link.springer.com/10.1007/s10898-012-9951-y>.
- E. K. Rodriguez, J. H. Omens, L. K. Waldman, and A. D. McCulloch. Effect of residual stress on transmural sarcomere length distributions in rat left ventricle. *American Journal of Physiology-Heart and Circulatory Physiology*, 264(4):H1048–H1056, Apr. 1993. ISSN 0363-6135. doi: 10.1152/ajpheart.1993.264.4.H1048. URL <http://www.physiology.org/doi/abs/10.1152/ajpheart.1993.264.4.H1048>.
- F. B. Sachse, K. G. Glänzel, and G. Seemann. Modeling of protein interactions involved in cardiac tension development. *International Journal of Bifurcation and Chaos*, 13(12):3561–3578, Dec. 2003. ISSN 0218-1274. doi: 10.1142/S0218127403008855. URL <http://www.worldscientific.com/doi/abs/10.1142/S0218127403008855>.
- W. P. Santamore and D. Burkhoff. Hemodynamic consequences of ventricular interaction as assessed by model analysis. *American Journal of Physiology-Heart and Circulatory Physiology*, 260(1):H146–H157, Jan. 1991. ISSN 0363-6135. doi: 10.1152/ajpheart.1991.260.1.H146. URL <http://www.physiology.org/doi/abs/10.1152/ajpheart.1991.260.1.H146>.
- H. Schmid, P. O’Callaghan, M. P. Nash, W. Lin, I. J. LeGrice, B. H. Smaill, A. A. Young, and P. J. Hunter. Myocardial material parameter estimation: A non-homogeneous finite element study from simple shear tests. *Biomechanics and Modeling in Mechanobiology*, 7(3):161–173, June 2008. ISSN 1617-7959, 1617-7940. doi: 10.1007/s10237-007-0083-0. URL <http://link.springer.com/10.1007/s10237-007-0083-0>.
- H. Schmid, Y. K. Wang, J. Ashton, A. E. Ehret, S. B. S. Krittian, M. P. Nash, and P. J. Hunter. Myocardial material parameter estimation: a comparison of invariant based orthotropic constitutive equations. *Computer Methods in Biomechanics and Biomedical Engineering*, 12(3):283–295, June 2009. ISSN 1025-5842. doi: 10.1080/10255840802459420. URL <http://dx.doi.org/10.1080/10255840802459420>.
- R. F. Schmidt, F. Lang, and M. Heckmann. *Physiologie des Menschen - mit Pathophysiologie*. Springer, 2007. URL [//www.springer.com/de/book/9783540329107](http://www.springer.com/de/book/9783540329107).
- W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit*. Kitware, 4th ed. edition, 2006. ISBN 978-1-930934-19-1.
- S. Schuler. Developing and coupling a lumped parameter model of the closed loop human vascular system to a model of cardiac mechanics. Master’s thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2016.
- M. Schünke, E. Schulte, U. Schumacher, M. Voll, and K. Wesker, editors. *Prometheus - Lernatlas der Anatomie: Innere Organe*. Georg Thieme Verlag, Stuttgart [u.a.], 2009. ISBN 978-3-13-139532-0.

Bibliography

- M. Sellier. An iterative method for the inverse elasto-static problem. *Journal of Fluids and Structures*, 27(8):1461–1470, Nov. 2011. ISSN 08899746. doi: 10.1016/j.jfluidstructs.2011.08.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S088997461100123X>.
- H. Senzaki, C. H. Chen, and D. A. Kass. Single-beat estimation of end-systolic pressure-volume relation in humans. a new method with the potential for noninvasive application. *Circulation*, 94(10):2497–2506, 1996. ISSN 0009-7322.
- H. Si. TetGen, a Delaunay-Based Quality Tetrahedral Mesh Generator. *ACM Trans. Math. Softw.*, 41(2):11:1–11:36, Feb. 2015. ISSN 0098-3500. doi: 10.1145/2629697. URL <http://doi.acm.org/10.1145/2629697>.
- N. Stergiopoulos, J. J. Meister, and N. Westerhof. Determinants of stroke volume and systolic and diastolic aortic pressure. *American Journal of Physiology-Heart and Circulatory Physiology*, 270(6):H2050–H2059, June 1996. ISSN 0363-6135. doi: 10.1152/ajpheart.1996.270.6.H2050. URL <https://www.physiology.org/doi/abs/10.1152/ajpheart.1996.270.6.H2050>.
- B. Stroustrup. *The C++ programming language*. Pearson Education, 2013.
- S. Sugiura, T. Washio, A. Hatano, J. Okada, H. Watanabe, and T. Hisada. Multi-scale simulations of cardiac electrophysiology and mechanics using the university of tokyo heart simulator. *Progress in biophysics and molecular biology*, 110(2-3):380–389, 2012.
- K. ten Tusscher, D. Noble, P.-J. Noble, and A. Panfilov. A model for human ventricular tissue. *American Journal of Physiology - Heart and Circulatory Physiology*, 286(4):H1573–H1589, Apr. 2004. ISSN 0363-6135, 1522-1539. doi: 10.1152/ajpheart.00794.2003. URL <http://ajpheart.physiology.org/content/286/4/h1573>.
- Y. Topilsky, N. Tabatabaei, W. K. Freeman, H. K. Saleh, H. R. Villarraga, and S. L. Mulvagh. Pendulum Heart in Congenital Absence of the Pericardium. *Circulation*, 121(10):1272–1274, Mar. 2010. ISSN 0009-7322, 1524-4539. doi: 10.1161/CIR.0b013e3181d73b9b. URL <https://www.ahajournals.org/doi/10.1161/CIR.0b013e3181d73b9b>.
- G. van Rossum. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.
- L. K. Waldman, Y. C. Fung, and J. W. Covell. Transmural myocardial deformation in the canine left ventricle. normal in vivo three-dimensional finite strains. *Circulation Research*, 57(1):152–163, July 1985. ISSN 0009-7330, 1524-4571. doi: 10.1161/01.RES.57.1.152. URL <https://www.ahajournals.org/doi/10.1161/01.RES.57.1.152>.
- J.-J. Wang, J. A. Flewitt, N. G. Shrive, K. H. Parker, and J. V. Tyberg. Systemic venous circulation. waves propagating on a windkessel: relation of arterial and venous windkessels to systemic vascular resistance. *American Journal of Physiology-Heart and Circulatory Physiology*, 290(1):H154–H162, 2006.

Bibliography

- V. Y. Wang, H. Lam, D. B. Ennis, B. R. Cowan, A. A. Young, and M. P. Nash. Modelling passive diastolic mechanics with quantitative mri of cardiac structure and function. *Medical image analysis*, 13(5):773–784, 2009.
- H. Watanabe, S. Sugiura, H. Kafuku, and T. Hisada. Multiphysics simulation of left ventricular filling dynamics using fluid-structure interaction finite element method. *Biophysical journal*, 87(3):2074–2085, 2004.
- T. Weimar, Y. Watanabe, T. Kazui, U. S. Lee, M. R. Moon, R. B. Schuessler, and R. J. Damiano. Differential impact of short periods of rapid atrial pacing on left and right atrial mechanical function. *American Journal of Physiology-Heart and Circulatory Physiology*, 302(12):H2583–H2591, Apr. 2012. ISSN 0363-6135. doi: 10.1152/ajpheart.01170.2011. URL <http://www.physiology.org/doi/abs/10.1152/ajpheart.01170.2011>.
- N. Westerhof, J.-W. Lankhaar, and B. E. Westerhof. The arterial Windkessel. *Medical & Biological Engineering & Computing*, 47(2):131–141, Feb. 2009. ISSN 0140-0118, 1741-0444. doi: 10.1007/s11517-008-0359-2. URL <https://link.springer.com/article/10.1007/s11517-008-0359-2>.
- Wikimedia Commons. Blood Vessel — Wikipedia, The Free Encyclopedia, May 2016. URL https://commons.wikimedia.org/wiki/File:Circulatory_System_en.svg.
- J. Xi, P. Lamata, J. Lee, P. Moireau, D. Chapelle, and N. Smith. Myocardial transversely isotropic material parameter estimation from in-silico measurements based on a reduced-order unscented Kalman filter. *Journal of the Mechanical Behavior of Biomedical Materials*, 4(7):1090–1102, Oct. 2011. ISSN 17516161. doi: 10.1016/j.jmbbm.2011.03.018. URL <http://linkinghub.elsevier.com/retrieve/pii/S1751616111000634>.
- P. A. Yushkevich, J. Piven, H. C. Hazlett, R. G. Smith, S. Ho, J. C. Gee, and G. Gerig. User-guided 3d active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. *NeuroImage*, 31(3):1116–1128, July 2006. ISSN 1053-8119. doi: 10.1016/j.neuroimage.2006.01.015. URL <http://www.sciencedirect.com/science/article/pii/S1053811906000632>.
- W. A. Zoghbi, G. B. Habib, and M. A. Quinones. Doppler assessment of right ventricular filling in a normal population. comparison with left ventricular filling dynamics. *Circulation*, 82(4):1316–1324, 1990.