Interaction of elastomechanics and fluid dynamics in the human heart

Opportunities and challenges of light coupling strategies

Zur Erlangung des akademischen Grades eines

DOKTORS DER INGENIEURWISSENSCHAFTEN (Dr.-Ing.)

von der KIT-Fakultät für

Elektrotechnik und Informationstechnik

des Karlsruher Instituts für Technologie (KIT)

genehmigte

DISSERTATION

von

Jochen Brenneisen, M.Sc.

geb. in Lörrach

Tag der mündlichen Prüfung:	14. Februar 2023
Referent:	Prof. Dr. rer. nat. Olaf Dössel
Korreferent:	Prof. Dr.-Ing. Bettina Frohnapfel
Korreferent:	PD Dr.-Ing. Axel Loewe

Abstract

The human heart is the highly complex core part of the cardiovascular system, permanently, reliably and autonomously keeping up the blood flow. In computational models, the cardiac function is reproduced in silico to deduce simulation studies that deliver deeper insights into underlying phenomena or investigate parameters of interest under perfectly controlled conditions. In light of cardiovascular diseases being the number one cause of death in Western countries, contributing to the early stage diagnosis is of high clinical relevance. In this context, computational fluid dynamics simulations can deliver valuable insights into blood flow dynamics and thus offer the opportunity to investigate a central area of physics of this multi-physics organ. As the deformation of the endocardial surface triggers the blood flow, the effects of elastomechanics have to be considered as boundary conditions for such fluid simulations. However, to be relevant in the clinical context, a balance between computational effort and necessary accuracy has to be met and models need to be both, robust and reliable. Thus, in this thesis the opportunities and challenges of light and therefore less complex coupling strategies are assessed focusing on three key aspects:

First, a fluid dynamics solver based on the immersed boundary approach is implemented, as this method attracts with a highly robust representation of moving meshes. Basic functionality was verified for various simplified geometries and showed a high accordance with analytical solutions. Comparing the 3D simulation of a realistic left heart geometry to a body-fitted mesh approach, basic global quantities were reproduced. However, variations in the boundary conditions revealed a high influence on the simulation results.

The application of the solver to simulate the influence of pathologies on the blood flow patterns showed results in good accordance with literature values. In mitral valve regurgitation simulations, the regurgitation jet was visualized by a particle tracking method. In hypertrophic cardiomyopathy, flow patterns in the left ventricle were assessed by a passive scalar transport to visualize the local concentration of the initial blood volume.

As in the just mentioned studies, only a unidirectional flow of information from the elastomechanical model to the fluid solver was considered, the retrograde effect of the spatially resolved pressure field resulting from fluid dynamics simulations on the elastomechanics is quantified. A sequential coupling approach is introduced to consider fluid dynamics influences in a cycle-to-cycle coupling structure. The low deviations in the mechanical solver of 2 mm vanished already after one iteration, implicating that the retrograde effect of fluid dynamics in the health heart is limited.

Summing up, especially in fluid dynamics simulations, boundary conditions have to be set with caution as due to their high influence the vulnerability of the models is increased. Nevertheless, light coupling strategies showed promising results in reproducing global fluid dynamic quantities while the dependence between the solvers is reduced and computational effort is saved.

Zusammenfassung

Das menschliche Herz ist das hochkomplexe Herzstück des kardiovaskulären Systems, das permanent, zuverlässig und autonom den Blutfluss im Körper aufrechterhält. In Computermodellen wird die Funktionalität des Herzens nachgebildet, um Simulationsstudien durchzuführen, die tiefere Einblicke in die zugrundeliegenden Phänomene ermöglichen oder die Möglichlkeit bieten, relevante Parameter unter vollständig kontrollierten Bedingungen zu variieren. Angesichts der Tatsache, dass Herz-Kreislauf-Erkrankungen die häufigste Todesursache in den Ländern der westlichen Hemisphäre sind, ist ein Beitrag zur frühzeitigen Diagnose derselben von großer klinischer Bedeutung. In diesem Zusammenhang können computergestützte Strömungssimulationen wertvolle Einblicke in die Blutflussdynamik liefern und bieten somit die Möglichkeit, einen zentralen Bereich der Physik dieses multiphysikalischen Organs zu untersuchen. Da die Verformung der Endokardoberfläche den Blutfluss antreibt, müssen die Effekte der Elastomechanik als Randbedingungen für solche Strömungssimulationen berücksichtigt werden. Um im klinischen Kontext relevant zu sein, muss jedoch ein Mittelweg zwischen dem Rechenaufwand und der erforderlichen Genauigkeit gefunden werden, und die Modelle müssen sowohl robust als auch zuverlässig sein. Daher werden in dieser Arbeit die Möglichkeiten und Herausforderungen leichter und daher weniger komplexer Kopplungsstrategien mit Schwerpunkt auf drei Schlüsselaspekten bewertet:

Erstens wird ein auf dem Immersed Boundary-Ansatz basierender Fluiddynamik-Löser implementiert, da diese Methode mit einer sehr robusten Darstellung von bewegten Netzen besticht. Die grundlegende Funktionalität wurde für verschiedene vereinfachte Geometrien verifiziert und zeigte eine hohe Übereinstimmung mit der jeweiligen analytischen Lösung. Vergleicht man die 3D-Simulation einer realistischen Geometrie des linken Teils des Herzens mit einem körperangepassten Netzbeschreibung, so wurden grundlegende globale Größen korrekt reproduziert. Allerdings zeigten Variationen der Randbedingungen einen großen Einfluss auf die Simulationsergebnisse.

Die Anwendung des Lösers zur Simulation des Einflusses von Pathologien auf die Blutströmungsmuster ergab Ergebnisse in guter Übereinstimmung mit Literaturwerten. Bei Simulationen der Mitralklappeninsuffizienz wurde der rückströmende Anteil mit Hilfe einer Partikelverfolgungsmethode visualisiert. Bei hypertropher Kardiomyopathie wurden die Strömungsmuster im linken Ventrikel mit Hilfe eines passiven Skalartransports bewertet, um die lokale Konzentration des ursprünglichen Blutvolumens zu visualisieren.

Da in den vorgenannten Studien nur ein unidirektionaler Informationsfluss vom elastomechanischen Modell zum Strömungslöser berücksichtigt wurde, wird die Rückwirkung des räumlich aufgelösten Druckfeldes aus den Strömungssimulationen auf die Elastomechanik quantifiziert. Es wird ein sequenzieller Kopplungsansatz eingeführt, um fluiddynamische Einflüsse in einer Schlag-für-Schlag-Kopplungsstruktur zu berücksichtigen. Die geringen Abweichungen im mechanischen Solver von 2 mm verschwanden bereits nach einer Iteration, was darauf schließen lässt, dass die Rückwirkungen der Fluiddynamik im gesunden Herzen begrenzt ist.

Zusammenfassend lässt sich sagen, dass insbesondere bei Strömungsdynamiksimulationen die Randbedingungen mit Vorsicht gewählt werden müssen, da sie aufgrund ihres großen Einflusses die Anfälligkeit der Modelle erhöhen. Nichtsdestotrotz zeigten vereinfachte Kopplungsstrategien vielversprechende Ergebnisse bei der Reproduktion globaler fluiddynamischer Größen, während die Abhängigkeit zwischen den Lösern reduziert und Rechenaufwand eingespart wird.

Acknowledgments

I would like to take this opportunity to thank all those who have supported me in my work at the Institute of Biomedical Engineering in the past years and also during the progress on the projects of this thesis in any way.

First of all, I want to thank Prof. Dr. rer. nat. Olaf Dössel who made it possible to me to conduct this research at the IBT and who laid the foundation for this work with his vision of seeing blood flowing through the heart chambers. Not less important was the the continuous support, supervision and guidance by PD Dr.-Ing. Axel Loewe. Thank you for all beneficial discussions, continuous feedback and motivation.

I want to thank Prof. Dr.-Ing. Bettina Frohnapfel for the willingness to co-referee this work, the joint supervision of several student projects, and the always valuable feedback on fluid dynamics. Thank you very much, Dr.-Ing. Alexander Stroh and M.Sc. David Müller for all support in immersed boundary verification questions and all enhancing discussions.

Additionally, I want to thank PD Dr. med. Marco Ochs from the Theresienkrankenhaus Mannheim for all collaboration and the possibility to acquire measurement data.

Furthermore, I want to thank the whole IBT team for all support and colleagueship. Thank you for all the talks, all the fun and all the productive time we had together! I wish you all the best for your thesis projects and your personal life.

All members of the CaMo team, I especially want to thank for all scientific discussions we had during our group meetings, all constructive feedback questions, all pictures of the week and all your will to again and again take the plunge into fluid dynamics.

A special thanks to all students, who contributed to the projects of this thesis and who were not deterred by the complexity of fluid simulations: Carlo, Farokh, Miriam, Michael, Johannes and Mathias.

Thank you to all members and part time members of the great LEN team for all exam task creations, exam reviews, corrections, and whatever else had to be done. A special thanks goes to Claudia who endured all these years with me: Thank you so much for always helping out!

Thanks to all friends, who supported me and brought me to other thoughts now and then. Especially thanks to Tobi for all talks during our (more or less) unstressed bike tours.

Last but not least, I want to thank Anna, as well as Angela, Fred and Carin for all your unlimited support and unconditioned love throughout the years and for making this work possible in the first place. Thank you so much for always being by my side, bringing me back down to earth and caring me through the times. You are the best that happened to me. This also applies to you, Lars and Mia. You are a wonderful gift.

Finally, to close with Psalms 115:1 - honor to whom honor is due.

Contents

Abbreviations

ALE	Arbitrary Lagrangian Eulerian	6, 25, 126, 148, 160
AV	atrioventricular	14 ff.
BC	boundary conditions	23, 29, 32, 51, 79, 101, 109, 168 f.
BF	back flow	128, 130
CFD	computational fluid dynamics	5, 23, 26, 109, 131
CT	computed tomography	4, 18
DEF	delayed ejected flow	128 f.
DF	direct flow	128 ff.
ECG	electrocardiogram	17 f., 120
ED	Euclidean distance	77, 87, 101, 104, 153–156, 162 f.
EDV	end-diastolic volume	14, 17
EF	ejection fraction	14, 17
ESV	end-systolic volume	14, 17
FEM	finite element model	29
FSI	fluid-structure interaction	6, 31, 145 f., 160
FV	Finite Volume	24, 26, 125, 127, 168
HCM	hypertrophic cardiomyopathy	17, 50, 135, 142
IBM	immersed boundary method	6 f., 37, 44 f., 53 f., 59, 69 f., 72, 74, 76, 85, 101 f., 105 ff., 109, 112, 115, 135, 142, 167 f.
LA	left atrium	5, 13 f., 17, 28, 58 f., 61, 63–66, 101, 107, 115, 125, 128 f., 141 f.
LV	left ventricle	7, 13 f., 17, 25, 28, 30, 37, 50, 53 ff., 58 f., 64 f., 83, 101 ff., 106 f., 112, 115–118, 120, 125, 128–132, 135, 140 ff., 156 f., 162 f.
MRI	magnetic resonance imaging	7, 18 f., 115, 120, 125, 148
MV	mitral valve	7, 14–17, 31, 37, 53 ff., 58 f., 63 f., 101 ff., 107, 119, 121, 125–128, 130 ff., 135, 137, 161
MVR	mitral valve regurgitation	17, 125, 127, 129–132
NSE	Navier Stokes equation	6, 22–26, 37, 54, 59, 69 f., 107, 127, 160
PF	pressure factor	145, 152, 154 f.
pV	pressure volume	14 f., 153 f., 158 f., 162
RA	right atrium	13 f., 30
RF	regurgitation fraction	17

RI	retained inflow	128 f.
RV	right ventricle	13 f., 50, 116, 132
SV	stroke volume	14, 17

Chapter 1

Introduction

The human heart is a masterpiece of highest complexity, responsible to reliably pump blood throughout the whole human body all around the clock. In order to accomplish this demanding task, the heart is a multi physics organ operating on a multi scale range, beating autonomously to drive the cardiovascular system. In computational cardiac modeling, this subject of interest is described by mathematical differential equations to adequately reproduce the cardiac function in silico. By applying such models, deeper insights in underlying phenomena can be gained, disease mechanisms can be further explored or the influence of a certain parameter of interest can be investigated under perfectly controlled conditions. Additionally, the models can be used to improve pathology diagnosis, to plan surgical interventions, handle and combine large sets of measurement data and finally support clinicians in decision making. All this is of high interest, as cardiovascular diseases are the leading cause of death today and early stage diagnosis may reduce the high mortality rates.

On this basis, computational fluid dynamic techniques are successfully applied to simulate blood flow dynamics inside the heart chambers and the circulatory system. More concretely, by executing fluid dynamics simulations, relevant effects like for example spatially resolved washout fractions can be quantified and evaluated for the identification of stasis areas which may increase the stroke risk due to cloth formation. However, due to the multi physics nature of the investigated subject, modeling the fluid dynamics in the heart can not be treated as an isolated problem, but has to be tackled under consideration of all contributing areas of physics and their interactions, of which the latter are captured by the respective coupling schemes. While electrophysiologic activity triggers each heart beat by electrical depolarization of the myocardial cells, the elastomechanical contraction of myocardial tissue results in a deformation and coordinated contraction-relaxation process of the cardiac chambers. This periodic process is the mechanism that perpetuates blood flow through a complex circulatory system to provide each cell in the whole human body with the necessary supply.

1.1 Motivation and aims of the thesis

In this scope, the overall aim of the thesis project is to contribute to this highly relevant area of research by further exploring the field of light coupling strategies. This discipline is of particular interest, as computational models always have to meet a balance between computational effort and necessary accuracy in the light of the respective application case, especially to be relevant in a clinical context. In addition, the models have to prove robustness and reliability. Thus, it is of relevance to investigate to which extent simple coupling algorithms are capable of reproducing relevant fluid dynamic characteristics.

In a first step, a simple, but in terms of mesh handling highly robust fluid dynamics solver is implemented based on an immersed boundary method approach. As the solver is designed to be able to incorporate endocardial surface boundary data from various sources, it is independent of other framework dependencies and follows a light coupling strategy based on a unidirectional cycle-to-cycle coupling. Thus, on the one hand the endocardial surface movement computed based on the existing mechanical solver framework, as well as measurement data can directly be used to supply the constraining boundary condition data for fluid simulations. This immediately leads to the first research question: Are simulations based on the immersed boundary method capable to deliver results with a similar accuracy than the already implemented body-fitted approach? To answer this question and systematically assess the opportunities and limitations, a solver based on the immersed boundary approach is implemented, verified on different simplified simulations scenarios and validated based on measured data.

In the next stage, hence the implemented solver (as well as the existing one) has to prove its capabilities in practical application and thus is applied to selected pathological scenarios of clinical relevance. This simulation study is deduced to answer the second research question: Can in silico fluid simulations reveal alterations in the flow conditions of patients suffering from hypertrophic cardiomyopathy as well as mitral valve regurgitation? Two independent simulation studies are deduced based on the immersed boundary method as well as the existing solver approach to evaluate these clinically relevant pathology scenarios.

As so far only unidirectional coupling schemes are utilized, in a third focus area subsequently the interdependence between the fluid solver and the elastomechanical solver is investigated. In a novel, loosely coupled approach, the influence of the spatially resolved fluid dynamic pressures on the deformation of the endocardial surface walls is quantified. This methodology is used to close the loop of a coupling scheme relying on a bidirectional exchange of state variables. In this context, the third question is addressed: Does the unidirectional coupling scheme lead to significantly different flow patterns in comparison to a bidirectionally exchange of information? Is it necessary to replace the unidirectional coupling by a bidirectional one and which alterations are introduced by the bidirectional coupling approach? To answer this question, a sequential coupling algorithm is introduced and the influence of fluid dynamics on the elastomechanical simulations is quantified.

Thus, as part of this thesis project two new methods are implemented and delivered to the cardiac modeling community: First, a solver based on the immersed boundary method is set up, verified and validated. On the other hand, a sequential coupling approach is introduced to model bidirectional fluid structure interaction in a simple way. To be independent of commercial and in-house software frameworks, the open source software package openFOAM (version 7 and version 1806) is used throughout all projects to solve the fluid dynamics equations [1, 2].

All in all, a realistic heart geometry obtained from measured data is evaluated to carve out possibilities and limitations of the light coupling methods covered in the scope of this thesis.

1.2 State of the art

1.2.1 Computational modeling

Advancements in computational resources as well as increasing availability of high performance computing systems boosted the use of computational modeling especially in the context of scientific research in the last decades. Nowadays, computational modeling is a firm foundation of research engineering across a variety of application fields [3]: No matter if in the clinical context [4] of the medical-biological field in the context of computational modeling manufacturing in industry in the context of thermal modeling [5] or in modeling behavioral data [6], computer models are the base to built on. This is not by chance, but because of the variety of benefits computational modeling offers. Computer models enhance the understanding of scientific phenomena [7], allow to predict certain simulation outcomes, to analyze, explain, optimize and control complex natural systems. They enable decision making and maximize efficiency and effectiveness in a design process [3] - just to name a few. In the last years, especially the integration of huge amounts of data, big data science and ubiquitous computing gained in importance [8]. Across the different application fields, the basic principles of applying models for simulation, parameter estimation and the comparison between models itself are common [6]. Basic procedures in scientific computing have been set up [8].

Of course, despite all advances and advantages, computational modeling is not the golden grail. Model specification and parameterization remains challenging as the knowledge of the underlying phenomenon is rarely complete [7]. Often, only partial information on the studied subject of interest is available, or can be measured or extracted from real world in a limited way and the model complexity is not always adjusted adequately to the research aim [7].

However, in the context of this thesis especially the advancements concerning the clinical application of computational modeling for personalized medicine as well as machine learning techniques are of great importance [4]. Diagnostics and treatment are improved by model based, individualized planning. Data integration across multiple measurement instrumentation as well as model validity play a key role in current and future modeling [4].

1.2.2 Cardiac computational modeling

Computational modeling and simulation comprises an immense number of applications in the context of cardiology without fully exploiting its overall potentials yet. However, the amenities concerning large data sets, integration of multiple measurement data sets, simulation studies, parameter variation, integration of different spatial and temporal scales, therapy planning, model individualization, cohort studies and implant optimization among many other make computer models an indispensable part of modern cardiology [9]. Computer simulations convince as they are cheap, reproducible, repeatable, fast compared to a measurement setup, enable a large scale, while all information is accessible, no risk is imposed to the patient and the procedure itself is non invasive [10]. A very comprehensive and detailed description of the historical developments in electrophysiology, elastomechanics and fluid dynamic advancements in the modeling of the heart as well as the mutual interdependence and coupling is presented by Verzicco et al. [11]. Many reviews on modeling the heart have been published over the years [9–12].

As the computational grid forms the base for all in silico simulations, an overview of the different mesh modeling approaches including structured and unstructured grids, tetrahedral and hexahedral elements as well as mesh generation processes are presented by Lintermann et al. [13]. An introduction to finite element modeling is included in the publication by Fish et al. [14]. The geometric heart models are usually obtained by segmentation of medical images. However, also deep learning approaches for direct model generation from images exist [15].

As the different areas of physics in the human heart are strongly dependent on each other, blood flow can not be computed without considering the electrophysiology and elastomechanical influences, as well as the systemic and pulmonary circulation [11]. Of course, all these disciplines are active fields of research and research groups focus either in detail on one area of physics, model the coupling between two areas of physics or even model all relevant parts for the whole heart at once. A huge number of groups focuses on cardiac electrophysiology. Thus, many introductions and overviews on the topic exist [12, 16, 17]. As an example, innovative treatment strategies have recently been published in [18]. On the other hand, other groups only focus on modeling the elastomechanical deformation. Jafari et al. introduced a simulation framework to simulate cardiac elastomechanics based on an idealized four chamber heart model [19]. This is similarly done with a high resolution mesh in [20]. Uniting both areas of physics, other groups presented electro-mechanical whole heart models [21–23]. To consider influences of the pulmonary and circulatory circulation, 0D lumped parameter models are common [20, 24–26]. While they differ in the elements used to model the system, a validation is suggested in [26].

Uniting all areas of physics, only a small number of research groups introduced electro-fluid-mechanic coupled simulation frameworks [27–30]. A very recent overview is presented in [11]. A strong coupling scheme between all areas of physics is applied in [24, 31]. Another possibility is, to directly use boundary conditions from measured data to compute blood flow dynamics. Mihalef et al. exemplarily use geometry data from 4D computed tomography (CT) measurements to simulate the fluid dynamics in the heart [32].

In the context of computational cardiac modeling the personalization of models plays a more and more important role for individualized treatment [33]. However, another emerging field is the credibility assessment of such models [34, 35].

Focusing on the simulation of fluid dynamics, different techniques are used to analyze fluid flow behavior and patterns which are relevant in the clinical assessment of heart function. Doost et al. present a very thorough review article including a large list of computational fluid dynamics (CFD) approaches, required stages in pre- and post-processing as well as an analysis of relevant boundary conditions. Besides a broad introduction, the review by Mittal et al. also highlights clinical applications [36]. A very recent study on the left atrium (LA) hemodynamics is published in [37].

In recent literature, mostly an arbitrary Lagrangian-Eulerian formulation of the Navier-Stokes equations was implemented to model the dynamic mesh behavior [28, 37–40]. Blood is commonly modeled as a Newtonian fluid and under consideration of laminar flow in the heart chambers and the great vessels [12, 41]. For modeling the valves, a variety of implementations and simplifications exist. However, as medical imaging delivers time resolved data, valves mostly are modeled as 3D structures by now [42]. Concerning the implementation, fixed and moving grid methods can be distinguished as reviewed in [43]. Concerning validation of fluid solvers, 4D flow MRI is the current method of choice [44]. Blanken et al. published validation data based on 4D flow MRI measurements [45].

1.2.3 Fluid structure interaction

To solve fluid-structure interaction problems two different basic approaches are common in literature (reviewed for example in [46–49]): On the one hand in the monolithic (also called implicit) approach, fluid and solid mechanics equations are computed simultaneously within one solver. Therefore, their mutual influence is directly taken into account, yielding high stability. In that case no explicit coupling algorithm is required as all dependencies are completely modeled in the system of equations. This approach impresses by its high accuracy and inherent inter-dependency of all results. On the other hand the mathematical formulations and their numerical implementation is a challenge. The time step is also restricted due to the requirements of a sufficiently small Courant-Friedrichs-Lewy number [50]. Nevertheless, the resulting quantities of both physical domains are immediately available in every time step.

On the other hand partitioned approaches have been suggested [51]. Fluid and elastomechanics equations are hereby solved separately, independent of each other by two single solvers. Classically, these two solvers are solved sequentially. Thus, a coupling scheme has to be implemented to account for the mutual influence of the two physical domains. In such a routine information must be communicated between the two solvers in a bi- or mono-directional manner. This approach is computationally more attractive and offers higher flexibility concerning the single solvers. Thus, in each solver the advanced techniques concerning the respective physics can be applied independently of one another [46]. However, the required coupling routine has to map data between two temporally and spatially asynchronous grid meshes.

Besides the two basic, purely implicit and explicit algorithms also semi-implicit schemes [52], implicit-explicit variants [24] and multi-way coupled algorithms [53] have been proposed. All in all, a high number of fluid-structure interaction (FSI) based algorithms and application fields in the cardiovascular system have been published in recent years [46, 54]. The number of applications in a clinical context is continuously increasing with computational power and better algorithms. The opportunities and potentials of computational modeling are immense, however not depleted.

1.2.4 Immersed boundary method

As it provides the necessary mathematical and computational frameworks, one method to model FSI is the immersed boundary method (IBM) [55]. Peskin et al. first applied a method that is nowadays referred to as the IBM in the 1970s [56]. This implementation was the first to be based on a Cartesian grid that did not correspond to the heart geometry and a procedure was formulated to model the effect of the IBM on the flow [57]. As reviewed in many publications, such methods to simulate viscous flow on grids that do not model the shape of the original geometries but immerse or embed the boundary have been investigated and applied by many research groups [55, 57, 58]. What they have in common is, that typically the Eulerian description is applied for the fluid, while the Lagrangian description is chosen for the structure. Therefore, thin immersed boundaries can be used to model flexible or rigid boundaries that move with a prescribed deformation kinematics [55]. Thus, applying the IBM, solid (focus in review [57]) as well as moving (focus in review [55]) boundaries can be immersed in the underlying grid structure. A number of methods have been proposed, comprising the use of the framework of isogeometric analysis, Lagrange multiplier methods or sharp-interface IBMs and proved good results in terms of accuracy [55]. Some of the latest approaches comprise the following procedures: One method to model especially moving boundaries was recently presented as a strongly coupled algorithm which treats the pressure and the boundary force as Lagrange multipliers to enforce incompressibility and no-slip wall constraints [59]. On the other hand, Wang et al. make use of the fractional step method to compute the implicitly determined body force implied by the immersed body. This direct forcing approach makes use of large-eddy simulation and enables the simulations of high Reynolds number turbulent flows [60]. Another alternative was introduced by Xu et al. to mimic details of tomographic particle image velocimetry experiments. A radial basis function mesh morphing method is introduced to model the deforming wall boundaries in an Arbitrary Lagrangian Eulerian (ALE) based manner [61]. In addition, the method that is followed up in this project is based on the addition of a force term to the Navier Stokes equation (NSE) to model the influence of the IBM and was introduced by Goldstein et al. [62]. To deduce the boundary of the immersed surface, a shape indicator can be computed as the result of a level set function [63].

In the application context of modeling the human heart, IBM so far was mostly applied for modeling the movement of the mitral valve (MV) plane and leaflets [55]. Exemplary, each valve is modeled by the IBM and is mounted in a semi-rigid flow chamber [64]. Alternatively, the 2D mitral leaflet flow can be computed based on an IBM scheme coupled to a Lattice Boltzmann scheme [65]. In a next step towards whole heart modeling, solely an idealized left ventricle (LV) geometry was modeled following the IBM approach [61, 66]. Kim et al. modeled the whole left heart side coupled to a lumped element circulatory system. However, the underlying heart geometry was not a 3D one but only a 2D approach [67]. A big step towards whole heart modeling, fully relying on an IBM approach was done by Tay et al. [68]. Based on 4D magnetic resonance imaging (MRI) flow measurement data, an idealized heart was simulated based on the model and the solver introduced by McQueen et al. [69]. In addition to the LV modeled by idealized cones, a more realistic 3D four chamber heart model model was presented [70] and used to simulate the flow in the LV obtaining realistic results for the flow in the LV and across the aortic valve [71].

1.3 Structure of the thesis

To introduce the reader to the research field covered in the scope of this thesis project, **Chapter 1** includes the introduction. After the research project is motivated, the aims of the project are stated. Subsequently the current state of the art is presented.

Part I comprises the basic fundamentals relevant in the context of this work in a concise form. It includes:

- **Chapter 2** that presents an overview of the medical fundamentals. Anatomic structures, physiological processes, selected pathologies as well as measurement principles are presented.
- **Chapter 3** includes the mathematical fundamentals necessary to model and simulate blood flow.

Part II includes the fluid dynamics solver implementation details:

- **Chapter 4** provides details on the immersed boundary solver implementation.
- **Chapter 5** presents an automated pipeline to extract the necessary boundary conditions from the elastomechanics simulation framework.
- **Chapter 6** gives an overview on the different approaches for the representation of valves investigated in the scope of this thesis project.
- **Chapter 7** describes a study on the influence of the length of the prolonged trunks that model the pulmonary veins and the aorta.

Part III is dedicated to the verification of the solver: Its aim is to demonstrate the functionality of the methods introduced in Part II and also show possible limitations.

- **Chapter 8** presents various simplified simulation setups to verify different aspects of the immersed boundary solver. Starting with simple geometries immersed in a 2D grid, complexity rises incrementally up to a moving boundary immersed in a 3D grid.
- **Chapter 9** focuses on the influence of the grid resolution and its effect on the simulation results.
- **Chapter 10** finally presents the 3D heart simulations based on the introduced procedure. To demonstrate functionality, results are compared against the existing body fitted solver implementation.

Part IV includes measured flow data to validate the introduced method against a measured gold standard data set. This is done in **chapter 11**.

In **Part V** the investigated fluid solvers are applied to 3D heart models under pathological conditions:

- **Chapter 12** focuses on mitral valve regurgitation.
- **Chapter 13** presents a study on altered fluid dynamics in hypertrophic cardiomyopathy.

Part VI presents a deeper investigation of the mutual influence between elastomechanics and fluid dynamics. A simplified cycle-to-cycle coupling is considered. In this context, the sequential coupling approach is introduced in **Chapter 14**

Finally, **Part VII** concludes the thesis with final remarks:

- **Chapter 15** presents an overall discussion of the projects included in this thesis.
- **Chapter 16** states the drawn conclusions as well as it gives an outlook on potential future fields of investigation.

Part VIII includes the appendix and thus presents additional information on the time series of the flow measurement, lists circulatory system parameters and presents the scripts for the prolongation of the trunks.

Regularly, all chapters follow a common structure: Starting with a motivation for the following study, an introduction is included where necessary. Subsequently, the applied methods are explained before the results are explained. Each chapter is closed by a short discussion section to finalize the respective topic.

PART I

FUNDAMENTALS

Chapter 2

Medical fundamentals

No life without blood – in its absence, human life would literally come to a standstill in shortest time. This unique and essential fluid thus fulfills a key role in the human body: If its supply is interrupted, permanent damage to the brain begins after just a few minutes [72].

The first of the following sections therefore summarizes the characteristics and tasks of blood and describes how it is transported through the body. In the second section, the heart with its anatomy and physiology is introduced as the driver of the cardiovascular system. To conclude this chapter of medical basics, the cardiac diseases relevant in the context of this thesis are presented.

2.1 Blood and the cardiovascular system

Blood is chemically a suspension – a heterogeneous medium consisting of a liquid that transports a fine distribution of small components and chemicals. The liquid portion is also referred to as the blood plasma. Depending on sex, plasma makes up a mean percentage of 53 % (female) to 59 % (male) of the overall blood volume of four to six liters (female four to five l, male five to six l). It consists of water (about 92 %) and the suspended substances, mostly proteins with different functions. The cellular components mainly comprise of red blood cells, the erythrocytes. In comparison to those, the proportion of leukocytes and platelets, is small. With all these ingredients, blood behaves as a viscous medium, with approximately the fivefold viscosity compared to water [73].

The central function of blood is the rapid transport of oxygen (mostly bound to the erythrocytes) and nutrients (dissolved in the plasma) towards the body cells, as well as the transport of carbon dioxide and other wastes away from the cells towards the organs that excrete or recycle them. Additionally, other important substances are transported by the blood, for example hormones that serve to regulate various body functions. The components present in the blood also play a central role in immune defense and wound closure. Another central function of the blood pool is the maintenance of homeostasis including many variables. For example, by transporting heat from the body core to the periphery, temperature is regulated. Blood also ensures the water balance, as well as chemical homeostasis (e.g. pH) [74].

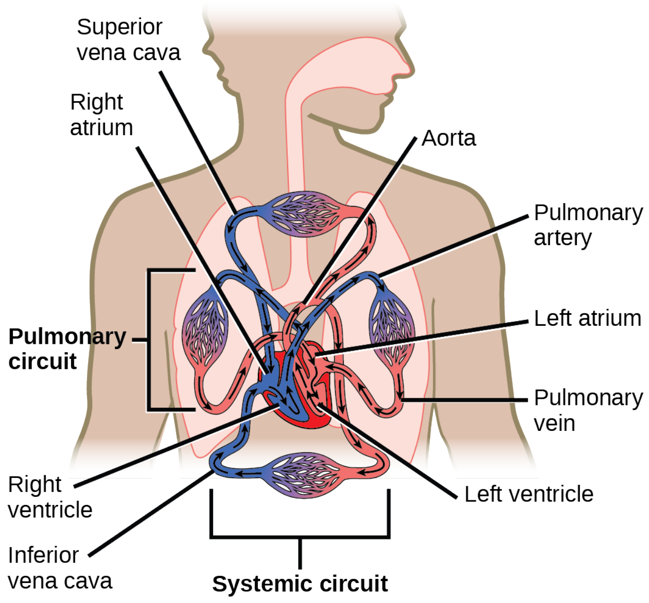


Figure 2.1: Compact schematic representation of the cardiovascular circulatory system: Oxygenated blood (red) is transported via the systemic circulation towards the cells, while desoxygenated blood (blue) is transported towards the lung to be oxygenated again. Adapted from [73], licensed under CC BY 4.0.

For all these tasks, blood needs to move. In this context, two physical transport phenomena play a central role: diffusion and convection. Diffusion describes the process of a balancing movement of a substance from regions with higher concentration to regions with lower concentration. This process is based on the self-motion of the particles and a consequence of the increasing entropy in the system. This process e.g. takes place in the capillaries of the lung and the tissue on a very small spatial scale (micrometer range) and enables the exchange of oxygen. In figure 2.1 this happens at the strongly branched blood vessels, where the color of the blood changes from blue to red (or vice versa). As this process is only fast on very small spatial scales (time t proportional to squared distance x^2), a diffusive transport of blood throughout the whole body is not feasible. Thus, in a second process, blood is actively transported by convection. To move between various organs, tissues and vessels throughout the whole human body, blood is pumped through a closed-loop circulatory system. This cardiovascular system is shown compactly in figure 2.1.

The cardiovascular system consists of numerous blood vessels of different sizes to link all organs and deliver the necessary supply as described before. It can be separated in two circulations that are connected in series to form a closed-loop system: While in the systemic circulation oxygenated blood is pumped from the left side of the heart towards the organs and back into the right heart, the pulmonary circuit connects the right heart via the lungs to the left heart. The vessels that transport blood away from the heart are called arteries, whereas the vessels ending at the heart are named veins. Thus, in the systemic circuit, arteries transport oxygen-rich blood towards the organs (which are connected in a parallel

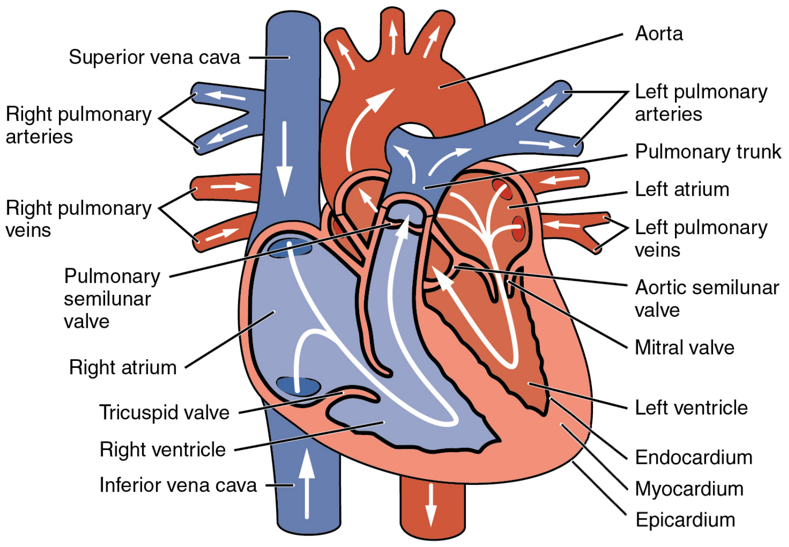


Figure 2.2: Heart anatomy. Edited from [75], licensed under CC BY 4.0.

arrangement), while veins transport desoxygenated blood back to the heart. This is the other way around in the pulmonary circulation: pulmonary arteries transport desoxygenated blood.

The blood flows through blood vessels, which are tubular, elastic structures. Due to their elasticity, the larger blood vessels are able to fulfill a pressure reservoir function (Windkessel effect). The pressure exerted by the blood against the vessels is known as blood pressure. As a diagnostic parameter, it provides information about the condition of the circulatory system.

2.2 Heart anatomy and cardiac cycle

The centerpiece of the cardiovascular system is the heart, a muscular organ that is located in the thorax centrally between both lungs. Constructed as a hollow muscle, its task is to continuously and reliably pump blood between all organs and thus drive the convective transport throughout the whole body as described above. The anatomic structures that are relevant in this context are illustrated in the schematic in figure 2.2.

To accelerate and eject blood into the two separate circulation systems, the heart cavity is vertically divided in two parts by the septal wall. Each side is again divided horizontally in two cavities by the atrioventricular plane: the atrium (left atrium (LA) and right atrium (RA)) and the ventricular chamber (left ventricle (LV) and right ventricle (RV)). This makes an overall number of four cardiac chambers through which the unidirectional blood flow (depicted by white arrows in figure 2.2) is guided by four valves that prevent regurgitating flows. All chambers are surrounded by elastic heart tissue (myocardium) that is responsible for the acceleration of the blood: A contraction of the chamber walls leads to an increased pressure and consequently in combination with the opening of the corresponding valves, to a volumetric change. Blood is ejected into the circulation systems. The inner endothelial layer of the myocardial tissue (endocardial surface) is in direct contact with the blood and forms its natural boundary. The whole heart itself is surrounded by an additional layer that constrains the overall movement and allows for a frictionless sliding. The pericardium, which is not explicitly shown in figure 2.2, surrounds the outer layer of the myocardial wall (epicardium) with a thin fluid film in between.

To fulfill the task of pumping blood autonomously, in a self sustaining manner and independently of external impulses, the heart is triggered by electrical signals. Therefore, the heart tissue is penetrated with a complex electrical excitation and activation propagation system. The heart's primary pacemaker is the sinoatrial node, located in the superior and posterior RA wall close to the superior vena cava. There, a recurrent spontaneous depolarization of pacemaker cells result in an action potential. This impulse spreads along intraatrial conduction paths towards the atrioventricular (AV) node as well as it is transferred to the cells of the atrial contractile conduction paths. Within the depolarized cells of the contractile myocardial tissue, the interaction of actin and myosin filaments results in a tension and the contraction of the cell. This process is referred to as the electro-mechanical coupling. A coordinated mechanical contraction of the excited atrial tissue results. Meanwhile, the AV node delays the electrical excitation propagation to ensure that the atrial contraction has ended before ventricular contraction starts. Subsequently, the electrical activation is transmitted via the bundle of His, the left and the right bundle branch towards the Purkinje fibers. In a healthy heart, the stimulated myocardial cells contract in a coordinated and directed movement from the bottom of the heart (apex) towards the top. This wringing movement is boosted by the alignment of cardiomyocytes in the formation of fiber layers. After the cells repolarize again, a refractory period follows until they are ready for the next cycle of excitation. This continuous and regular pacing is referred to as sinus rhythm – however, in many heart diseases this structured excitation propagation is disordered.

All in all, the human heart is a multi physics organ operating on a multi scale range – a masterpiece of highest complexity beating reliably all around the clock. To adapt the performance of the heart in correspondence with the physiological conditions, several control and feedback loops are implemented. Most important in this context is the Frank Starling mechanism that describes the reaction on varying pre- or afterload.

The heart thus works as a time-varying pulsed pump that transports a certain stroke volume (SV) of blood with each beat of the heart. In the following paragraph, the cardiac cycle is explained based on its pressure volume (pV) relationship. Figure 2.3 shows a pV diagram, which also serves as a diagnostic tool in the clinical context. There the chamber pressure is plotted versus the volume. The numbers added in brackets in the following mark the relevant characteristic points in time and are depicted in the pV diagram in figure 2.3.

The beginning of a cardiac cycle is usually chosen at the diastatic state (position 1 in figure 2.3), in which the heart is completely relaxed. During the previous (ventricular) diastole both AV valves that separate the atrial chambers from the ventricles were open. Thus, both sides of the heart were passively filled with blood: Oxygen-rich blood had flown from

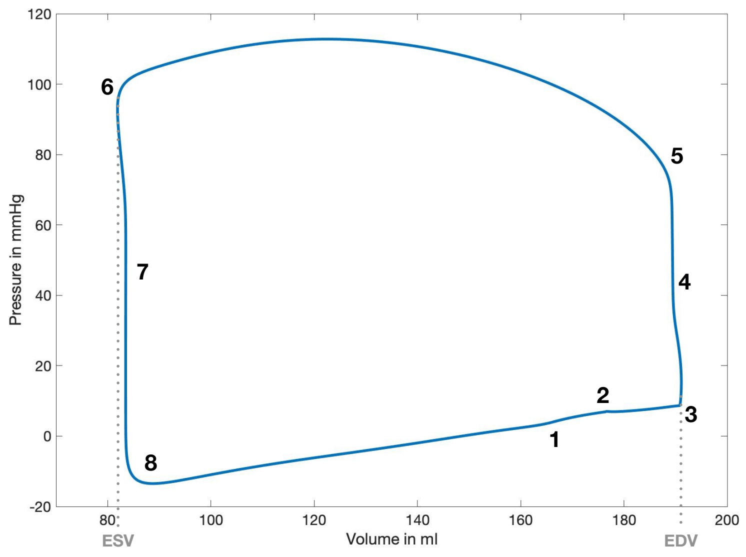


Figure 2.3: Pressure volume diagram of one full heart cycle exemplary for the LV. The course of the cardiac cycle is described in the text.

the pulmonary veins through the LA into the LV as well as desoxygenated blood from the systemic circulation into the RA and RV. The other two valves that are located at the outflow trunks of both ventricles (aortic valve in the left heart and pulmonary valve in the right side of the heart) are closed. At the diastatic state, both ventricles are filled between 70 % and 80 % of their maximal volume. The end diastatic state is followed by the contraction of the atria, the atrial systole, which is also referred to as the late filling phase (2). This atrial kick pumps an additional blood volume into the ventricles, such that they are filled with an end-diastolic volume (EDV) of approximately 130 ml (literature value, in the existing geometry 190 ml). As the ventricular myocardium starts to contract (3), the AV valves (mitral valve (MV) in the left heart and tricuspid valve in the right side of the heart) close and prevent a regurgitating flow from the ventricles towards the atria. An isovolumetric compression (4) follows. When the ventricular pressure exceeds the aortic (respectively pulmonary artery) pressure, the aortic (respectively pulmonary) valve opens (5) and blood is ejected in the systemic (pulmonary) circulation. The amount of ejected blood is the SV, while its relation to the EDV, known as the ejection fraction (EF) ($EF = \frac{SV}{EDV}$), is a diagnostically relevant measure for the cardiac performance. The ventricles contract until the chamber volume is minimal, the so called end-systolic volume (ESV) (6). This marks the end of the (ventricular) systole and is directly followed by the beginning of the diastole. As soon as the relaxation of the ventricles begins, both valves in the outflow trunks close and an isovolumetric relaxation phase (7) follows. When the ventricular pressure falls back below the atrial pressure, the AV valves open and blood is sucked into the ventricles (8). The filling phase resumes and thus the heart cycle can begin once again.

2.3 Heart valves

To ensure unidirectional blood flow in the cardiovascular system and to enable isovolumetric phases during the heart cycle, valves are included in the system. They thus separate the heart into different compartments, just as in a classic mechanical system. As mentioned before, those valves open and close at characteristic points of the cardiac cycle to ensure proper functionality of the heart. Their opening and closing results in the characteristic shape of the pV-loop. Separating the ventricular chamber from the atrium, the AV valves are located in the transversal cutting plane in the so called atrioventricular septum or the base of the heart. The valves itself are called mitral (bicuspid) valve in the left heart side and tricuspid valve in the right heart. Both of them are made of endocardial tissue and consist of two (respectively three) flaps, which are connected to the papillary muscles via the chordae tendineae. These small, string-like structures ensure a correct closing of the valves and prevent from bulging into the atrial chamber during the strike back of the flaps while the valve is closing. A close-up of the left heart side depicting the MV, namely the anterior and the posterior leaflet is visualized in figure 2.4.

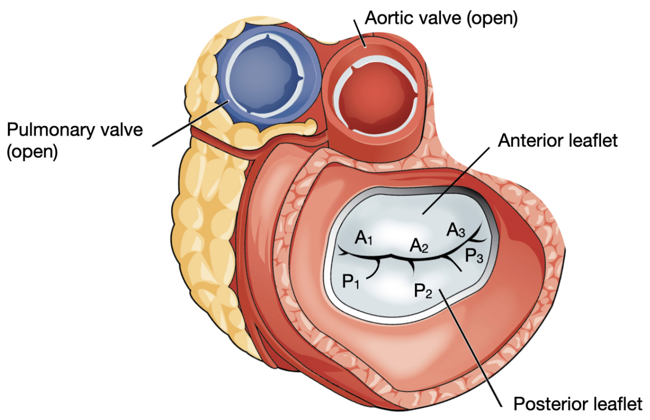


Figure 2.4: Schematic figure of the closed mitral valve. Adapted from [75], licensed under CC BY 4.0.

The closing of a valve is triggered by the blood flow: as soon as for example the left ventricle contracts and blood tries to regurgitate into the left atrium, the MV closes immediately. In contrast, the opening of the valve is controlled by the pressure difference between the area before and after the valve: If the pressure in the atrial chamber exceeds the ventricular pressure during the atrial kick, the AV valve opens. Two additional valves are located at the outflow area of each ventricle, separating them from the subsequently following artery. In the left heart side, the aortic valve separates the chamber from the aorta; in the right heart side the pulmonary valve (pulmonary semilunar valve) separates the right ventricle from the pulmonary artery (see also figure 2.4). Both of them are not linked to string-like structures, but with their pocket-like shape of the three leaflets in each valve, the ventricular blood seals the valves and prevents from regurgitating flows.

2.4 Selected cardiac pathologies

Cardiovascular diseases are the most common cause of death in Western countries [76] and the variety of cardiac and cardiovascular diseases is huge. In this section, two prevalent diseases are introduced that are investigated in computational studies in this thesis (chapter 12 and 13). Causing altered flow conditions in the heart chambers, valvular defects are relevant from a fluid point of view. In this work, more concretely mitral valve regurgitation is investigated and thus introduced here from a medical point of view. Second, the basics of hypertrophic cardiomyopathy are explained. In this disease, altered LV volume conditions also lead to altered flow conditions.

2.4.1 Mitral valve regurgitation

Mitral valve regurgitation (MVR) describes the inability of the MV leaflets to close tightly and thus not sufficiently covering the orifice of the mitral annulus, resulting in a regurgitating flow from LV to LA [77]. Anterior and posterior leaflet (visualized in figure 2.4) do not form a closed surface and do not ensure unidirectional flow as intended. Thus, this disease is characterized by a regurgitation of parts of ventricular blood back to the LA instead of being ejected in the systemic circulation. The regurgitation fraction (RF) represents the volume of blood that regurgitates into the LV (V_{BF}) in relation to the SV in percent:

$$RF = \frac{V_{\mathrm{BF}}}{SV} \cdot 100\,\% \tag{2.1}$$

The biggest volume regurgitates during ventricular systole, small regurgitation continues until the diastolic relaxation phase and occurs already at the beginning of contraction [78]. MVR has a high prevalence and affects every fifth person at least in a mild state [79]. After a myocardial infarction this prevalence is increased up to the threefold [80]. Beneath myocardial infarction, MVR arises mainly in the context of (mostly congenital) MV prolapse. Rheumatic MVR is only relevant in developing countries. Two types of MVR are distinguished: Organic (primary) MVR, in which changes at the valve itself are the cause of the regurgitation, is differentiated from functional (secondary) MVR, in which the valve is intact but the surrounding tissue or the LV is the primary cause of the disease [81]. Commonly a classification in three severity stages is made: mild, moderate or severe, depending on regurgitation area (see table 12.1 based on [82]) or regurgitation fraction [83]. The reduced forward SV may limit physical capabilities and lead to long term overload of the heart. Main symptoms are palpitations, dyspnea and fatigue. At low severity levels MVR often remains undetected.

2.4.2 Hypertrophic cardiomyopathy

The term 'cardiomyopathy' summarizes several diseases concerning structural changes in the myocardial muscular tissue. A relatively common type in this context is the hypertrophic cardiomyopathy (HCM). The specification 'hypertrophic', which means as much as 'enlarged', implies that the myocardial wall is pathologically thickened. This pathology is diagnosed if the LV end diastolic wall thickness exceeds $d_{ED,LV} > 13\,\mathrm{mm}$ [84] respectively the latest if $d_{ED,LV} > 15\,\mathrm{mm}$ [85]. HCM is also associated with myocardial fiber disarray, which is quantified by fractional anisotropy. The latter was found to be lowered in HCM patients [86]. Additionally, remodeling changes occur and passive ventricular stiffness is increased (decreased compliance) [78]. With HCM, often an LA enlargement comes along. This, as well as a ST-segment depression can be diagnosed using a 12-lead-electrocardiogram (ECG), which is abnormal in 75 % to 95 % of the patients. However, the differentiation of HCM and other pathologies from the ECG is challenging. HCM is characterized by a non-dilated LV, resulting in normal or slightly diminished EDV and ESV [78]. In 90 % of cases an asymmetric disease pattern with a septal location is predominant [78]. HCM is an autosomal-dominant inherited genetic disorder with large genomic heterogeneity and the most common monogenetic cardiovascular disease [87]. Approximately 60 % of HCM patients have a familial predisposition. In two of three patients the thickened myocardial tissue in the area of the LV outflow tract leads to a functional aortic stenosis (peak gradient above 30 mmHg), as the outflow tract is cramped. Therefore, patients are categorized in three LV outflow tract obstruction categories, which all are of the same prevalence: obstruction at rest, obstruction in exercise (labile) and no obstruction. The overall prevalence of HCM was estimated between 0.16 % and 0.29 %, so approximately one in 500 adults is affected [84]. HCM is without doubt the most common cardiovascular disorder [88]. The incidence of sudden death is significantly increased by the fivefold in HCM patients. While patients with diagnosed HCM have normal or near-normal life expectancy, it remains the most common cause of sudden cardiac death in those younger than 35 years [89]. HCM symptoms are shortness of breath, chest pain, dizziness, a decrease of physical performance as well as cardiac arrhythmia like syncope. A preserved or increased EF make the diagnose more difficult. For further reading, the reviews [84, 90] or books [78, 88, 91, 92] provide deeper insights.

2.5 Cardiac imaging

The medical imaging technique of particular interest for this work is magnetic resonance imaging (MRI), as it is able to visualize blood flow by non-invasive measurements. The so called 4D-flow MRI acquisition is a further development of conventional MRI, in which only cross-section images of the human body are acquired. It is discussed in several reviews [93–96]. MRI imaging also provides better contrast of soft tissue compared to computed tomography (CT).

In general, MRI is based on the principle of nuclear magnetic resonance, which states that hydrogen protons in a strong magnetic field respond with a measurable electromagnetic signal when excited by a radio-frequency pulse. This phenomenon is based on the spin of nuclei and a the torque-induced precession caused by a constant outer magnetic field. Thus, an MRI scanner combines the three main components of a strong magnetic field (background), gradient coils that induce magnetic gradient fields and thirdly coils that induce waves (pulses) in the radio frequency spectrum (the so called Larmor frequency). Measurement coils capture the emitted signal. Unlike CT, MRI does not use ionizing radiation. The resulting images reflect the density of protons and are often computed as stacks of slices (2D image planes). As different materials are related to different T1 and T2 relaxation times, a contrast between them is visible. The resulting different resonance behavior of body tissue and blood enables to extract the heart geometry as a basis for the computer model from the classic MRI images. The contrast is thereby determined by the T1- and T2-weighting. Image acquisition is triggered by different possible options, either for example the ECG or respiration-gated, i.e. triggering on position of diaphragm. During data acquisition the Fourier-space (k-space) of the image is sampled [97]. With an inverse Fourier transformation, the resulting image is computed. Based on the 2D slices, computers are able to reconstruct the corresponding 3D images [98].

It can be distinguished between two different procedures: While with classic MRI, geometric images of the (static) anatomy are acquired as described previously, in functional MRI time resolved physiological processes are captured. In the classic MRI analysis, the majority of diagnostic findings are based on the signal amplitude [99].

Quantitative flow imaging is based on the phase contrast principle. This works in the following way: If a gradient magnetic field is applied to a test object and additionally an equal, but opposite gradient field is applied subsequently, the protons that are stationary will show no phase shift. However, those protons that are moving will undergo a phase shift because their location along the gradient is constantly changing. This concept can be used to identify protons that are moving through the relevant plane during the measurement process (or through the 3D space in 3D acquired imaging). During this phase shifting process, faster protons will experience a greater phase shift than slower protons [99]. In this way, the application of a bipolar gradient enables an encoding of the velocity data in the image. To calculate a phase shift image, the area of interest is compared with a background image to compute the difference between the phases. In general, phase-contrast methods are widely used in cardiac imaging for functional assessment of blood flow across the valves and to visualize and quantify velocity [99].

With the transition from single-directional flow (which is commonly referred to as 2D-flow imaging) to way more complex acquisition in phase contrast MRI, the use of three-directional velocity-encoded gradient fields is required. This acquisition, however, allows for visualization of spatial and temporal evolution of 3D blood flow patterns within the acquired 3D volume [93]. This technique is referred to as 4D MRI flow measurements [100]. To fully construct the 3D blood flow, the velocity encoding has to be applied for each of the three principal spatial directions x, y and z.

This results in three phase images in addition to the background image for each time step [101]. This is exemplary shown in figure 2.5.

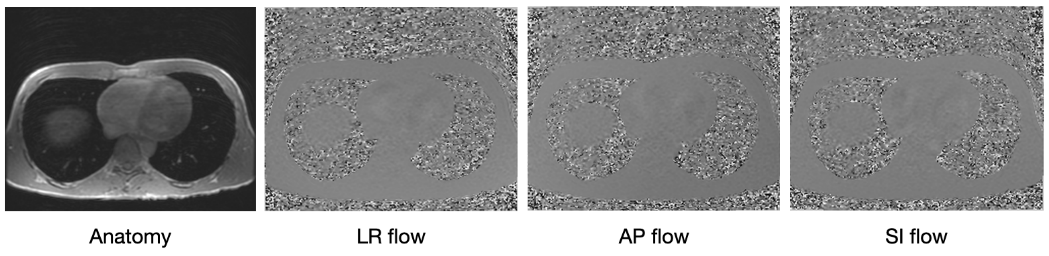


Figure 2.5: Exemplary 4D flow measurement: For one specific point in time and one specific slice of the heart four images are acquired. In addition to the static background that reflects the magnitude image (left), three phase images are captured, representing the phase shift in the three principal spatial directions. Phase shifts encode flow velocity in left-right (LR), anterior-posterior (AP) and superior-inferior (SI) direction.

Chapter 3

Mathematical fundamentals

In this chapter, the mathematical basics are introduced that form the ground for the in silico simulations in the scope of this thesis. In this context, two areas of physics are of central importance: fluid and solid mechanics. The main focus is laid on modeling fluid dynamics. As blood flow is driven by the deformation of the myocardial walls, elastomechanics are also covered in a subsequent section in a concise form. An overview of the circulatory system finalizes this chapter. While the third highly relevant area of physics in the context of the cardiac function, electrophysiology, is not directly considered in this work, the reader may be directed to literature for further details [102, 103].

3.1 Fluid mechanics

Fluid mechanics describe the behavior of liquids. For this aim, various field quantities are introduced to fully describe a flow process. For these field quantities, several conservation laws have to be fulfilled at each point in time. While the basics of this branch of science are deduced in many educational books, in this section only a condensed overview is included based on [104–106].

3.1.1 Basic principles

The first basic principle is the law of **conservation of mass**. The quantity of mass has to be conserved at each point in time, meaning that the sum of in- and outflow of mass, as well as mass accumulated in the system has to be constant:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \,. \tag{3.1}$$

Equation 3.1.1 is the continuity equation, where ρ stands for the volumetric mass density and u for the velocity vector. The second term of this equation, the divergence of the flux, can be interpreted as the net loss of mass at a point due to the divergence of the flux. Therefore, since this term transfers quantities from one location to another, it is also called transport term.

With constant density ρ, which corresponds to the assumption of an incompressible fluid, both derivatives vanish and the continuity equation simplifies to

$$\nabla \cdot u = 0 \ . \tag{3.2}$$

The second basic principle is the **conservation of momentum**. It is based on the fundamental principle of Newton's second law and it is deduced following the Gauss theorem as follows:

$$\frac{\partial}{\partial t}(\rho u) + \nabla \cdot (\rho u u) = \rho g + \nabla \cdot \sigma \ , \tag{3.3}$$

with the stress tensor σ and its divergence $\nabla \cdot \sigma$. This equation is also called Cauchy's equation of motion. It relates fluid-particle acceleration to net body forces (first term on right side) and surface forces (second term on right side) on the fluid particle. Depending on the application, additional external force terms f can be added on the right hand side. Assuming again a constant density ρ and neglecting the influence of gravity g due to the small height differences in the heart, to a simplified form is obtained:

$$\rho(\frac{\partial u}{\partial t} + (u \cdot \nabla)u) = \nabla \cdot \sigma \ . \tag{3.4}$$

In general, the stress tensor σ can be split up in two parts: The influence of the static pressure, as well as the deviatoric part τ following

$$\sigma = -pI + \tau \ . \tag{3.5}$$

Newtonian fluids are characterized by a linear viscous behavior, which means that the shear stress τ is linearly proportional to the shear rate $\frac{d}{dt}\gamma$. This ends up in the Newtonian law of viscosity:

$$\tau = \mu \cdot \frac{d}{dt}\gamma \ , \tag{3.6}$$

in which the constant proportionality factor μ is called dynamic viscosity. This is also referred to as the constitutive equation for Newtonian fluids. Thus, the stress tensor for an incompressible Newtonian fluid results as

$$\sigma = -pI + \mu(\nabla u + (\nabla u)^T) \ . \tag{3.7}$$

In all projects of this thesis, blood flow was modeled as laminar and incompressible like previously reported in [107]. Even though blood is physically a suspension of particles in a fluid (see also section 2.1) and thus blood flow actually is a multi phase flow, modeling blood as a Newtonian fluid results in bulk velocity differences below 3 % in the large vessels because of the high percentage of water [107]. So the blood flow throughout the whole heart can be modeled as a Newtonian fluid, as also common practice in literature [12].

3.1.2 Navier Stokes Equation

Using equation 3.4 in combination with the Newtonian constitutive law (equation 3.7) and considering external forces f, the basic form of the Navier Stokes equation (NSE) results:

$$\frac{\partial u}{\partial t} + (u \cdot \nabla) u = -\frac{1}{\rho} \nabla p + \frac{\mu}{\rho} \nabla^2 u + f \ , \tag{3.8}$$

which is one of the most famous fluid dynamics equations. It includes the velocity vector u, the constant density ρ, the static pressure p, the dynamic viscosity μ and the external body forces f. These external forces offer a huge flexibility in computational modeling, as they can be used, for example to force different local conditions, as seen later on, e.g., in the context of valve or boundary modeling.

Solving the NSE in combination with the continuity equation (equation 3.2) delivers the velocity vector field u as well as the scalar pressure field p.

3.1.3 Boundary conditions

To calculate a specific solution, every partial differential equation needs boundary conditions (BC)s. Therefore, computational simulations start from the relaxed state, in which the cardiac wall does not have any initial velocity assigned. The flow (or other relevant quantities) through the computational cells located at the edge of the grid requires special treatment. As there are no further points in which equations can be computed, the relevant quantities must be computed based on extrapolation or on one-sided differences. Additionally, the flow at the boundary faces have to be either known (prescribed) or composed of internal parameters. Thus, either the values are prescribed (Dirichlet BC) or their gradient (Neumann BC) in a specific direction (mostly surface normal direction) is presumed. In practice, most commonly convective flows (at in- or outflow patches) or a zero velocity BC are assumed at the walls.

3.2 Numerical fluid dynamics

As the previously deduced coupled nonlinear partial differential equations in most pratical situations can't be solved analytically, an approximated solution has to be found instead. With the help of computational fluid dynamics (CFD), such a solution of the set of governing algebraic equations can be computed. For this aim, the continuous problem has to be discretized in space and in time.

3.2.1 Basic principles

The continuous domain is therefore converted to a computational grid with a finite number of grid points and volumes, allowing for the definition of relevant quantities such as pressure at each grid point. The continuous variables are thus transformed to discrete ones, e.g. the continuous pressure p is then defined at a discrete point or center of a control volume (also referred to as a cell), resulting with the pressure denoted as p_i.

On the way to a discretized representation, the continuous NSE representation equation 3.8 can be transformed into index notation

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\partial}{x_j}\left(\frac{\mu}{\rho}\frac{\partial u_i}{\partial x_j}\right) + f_i \; , \tag{3.9}$$

with the velocity vector u_i, the Cartesian coordinates x_i, which denote the three coordinate directions, the constant density ρ, the static pressure p, the dynamic viscosity μ and the external body forces f_i. The continuity equation from equation 3.2 results as

$$\frac{\partial u_i}{\partial x_i} = 0 \; . \tag{3.10}$$

In the scope of this thesis, the blood density $\rho = 1055\,\mathrm{kgm^{-3}}$ and the dynamic blood viscosity $\mu = 0.004\,\mathrm{kgm^{-1}s^{-1}}$ are applied as in [40, 107].

For the computation of the discrete solution, mainly three numerical methods are used: Finite Difference Method, Finite Element Method and **Finite Volume (FV) Method**. In the scope of this thesis, the latter is applied. To solve the governing equations, the computational grid is discretized in non-overlapping control volumes (mesh elements). All conservation laws are applied to each volume. The computational node is modeled in the center of each volume: One algebraic equation for each control volume is computed as the result of its own variable values and the impact of neighboring nodes. The integral form of the conservation laws is used.

In distinction to this, using the Finite Difference Method does not rely on control volumes. One algebraic equation is evaluated per grid point. The Finite Element Method is quite similar to the FV Method but makes use of a weight function before the integration across the whole computational domain is computed.

In the 2D case, the net flow through a control element equals the integral over the four sides (in case of tetrahedral elements) shared with the other neighboring elements. Computing the solutions of the algebraic equations based on the FV Method requires an approximation scheme for this surface integral. The easiest way is to approximate the integral by the mean value for the side, with, e.g., the overall flow at point e is approximated by

$$Q_e = \int_{S_e} q \, dS = \bar{q}_e S_e \approx q_e S_e \; , \tag{3.11}$$

with $\bar{q}_e$ denoting the mean flow. Of course, more complex and more accurate formulations can be applied. This can be directly transferred to the 3D case and for the volume $Q_p \approx q_P \Delta V$.

The most basic and most known method to solve systems of linear equations is the Gaussian elimination scheme. However, the computational costs of this **direct solver** rise significantly with the number of equations n ($\sim n^3$) due to the costly forward elimination scheme. LU decomposition as well as various other techniques have been proposed to improve. However, the computed triangular matrices are not sparse and thus computational cost remains high. Also, the discretization error is clearly lower than the discretization length itself, which is a non necessary accuracy. Therefore, **iterative solvers**, which are the method of choice in nonlinear systems anyway can drastically improve computational cost while still keeping a sufficient accuracy. In this method, an approximated solution is computed. This approximated solution is then used to systematically improve the solution accuracy until a certain threshold (in accuracy or number of iterations) is reached. For an effective procedure, each iteration has to be cheap in terms of computational cost and effort in combination with the fast convergence of the method.

As discretization affects not only spatial but also time scale, all time-dependent equations have to be solved (time) step by (time) step. Compared to the analytical solution in the continuous case, depending on the approximation scheme, **implicit** and **explicit schemes** can be distinguished in the discrete case. While in an explicit scheme only the current state is used to compute the system state of the following time step, data needs to be extrapolated: $y_{n+1} = y_n + c \cdot \frac{dy}{dt}|_{(y_n,t_n)}$. This yields to good results, only if the time step is sufficiently small and it holds the risk of the solver to diverge. The error increases exponentially with the time step width. On the other hand, implicit solvers also consider the future state of the system for the computation of this future state: $y_{n+1} = y_n + c \cdot \frac{dy}{dt}|_{(y_{n+1},t_{n+1})}$. This results in high computational cost per time step. However, the benefits coming along with it are worth the effort: In this case, the method allows larger time increments, convinces with a high stability as well as high convergence rate and accuracy.

3.2.2 Mesh movement

An additional challenge while modeling the heart in silico is the periodical beating and the resulting movement of the mesh. To model this dynamic behavior, two basic approaches can be distinguished: Either the mesh, more concretely the mesh points move or while the mesh stays static, a boundary interface moves.

To model a moving boundary, the wall velocity can be considered as an additional term in the Navier Stokes equation. To account for the movement, the system of equations is solved in an Arbitrary Lagrangian Eulerian (ALE) framework, where the mesh motion velocity c_i is included in the NSE [108]. This yields the equation

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial (u_i - c_i)}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \frac{\mu}{\rho}\left(\frac{\partial^2 u_i}{\partial x_j \partial x_j}\right), \tag{3.12}$$

in which blood material parameters are chosen equal to those previously reported in [107]. The index i again represents the respective concerning coordinate direction. This is a straightforward implementation with a preserved number of mesh points. However, the displacement of the nodes may result in large differences in cell size of the volume elements and extremely distorted elements that lead to numerical instability. As a compromise, an additional Poisson equation can be solved to equally distribute the node displacement also to nodes located in the inner region of the geometry. Nevertheless, the risk of highly deformed or even corrupted volume elements remains a challenge, especially in highly deformed geometries, as it is the case in the left ventricle (LV). An alternative to counteract highly deformed cells is provided by the procedure of remeshing the geometry from time to time. This would significantly increase mesh quality, but results to additional computational effort and poses high requirements to ensure mesh continuity.

Chapter 4 presents an alternative method: While the mesh remains completely static throughout the simulation, a time-varying indicator function can specify the location of the wall. For a static wall, the normal component of the velocity at that interface region has to be forced to zero. This method attracts as no changes in the grid have to be applied, a simple, structured Cartesian grid can be used and wall velocity is considered as an external forcing term in the NSE. This method is therefore highly robust but challenges introduced by the method (e.g., the possible leaking of flow through the walls) have to be assessed.

3.3 Mesh representation

As described in section 3.2, the continuous geometry has to be discretized in space to be processed in CFD tools. A discrete grid results, which splits up the computational domain in small control volumes. Following the FV method, the governing equations in the integral form are solved for each of these control volumes to obtain the desired quantities of interest.

For modeling the required mesh, two basic representations can be distinguished: Structured or unstructured grids, as visualized in figure 3.1.

The **structured grid** is the most simple structure as it is topologically easy and similar to the Cartesian grid structure. In cubic grids all nodes are equidistant from one another and a straight-forward labeling of the 3D geometry in the coordinate directions is possible. Therefore, the coordinates of a node can be calculated from the indices. This type of grid is characterized thereby that each set of lines does only cross other lines once. The main drawback of this structure is the missing option to realize a mesh refinement in important

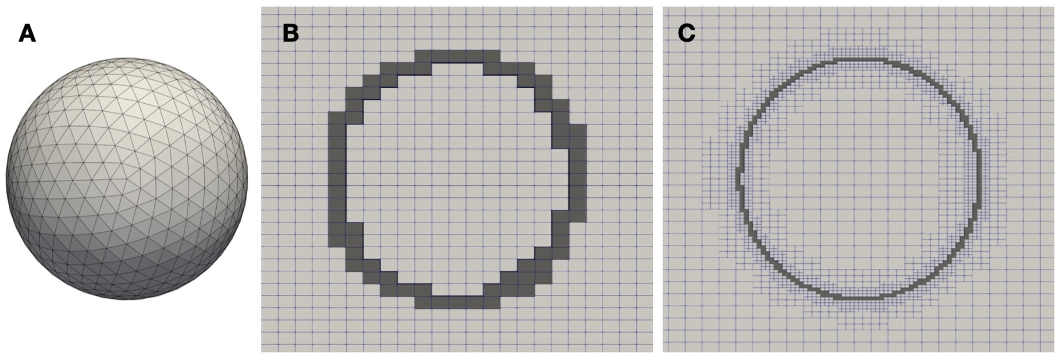


Figure 3.1: Left: Body fitted mesh of a sphere consisting of tetrahedral finite elements. Center: Same sphere projected on a structured hexahedron block mesh (light gray) with the corresponding wall boundary area (dark gray). Right: Block structured grid with refined structure at the border of the interface.

regions. That is why a specialized type of this grid is the block structured grid: Here, a separation of the grid in different areas can be realized, such that more relevant structures such as a wall can be resolved with a finer grid.

In comparison, the **unstructured grid** convinces with an optimal surface modeling. While the grid points are arranged such that the shape of the modeled geometry is matched, the labeling of the elements does not follow a structured order. This results in a higher computational effort: As the matrix of the algebraic equation system is not in a diagonal structure because the points do not have a predefined ordered structure, the solver is slower. While also for structured grids the matrix is not diagonal, the corresponding neighbors of one node can easily be found. However, the unstructured grid is the most flexible one, especially for modeling complicated geometries: For a smooth shape of the geometry, mostly triangular surfaces are used and volume elements consequently are modeled as tetrahedra.

3.4 Structural mechanics

The second area of physics that plays a central role in modeling the cardiac function is elastomechanics, which is responsible for reproducing the deformation of the myocardial tissue. Since the details of the mechanical solver itself did not play a key role in this project, only a concise overview is included. For further basics on the structural mechanics the reader may be directed to [25, 109–112].

3.4.1 Balance of force

A central aspect of mechanical modeling is the displacement d of the tissue at each point of the cardiac cycle. To compute this displacement, the equilibrium condition for the balance of forces has to be fulfilled: At each point in time, external and internal forces acting on the heart tissue have to be balanced. This results in the mechanical differential equation

$$\mathbf{M}\ddot{\mathbf{d}} + f^{int}(d(t),t) + f^{ext}(d(t),t) + \mathbf{C}\dot{\mathbf{d}} = 0 \; . \tag{3.13}$$

with the mass matrix M, the displacement vector d, damping matrix C as well as the internal and external (nodal) forces. For the damping matrix, Rayleigh damping following

$$C = \alpha M + \beta \nabla f^{int} \tag{3.14}$$

with the damping parameters α and β is applied. In the context of this work, mass inertia is neglected, with $\mathbf{M}\ddot{\mathbf{d}} = 0$ causing the first term in equation 3.13 to vanish.

External forces that are considered in the model are the pressure in each heart chamber as well as the forces resulting from the surrounding pericardial tissue. The relevant internal forces comprise the passive material behavior and the influence of active forces. All of the forces are shortly introduced in the following.

To simulate the dynamic behavior in the mechanical solver, time integration is implemented based on the Newmark-Beta scheme [109].

3.4.2 Passive material model

As introduced in [40], the myocardial tissue of both ventricular chambers was modeled as a hyperelastic, nearly incompressible material with the constitutive relation introduced by Guccione et al. [113]. The strain-energy function W follows equation 3.15 with the indices f for fiber, s for sheet and n for sheet normal direction and the four constants C, b_{ff}, b_{xx} and b_{fx}. Incompressibility was enforced with a penalty formulation related to the Jacobian of the deformation gradient $J = \det(F)$ scaled by a bulk modulus $\kappa \gg 0\,\text{kPa}$.

$$W = \frac{C}{2}(e^{Q} - 1) + \frac{\kappa}{2}(J-1)^2 \tag{3.15}$$

$$Q = b_{ff}E_{ff}^2 + b_{xx}(E_{nn}^2 + E_{ss}^2 + E_{ns}^2 + E_{sn}^2) + b_{fx}(E_{fn}^2 + E_{nf}^2 + E_{fs}^2 + E_{sf}^2)\ , \tag{3.16}$$

with $C = 162.78$, $b_{ff} = 11.01$, $b_{xx} = 4.4$ and $b_{fx} = 7.71$ as default values.

Fiber directions are assigned by the Laplace Dirichlet Rule-Based (LDRB) algorithm proposed by Bayer et al. [114]. Fiber angles are $+60°$ on the endocardial wall and $-60°$ on the epicardial wall.

3.4.3 Active force model

Since no electromechanical coupling is implemented in the projects of this thesis, the spread of activation that usually is computed by the 3D electrophysiology model has to be supplied by an additional mathematical function. The so called 'double Hill model' introduced by Stergiopulos et al. [115] as a mathematical model to quantify active force is applied to trigger contraction and relaxation of the heart tissue.

The relevant free parameters for periodicity, onset time, contraction and relaxation time offsets as well as constants for contraction and relaxation rates are chosen as listed in table 3.1.

The resulting time courses of the normalized active stress in the left atrium (LA) and LV are shown exemplary for one heart cycle in figure 3.2.

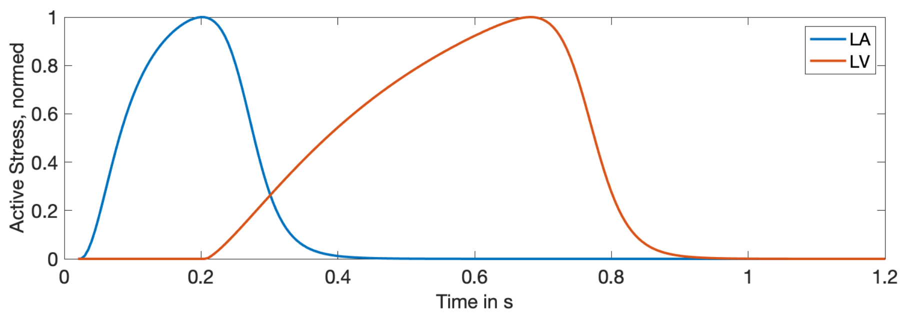


Figure 3.2: Time course of the normalized active stress in LA (blue) and LV (red) resulting from the double Hill function for one heart cycle.

parameter	periodicity in s	onset time in s	contraction time offset in s	relaxation time offset in s	contraction rate	relaxation rate
value LV	1.247	0.184	0.335	0.563	1.32	21.9
value LA	1.247	0.0	0.066	0.251	1.99	11.2

Table 3.1: Parameters set chosen for LA and LV modeled by the double Hill function to compute active strain.

3.4.4 Prestress

As the geometric model is based on measurement data, the problem of computing a stress-free state has to be addressed: While measurement data is acquired in the diastatic state, stresses in the myocardial tissue are minimal, but not zero. Therefore, an unloaded geometry has to be estimated. Assuming that the pressure-free state corresponds to the stress-free state, e.g. an iterative algorithm introduced by Bols et al. [116] can be applied to compute the initial and stress-free geometry. A study on the main influencing factor of this method was published in Brenneisen et al. [117]. Alternatively, estimating the unloaded geometry based on the Klotz curve is possible [118, 119].

3.4.5 Boundary conditions

To reproduce the influence of the surrounding organs and tissue as well as the pericardial sac, which constrains the movement of the heart, a surrounding pericardial layer was included, as proposed in [120]. While the contact surfaces of the epicardial wall and the endocardium are determined, a sliding of both surfaces perpendicular to the normal direction of the layer is possible. If the heart expands, the interface ensures that an external force counteracts the intended movement in a physiological way.

Of course, to deduce in-silico simulations also the continuous computational domain has to be discretized. While a finite element model (FEM) formulation is used and the heart geometry is split up in small tetrahedral volume elements, nodal internal and external forces are computed at every node of the grid. This also results in a discretization of the included vectors, e.g. transforming the continuous displacement d to a discrete vector with an entry for each node d_i. Time step width Δt is adjusted dynamically.

3.5 Circulatory system

As explained in section 2.1 the main task of the cardiovascular system is to pump blood throughout the body. Thus, to physiologically reproduce the whole cardiac function, the four heart cavities are complemented by a model of the systemic and the pulmonary circulation. As common practice in literature the circulatory system is modeled by a 0D model. The closed-loop lumped element model, in detail described in [25, 112, 121, 122] is strongly coupled to the 3D FEM simulation: Pressure and volume are balanced in an iterative procedure in each time step. Volumetric changes resulting from the deformations in the mechanical solver step serve as a BC to the circulation. Balancing variations of the chamber volumes, a mean spatial pressure is computed for each heart chamber, such that the system is in balance. In the subsequent time step, the pressure again serves as an external force acting on the endocardial surfaces that is considered by the mechanical solver. In this way, the circulatory system with its Windkessel elements reproduces the pre- and after-load of the heart chambers.

A schematic overview of the closed-loop system is provided in figure 3.3 with the resistances *R* and the static or dynamic compliance *C*, as well as the nonlinearly modeled valves.

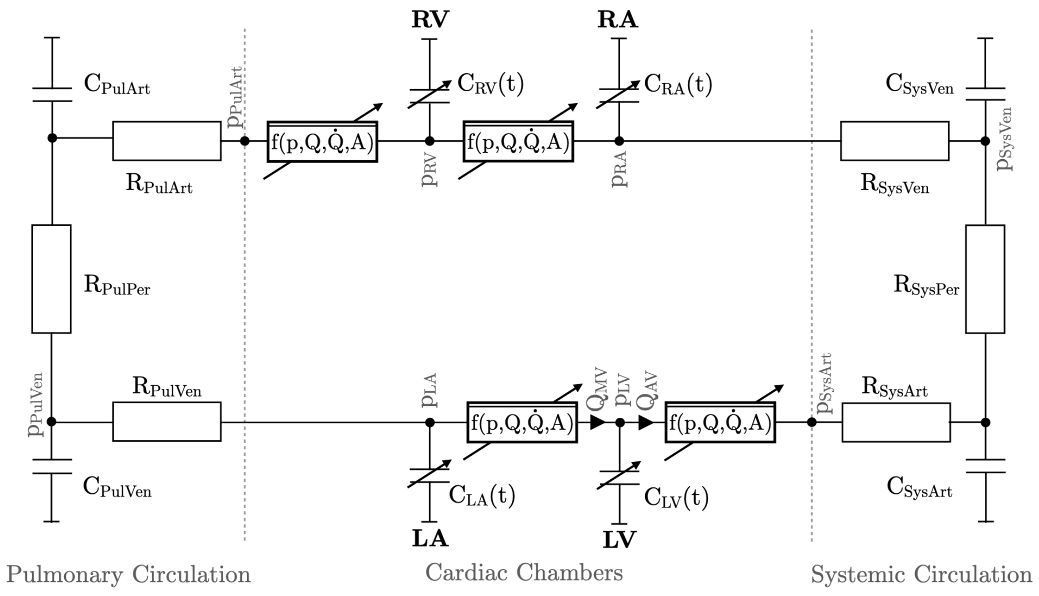


Figure 3.3: 3D-0D coupled model of the circulatory system.

Volume changes are caused by the balance of in- and outflow of blood to a chamber, exemplary for the LV:

$$\frac{dV_{LV}}{dt} = Q_{MV} - Q_{AV} \tag{3.17}$$

Exemplary the flow Q_{SysVen} which quantifies the amount of blood transported from the systemic circulation into the right atrium (RA) is computed based on the pressure differences:

$$Q_{SysVen} = \frac{p_{SysVen} - p_{RA}}{R_{SysVen}} \tag{3.18}$$

All parameters chosen for the individual elements are listed in appendix B.

A key element in the closed-loop circulatory system model are the four cardiac valves. As the remaining part of the circulatory system they all are modeled as 0D elements. However, their behavior is nonlinear and varies over time. The pressure drop across the valve is accordingly computed depending on the respective flow Q, its time derivative $\dot{Q}$ and the valve area A by a differential equation:

$$\Delta p_{net} = \alpha\rho \cdot A_{Eff}^{-\beta} \cdot \frac{\partial Q}{\partial t} + \frac{1}{2} \cdot \rho (\frac{1}{A_{Eff}})^2 \cdot Q^2 \ , \tag{3.19}$$

using the effective area A_{eff} calculated as

$$A_{eff} = \frac{A_{EO} \cdot A_{ref}}{A_{ref} - A_{EO}} \ . \tag{3.20}$$

This 0D valve model was introduced by Garcia et al. [123] and results from the flow conditions in the schematic overview in figure 3.4. Due to the narrowing by the reduced valve diameter, a speedup of the flow can be observed and the static pressure is reduced. A transvalvular pressure gradient Δp arises in dependence of the distance to the valve. Due to the turbulence occurring behind the valve, the pressure does not return to the state before the valve, an overall pressure drop remains due to the loss of energy.

The assumed areas for the different valves are parameterized as listed in table 3.2. While $A_{EO,min}$ denotes the surface if the valve is closed, the open valve state corresponds to $A_{EO,max}$. While in the initial publication, only the parameters for the aortic valve have a physiological basis on measured data, the parameters for the mitral valve (MV) were adapted based on fluid simulations in the scope of the student thesis [124]

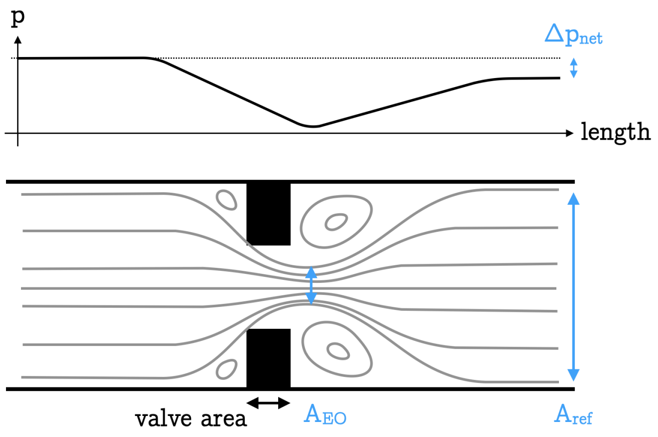


Figure 3.4: Schematic overview of the quantities included in the 0D valve model in accordance with [123]: Blood accelerates while passing by the narrow valve region. Thus, the static blood pressure which is plotted in the upper part of the images reduces and retains a pressure drop.

3.6 Fluid and solid mechanics interaction

As different areas of physics are united and interdependent to fully reproduce the cardiac function, the interaction among them has to be computed. A special focus is on the interaction between fluid and elastomechanics, the so called fluid-structure interaction (FSI), which is discussed in many review articles [47, 49].

Two basic approaches can be distinguished: The monolithic and the partitioned approach. Following the **monolithic approach**, fluid and solid mechanics equations are computed in the same mathematical framework, meaning they form one single system of equations. These equations are solved simultaneously by a single solver. This realization captivates with a high accuracy, but is bought with high computational effort as well as a specialized code that needs great expertise.

In contrast, fluid and elastomechanics are computed in two separate computational fields in a **partitioned approach**. Individual meshes with individual resolution as well as numerical algorithms can be implemented - thus independent solvers for each area of physics can be used. In order to model the interaction of the two solvers, an interface must be created to exchange information among them. This communication results in additional effort but because of the independent solver structure, a clear overall benefit in computational effort is achieved. Depending on the implementation of the coupling interface, the partitioned approach can be further classified in a **strong coupling** and a loose coupling approach. In the former, information is exchanged in an iterative manner in each time step, such that a balance between the influences resulting of both areas of physics is reached. In the **loose coupling** approach the exchange of the coupling quantities is less often, e.g., with an independent coupling step size or also after a whole heart cycle.

Depending on the exchange of information, **unidirectional** approaches in which information only is communicated from one area of physics to the other without any feedback is differentiated from a **bidirectionally coupled** system in which the loop is closed and information is handed back from one solver to the other after a certain time series is computed. In chapter 14 the influence of such a bidirectionally coupled approach is investigated and compared to a unidirectional one. In the remaining part of this thesis, a loose coupling approach with a unidirectional flow of exchange of information is investigated for the sake of simplicity. Therefore, the movement of the endocardial surfaces is computed by the existing mechanical solver framework for a few heart cycles. After the dynamic processes have decayed and a steady state is reached, the deformation of the walls is extracted and transferred to the fluid solver where it is considered as a BC. In addition, the pressures

Surface in cm^2	**aortic valve**	**mitral valve**	**pulmonary valve**	**tricuspid valve**
A_{ref}	7	15	7	15
$A_{EO,max}$	6.65	10.5	6.86	10.5
$A_{EO,min}$	0.007	0.015	0.007	0.015

Table 3.2: Reference and effective opening surfaces of the four valves in cm^2.

computed by the circulatory system are extracted for the inlet surfaces in the pulmonary veins, as well as the outlet pressure in the aorta.

PART II

FLUID DYNAMICS SOLVER

Chapter 4

IBM solver implementation

4.1 Introduction and motivation

Simulating the cardiac function based on a tetrahedral element mesh implementation, computational modeling faces the problem of the partially large deformation of grid cells. Especially in the systole, when the left ventricle (LV) contracts and significantly reduces its volume, cells are deformed to a high extent and low quality cells result. As experienced in the projects of this thesis, this leads to instabilities of the algorithm and the robustness of the method is not ensured any more. One possible alternative is delivered by the immersed boundary method (IBM). Without the need for remeshing or issues with highly deformed cells, the IBM approach is able to provide a more robust representation of the geometry and its boundaries. While the IBM is a method that has existed for many years and has been applied successfully in different fields of engineering, in cardiac modeling it is mainly used to account for the movement of the mitral valve (MV) throughout the cardiac cycle. In this context, it is often combined with regular body fitted mesh approaches. However, IBM comes along with a number of convincing benefits which all concern the simplicity of the mesh. While the moving surfaces are immersed into the static mesh, no remeshing is required and using a structured grid is permitted. Structured grids allow for a high efficiency while solving the equation in every time step. Moreover, the preprocessing workload is small as no complicated meshes have to be set up. Thus, using the IBM complex moving geometries can be modeled.

In the context of this thesis, it is investigated to which extent the IBM is capable of modeling a realistic whole heart geometry. Therefore, the IBM solver is introduced here (chapter 4), in chapters 8 to 9 different verification scenarios in 2D and 3D geometries are conducted in several computational experiments. In chapter 11 the solver is validated based on measurement data.

Parts of the content of this chapter are adapted from the conference publication [125] under the Creative Commons Attribution 4.0 (CC BY 4.0) international license.

Fluid dynamics simulations are executed in the open-source software framework openFOAM under the version 7 [2] based on an IBM implementation by Müller [126].

4.2 Methods

The central characteristic of the IBM approach is that the boundary of the computational grid not necessarily equals the fluid boundary. Therefore, the fluid boundary is prescribed by an additional wall inside the computational domain. To force the velocity in the introduced wall to meet the desired, prescribed value, the force term f_i in the Navier Stokes equation (NSE), (equation 3.9) is set [62]. The body force necessary to enforce the velocity of the wall $U_{\mathrm{w},i}$ is computed based on the constants P_c and I_c which act as proportional and integral controllers as

$$f_i = \frac{\partial \Phi}{\partial \Lambda} \left[P_{\mathrm{c}} \left(u_i - U_{\mathrm{w},i} \right) + I_{\mathrm{c}} \int_{t_0}^{t} \left(u_i - U_{\mathrm{w},i} \right) d\tau \right] . \tag{4.1}$$

The correct choice of the controller parameters forces the blood velocity to meet the prescribed wall velocity. For this approach a simple, static and well structured hexahedron Cartesian grid can be used, either structured or block structured as introduced in section 3.3.

To correctly prescribe the wall velocity $U_{\mathrm{w},i}$, a smoothed indicator field Φ is introduced [63] for all cells, acting like a mask function to indicate the wall location following the equation

$$\Phi = \begin{cases} 1 & \text{at the exact position of the wall} \\ 0 < \Phi < 1 & \text{for a thin interfacial region, where} |\Lambda| < w \\ 0 & \text{everywhere else.} \end{cases} \tag{4.2}$$

In equation 4.2, Λ is the distance from the interface. To compute this distance, a continuous distance field that contains the euclidean distance from each cell center to the closest point of the immersed wall has to be computed. The parameter w represents the thickness of a narrow region close to the solid-fluid-boundary. The indicator field Φ is computed from the distance field Λ by applying a cosine function under consideration of the width parameter w:

$$\Phi_i = \begin{cases} 0 & \Lambda > w \\ \cos(\frac{\Lambda_i}{w} \cdot \frac{\pi}{2}) & \text{else} \end{cases} \tag{4.3}$$

This straightforward procedure of first computing the distance field, then computing the indicator field and with that weighing the wall velocity field already succeeds in the static case. In the more relevant dynamic case, which is indispensable to reproduce any movement over time, the distance field has to be shifted in each time step. In a simple 2D case this is rather straightforward to implement: The movement of a circular region can be computed for each point in time as the result of a sine function, e.g. the radius r resulting as $r = r_0 + dr \cdot \sin(\omega t)$ with a certain angular frequency ω. From this function the distance field Λ can directly be computed for each cell center. In more complex scenarios the distance field cannot be deduced by computing the distance from an analytical function, but prescription can be done via surface files (e.g. in the stl format). For such a constraining surface the shortest euclidean distance to each cell has to be computed.

The most simple approach is to find the nearest neighbor for each cell center. However, especially if the structured grid is smaller than the unstructured one that is used as the constraining surface (which typically is the case), an introduced error may be the result for the cell centers located on the connecting line between two points or rather close to it (compare figure 4.1). Therefore, additionally the distance between the grid point $\mathbf{x}_0$ and the vertex that connects the two closest points $\mathbf{x}_1$ and $\mathbf{x}_2$ can be evaluated. The geometrical setup that illustrates the situation for the relevant equation is shown in figure 5.4.

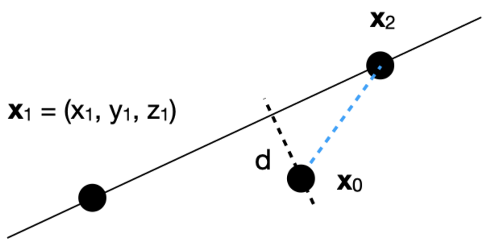


Figure 4.1: Minimum distance from point to line (dashed black line). Lower than vertex to vertex distance (dashed blue line).

In figure 4.1, d is the distance between the point x_0 and the line connecting the two closest points of the geometry and results as

$$d^2 = [(x_1 - x_0) + a \cdot (x_2 - x_1)]^2 + [(y_1 - y_0) + a \cdot (y_2 - y_1)]^2 + [(z_1 - z_0) + a \cdot (z_2 - z_1)]^2 \quad (4.4)$$

using the factor a resulting from

$$a = -\frac{(x_1 - x_0) \cdot (x_2 - x_1)}{|x_2 - x_1|^2} \quad (4.5)$$

with the $\cdot$ in the numerator referring to the scalar (dot) product.

Following these equations, the distance field is computed for each update time step dt_u. This additional time step is introduced to mark the time step in which an external surface file is read and the new distance field Λ_t is computed. It is a multiple of the 'normal' solver time step dt and computed based on the number of available geometries n_e for the duration of a complete heart cycle cl. In every solver time step dt it is evaluated whether the geometry has to be updated. This is the case if the current and the previous stl step are different:

$$n_{STL,t} = \text{round}(\frac{t}{cl \cdot n_e}) \quad (4.6)$$

$$n_{STL,t-1} = \text{round}(\frac{t - dt}{cl \cdot n_e}) \quad (4.7)$$

If $n_{STL,t} \neq n_{STL,t-1}$, a new distance field is computed.

In addition to the current distance field Λ_t the upcoming update step's distance field Λ_{t+dt_u} is computed. To cover all solver steps in between, the distance field is linearly interpolated between the two processed points in time by weighing the computed distance fields Λ_t (current) and Λ_{t-dt_u} (previous) with the fraction of time that has passed since the last update step:

$$\Lambda_t = \Lambda_{t-dt_u} + (n_{\mathrm{STL},t} + 0.5) \cdot \frac{\frac{cl}{n_\mathrm{e}} - t}{\frac{cl}{n_\mathrm{e}}} \cdot (\Lambda_t - \Lambda_{t-dt_u}) \tag{4.8}$$

Applying this linear interpolation, computational resources are saved. While geometrical changes between the update time steps are small, the introduced error is low. Following equation 4.2 the indicator field Φ is updated based on the distance field Λ in every solver time step.

While the scalar indicator field Φ denotes which elements are part of the wall and therefore are constrained by the controller, the vector field $U_{\mathrm{w},i}$ denotes which velocity is assumed for the corresponding cells. Following the approach of prescribing wall motion by external geometries, the wall velocity can also be computed from the constraining surface (stl) files. To obtain the velocities in the three coordinate directions, the nodes of the current and the upcoming update time step dt_u are matched. This is necessary, if it is not assumed that the order and labeling of the constraining mesh points is unchanged. Even the number of points are allowed to vary in this case. In the most simple strategy, again the nearest neighbor can be determined and the vector between both of them $\mathbf{d} = \mathbf{x}_2 - \mathbf{x}_1$ can serve as the input for the computation of the wall velocity field. As this again leads to interpolation errors, especially for control volumes near edges and bended areas, alternatively the point in the center of the two closest points can be examined. If a lower distance is obtained, the optimization is applied. The wall velocity $U_{\mathrm{w},i}$ can be obtained by a division of the distance to the closest point by the time step size following

$$U_{w,y} = \frac{y_{t+1} - y_t}{dt} \ . \tag{4.9}$$

4.3 Results

To demonstrate the functionality of the suggested procedure, to systematically investigate the influence of the parameters introduced and to quantify the impact of different approximation steps listed in section 4.2, resulting fields (distance field Λ, indicator field Φ and wall velocity field U_w) are simulated for a simple 3D sphere immersed into the structured 2D block mesh introduced in figure 3.1.

The prescribed constraining surface geometry can be seen in figure 4.2 (A): Three update geometries are visualized, while the first time step corresponds to the sphere with the smallest volume (most inner). In the subsequent time steps, the radius increases, which models an inflation. For a better visualization of the geometries, the spheres with higher diameter are cut along the y-plane. As described before, the indicator field Φ is computed from the distance field Λ. The results for both are visualized in figure 4.2B and D.

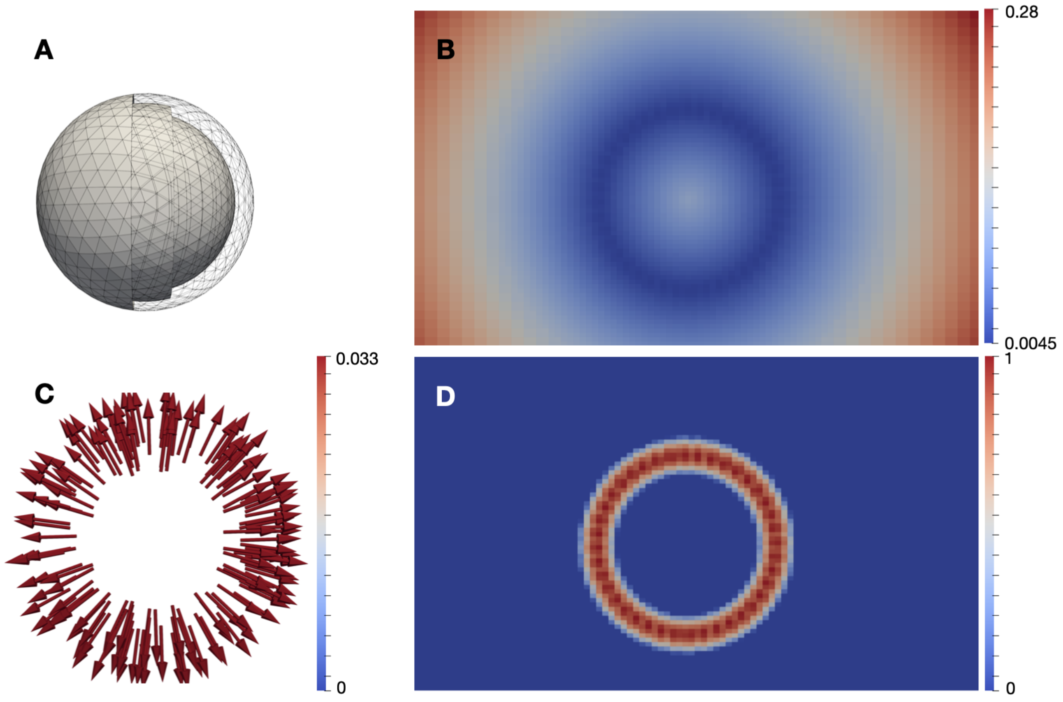


Figure 4.2: Top row: Input geometry (A) for three update time steps which prescribe the wall location and resulting distance field Λ in mm for the first time step (B). Bottom row: Indicator field Φ (D) for the first time step computed from the distance field (B) and wall velocity $U_{\mathrm{w},i}$ in m/s (C) for the first time step resulting from the prescribed geometries.

All result visualizations in figure 4.2B to D show the expected results and clearly symmetric fields. As expected, the distance (B) is minimal at the desired location and increases towards the corners of the simulation grid. The resulting ring structure is reflected by the indicator field in the desired cosine-weighting: The center line is prescribed by a high indicator $\Phi > 0.96$, while a second and third row of cells are added with a reduced indicator value directly attached to the center line. In this way at least two cells in each circumferential direction are covered with an indicator value $\Phi > 0.8$, which ensures that also in the direction diagonal to the grid orientation no flow can penetrate the wall and exit the cylinder. The wall velocity field computed from the constraining input geometries (C) shows a symmetric behavior with velocity vectors of the same length in all directions pointing from the cylinder center outwards (inflation). This is the intended shape and therefore also fulfills the requirements.

In a second simulation, the same simulation setup is used to quantify the impact of the point-to-line interpolation as introduced in figure 4.1. While all parameters are kept constant, solely the interpolation is turned off. Therefore, it allows for a comparison between only point-to-point and point-to-line interpolation in figure 4.3.

While both indicator fields in figure 4.3A and B show the same basic and symmetric structure, a clear difference in smoothness can be observed: As expected using the point-to-

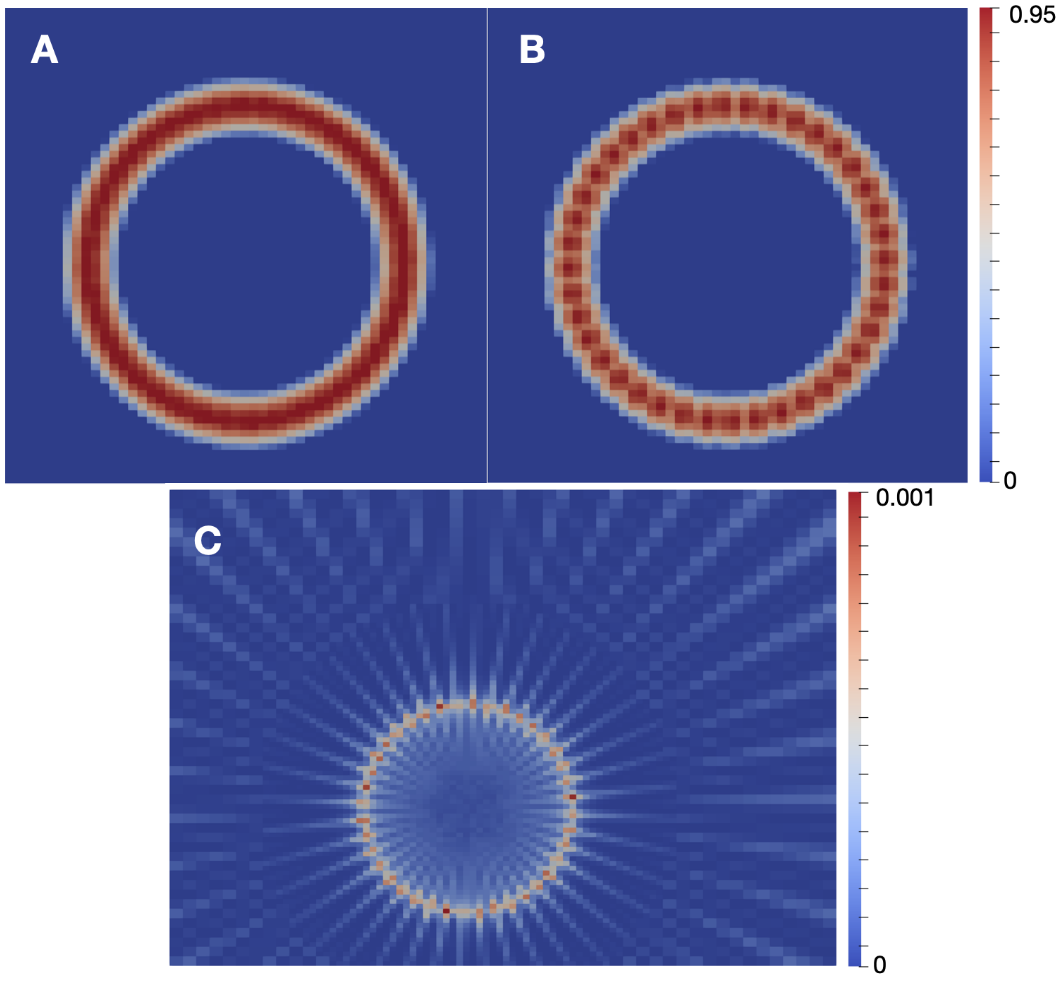


Figure 4.3: Top row: Close-up of the dimensionless indicator field: In the first plot, the connecting vertexes are included during the distance field calculation (A). In contrast thereof, the distance is computed only on the point-to-point distance (B). Bottom row: Difference of both distance fields $\Lambda_C = \Lambda_A - \Lambda_B$ for each control volume in m (C).

point method, a systematic error in the control volumes located exactly or closely between two points of the constraining geometry is introduced. In those elements, it's not the distance to the constraining surface which is computed (which would be 0 mm as the control volume is located on or close to the connecting vertex). Instead the distance between the two closest constraining grid points is used as the basis for calculation and an error up to a deviation of half the mesh resolution of the constraining geometry results. Visually this results in a rippled edge structure following the superposition of the individual circular distances around each point of the constraining geometry. This error continues radially to the edge of the grid as seen in figure 4.3C. Following equation 4.3, for the computation of the indicator field Φ anyway only distances smaller than the threshold w are considered. As the circumferential rays are distant to the center line, they will not have any impact on the indicator field.

Concerning the cells in the proximity of the walls center line, the distance of up to 0.001 m makes only one tenth of the grid edge length in the wall area of 0.01 m. For all cells with an indicator $\Phi > 0.5$ the differences are given in more detail: while the maximum difference equals $\Delta\Lambda_{max} = 0.994$ mm, the mean difference is $\Delta\Lambda_{mean} = 0.336305$ mm. Therefore the introduced error is considered to be neglectable. However, to also exclude influences on the resulting flow quantities the reader has to be directed to chapter 8, where this is investigated on relevant geometries that comprise an inflow and outflow tract.

Finally, these influences of the parameter w are investigated: While w weighs the distance field Λ during the indicator Φ computation following equation 4.3, it thereby determines the thickness of the wall region. Therefore, the identical simulation setup as before is used, resulting in an unchanged distance field. The factor which weighs the thickness of the region w around the wall is decreased to 75 % and increased to 150 % from the standard value of $w_0 = 0.023$ m. to evaluate the influence on the indicator field Φ. In figure 4.4 a close-up of the upper wall region shows the influence of the parameter. The indicator field Φ is visualized for the three named scenarios.

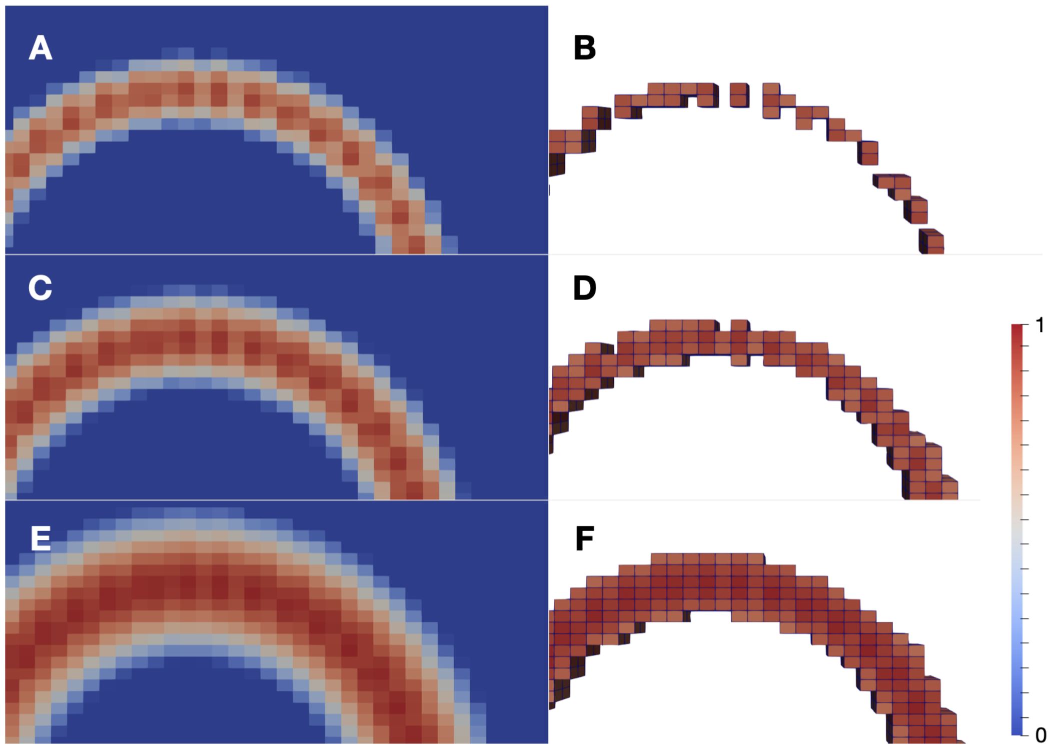


Figure 4.4: Close-up of the indicator field (dimensionless) under variation of the thickness parameter w. In the left column the indicator field for all cells of the close-up is shown. On the right hand side, only the indicator field for cells that have an indicator value that meets $ind > 0.8$ and therefore reliably blocks blood flow is depicted. In the first row Φ results for 0.75w_0 is shown, in the middle row the standard value w_0 is included, and lastly the lower row depicts Φ for $w = 1.5w_0$.

While on the left hand side in figure 4.4 the indicator field is shown for all cells in the close-up, on the right hand side a threshold $\Phi > 0.8$ is applied to show the cells that reliably close the cavity and prevent from leaking flows. It can be seen that after applying the threshold the standard parameter $w = w_0$ delivers a closed ring (D). In comparison, the reduced w results in holes in the ring (B) and the increased parameter leads to a thickening of the wall (F). While this wall thickening prevents a flow throughout the wall, it significantly reduces the remaining internal volume enclosed by the wall. This also illustrates a problem imposed by the procedure: while the prescribed surface is assumed to be the center line of the wall, half the wall thickness is subtracted from the free enclosed volume, because the wall velocity is prescribed. This has to be taken into account when prescribing the endocardial wall.

In table 4.1 the quantitative results for the remaining enclosed surface in dependence of the three wall thickness parameters w is listed. In the first two result columns, only cells that are assigned an indicator value $\Phi = 0$ are counted. The number of cells n_i enclosed by the wall and the resulting enclosed surface A_i is denoted. In the following columns, the condition $\Phi < 0.5$ holds. Additionally, considering the radius of the constraining cylinder $r = 0.1\,\mathrm{m}$ a prescribed surface of $A_0 = 0.031\,\mathrm{m}^2$ is used to compute the remaining fraction.

	$\Phi = 0$	$\Phi = 0$	$\Phi < 0.5$	$\Phi < 0.5$	$\Phi < 0.5$
w	$\mathbf{n_i}$	$\mathbf{A_i}$ in m^2	$\mathbf{n_i}$	$\mathbf{A_i}$ in m^2	$\frac{\mathbf{A_i}}{\mathbf{A_0}}$
$\mathbf{0.75w_0}$	664	0.02324	750	0.02625	83.7 %
$\mathbf{w_0}$	566	0.01981	688	0.02408	76.7 %
$\mathbf{1.5w_0}$	418	0.01463	579	0.02027	64.6 %

Table 4.1: Resulting number of cells n_i enclosed by the indicated wall as well as the resulting enclosed surface area A_i in m^2 for different indicator Φ conditions and varied wall thickness w.

4.4 Discussion

Referring to literature, for a good mesh quality, three main criteria have to be assessed [10]: The dimensionless characteristic factor of the cell's skewness has to be as low as possible. Good values are in a range below 0.5, whereas the skewness at least has to be below 0.98. Additionally, the mesh has to be smooth, thus no jumps are allowed. Finally, the aspect ratio denoting the relation of the side lengths in a cell has to be assessed. Of course, in the optimal case it is balanced. Evaluating all these mesh criteria, a static, structured hexahedron grid offers one of the best possible results in terms of mesh quality. In the body-fitted mesh, especially during the systolic contraction, in which a significant decrease in cell volume occurs, the large deformation of the cells leads to a low mesh quality and even to an unstable algorithm. Thus, the IBM is highly promising in terms of a high quality of the mesh representation. However, this approach introduces the burden of an accurate representation of the wall shape, resulting in higher demands on the grid resolution.

On top, the structured grid representation coming along with the IBM implementation is superior compared to the unstructured mesh in the body-fitted representation. A fast convergence of the solver is achieved [10].

First simulations evaluating the resulting geometric fields proved the expected behavior. While the position of the immersed surface can be deduced by evaluating the distance field Λ based on equation 4.3, this raises the question how to adequately choose the thickness of the wall, which is prescribed by the distance w. As assessed in chapter 9, it proved good results to model the wall with at least four cells. This ensures, that at least two cells with an indicator $\Phi > 0.8$ prevent a flow across the wall through diagonally neighboring cells. This results in an wall thickness chosen based on the grid resolution.

In addition, the reduction of the effective area, respectively of the enclosed volume was observed. In order to actually obtain the prescribed volume, this circumstance has to be considered in the applied geometric surface. Therefore, the prescribed surfaces have to be inflated prior to their application, taking into account the wall thickness parameter w.

Of course, it has to be verified whether the implemented IBM solver is capable of reproducing central fluid dynamics quantities. This is assessed in the chapters 8 to 10.

Chapter 5

Workflow automation

5.1 Introduction

As previously described, the loosely coupled scheme for solid and fluid mechanics interaction requires the transfer of relevant quantities. The main focus in this context is on the geometrical data, more precisely the endocardial surface in the form of a connected mesh for the whole heart side. In this procedure, also the small trunk bases included in the mechanical geometry have to be prolonged to straight tubes as will be discussed in chapter 7. To quickly, robustly and reproducibly compute this fluid boundary mesh, an automated pipeline was implemented and evaluated. In this chapter, the workflow is illustrated exemplary for the left heart. Implemented scripts are attached in appendix A.

5.2 Methods

The structure of the implemented pipeline is modular and comprises three scripts, such that partial functions can also be used separately. Basic data for this algorithm are provided by the mechanical solver. In a first step, a continuous mesh for the initial geometry ($t_0 = 0\,\mathrm{s}$) is processed. As the different tissue areas are modeled by characteristic tissue properties in the mechanical domain, a material index is available to distinguish between them. For most relevant areas, the surfaces can directly be extracted by thresholding this property. For all materials that are only available as volumetric meshes, which are characterized by a material index smaller than 100, the surface is extracted additionally. Relevant indices are listed in table 5.1.

All extractions of the single material classes in correspondence with table 5.1 are automatized in the function *extractFluidSurfaces*. This function, as well as an overview of the whole workflow parts and the included relations and their interdependence are depicted in figure 5.1.

The various exported surfaces serve as an input to the *prolongTrunks* function, which forms the core part of the workflow. In this script, all surfaces are connected to a single

surface name	ventricle	atrioventriuclar plane	atrium	semilunar valve plane	short trunks
material index left heart side	131	37	133	38	38
material index right heart side	130	36	132	38	38

Table 5.1: Material indices of surfaces relevant for fluid geometry creation.

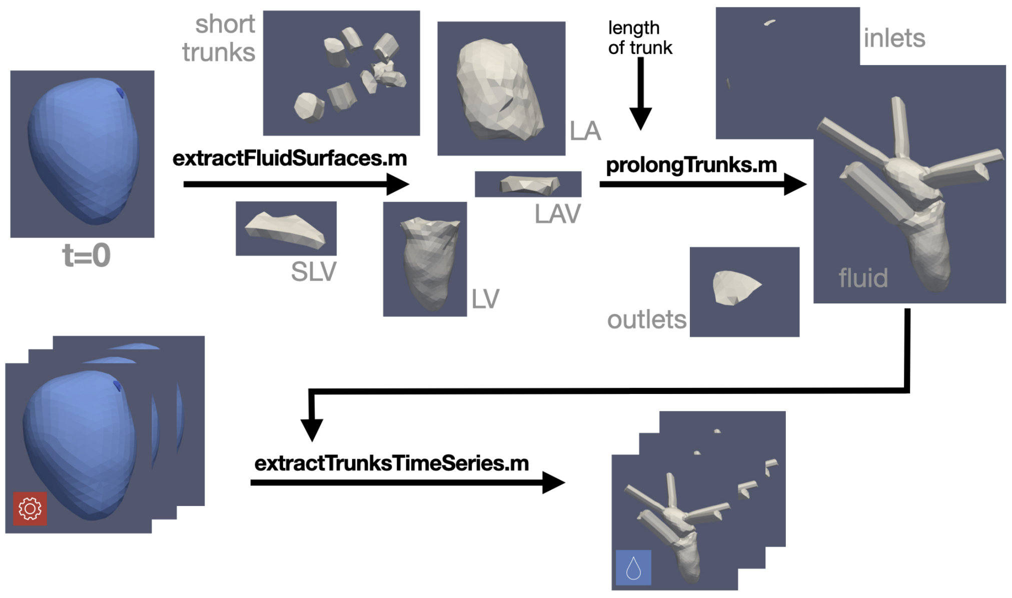


Figure 5.1: Schematic overview of the trunk prolongation algorithm. The three modular functions are sequentially executed one after the other. Input data are directly obtained from the mechanical simulation framework. In the end, a time series of fluid surface files are exported which in the following can be used to constrain the endocardial wall in fluid simulations.

surface, associated inlet and outlet surfaces are exported as separate surface files and the trunks are prolonged. A flow chart of the relevant surfaces and interdependencies is included in figure 5.2.

For this script, the desired length of the trunks as a multiple of the average edge length serves as an additional input. To create the connected mesh, the endocardial surfaces of the ventricle, atrium and atrioventricular plane are merged and duplicated faces at intersecting regions are removed. The short trunks and the semilunar valve plane are first split into a subset comprising all elements which belong to the left heart by searching for intersecting areas. Duplicated faces as well as trunk bases belonging to the other heart side are again removed. In the subsequent step, all trunks are prolonged. While the lower lid is identified based on the contact to the atrial wall, the outer lid is on the opposite side. In comparison to prolonging the trunks in the normal direction of the upper lid, it was found to be way more robust to prolong the trunks in the direction of the vector connecting the mean point of both lids. This is especially the case if upper and lower lid are not in parallel as illustrated

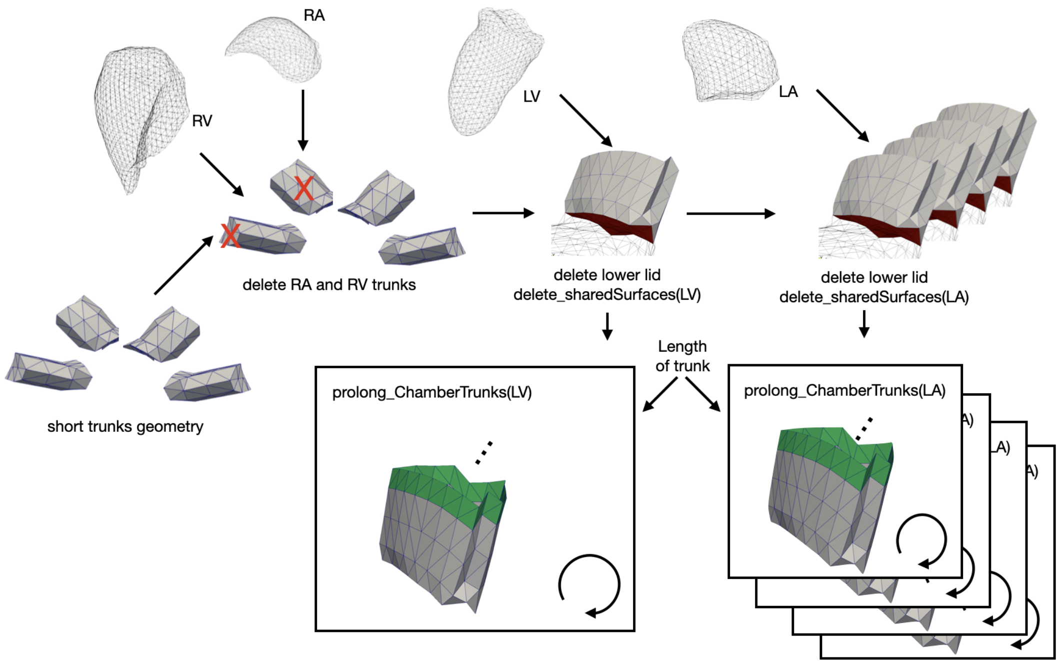

RA
RV
LV
LA
delete RA and RV trunks
short trunks geometry
delete lower lid
delete_sharedSurfaces(LV)
delete lower lid
delete_sharedSurfaces(LA)
Length
of trunk
prolong_ChamberTrunks(LV)
prolong_ChamberTrunks(LA)

Figure 5.2: Schematic flow chart of the sub module to prolong the trunks. Based on the surfaces obtained from the extraction sub module and the desired trunk length, the prolonged trunks for the corresponding atrium and the corresponding ventricle are computed.

in figure 5.3. If simply the upper lid normal vector is chosen, the trunk is bend with a high angle (see center of figure 5.3) and may intersect with other trunks. In the symmetric case, both variants are identical.

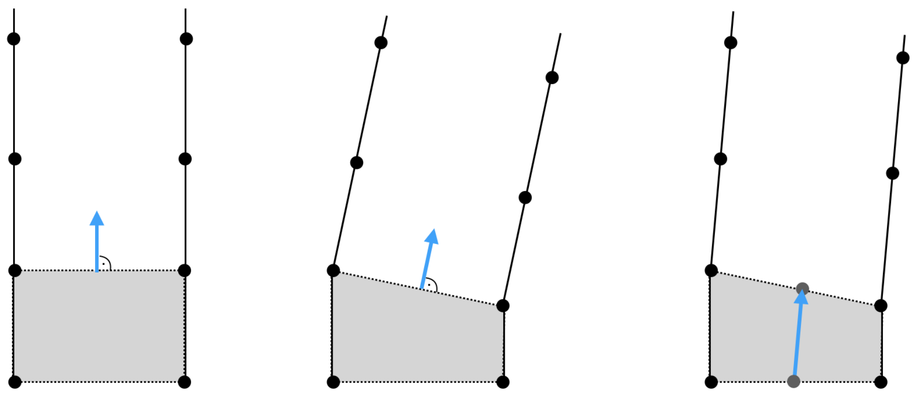
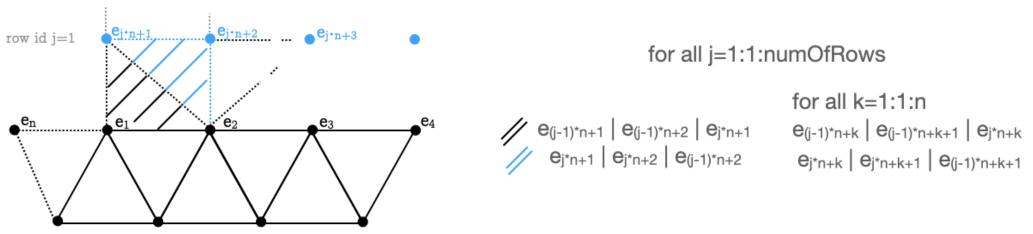


Figure 5.3: Comparison of different possible vectors denoting the prolongation direction of the trunks. Left and center: Normal vector of the upper lid. In asymmetric trunk bases this leads to a bend trunk. Right: Vector connecting lower lid center point with upper lid center point.

For the prolongation of each trunk, the upper lid points are iteratively duplicated with a shift of the average edge length in the direction of the previously calculated direction vector. The length of the trunk is prescribed by the additional length parameter and set in relation to the average edge length of the mechanical surfaces. To create a closed triangular surface mesh, the inserted points are connected by edges. Each three of those edges are grouped as a triangle, following the schematic procedure illustrated in figure 5.4. In this way, for a number of points n in the upper ring of the trunk, $2 \cdot (n-1)$ triangular surface elements are added per row and merged with the existing mesh. This is repeated row by row until the desired length is met.



```
for all j=1:1:numOfRows
    $e_{(j-1)*n+1}$ | $e_{(j-1)*n+2}$ | $e_{j*n+1}$
    $e_{j*n+1}$ | $e_{j*n+2}$ | $e_{(j-1)*n+2}$
    for all k=1:1:n
        $e_{(j-1)*n+k}$ | $e_{(j-1)*n+k+1}$ | $e_{j*n+k}$
        $e_{j*n+k}$ | $e_{j*n+k+1}$ | $e_{(j-1)*n+k+1}$
```

Figure 5.4: Visualization of the trunk prolongation procedure. To create a closed surface mesh, each trunk is prolonged row by row in the direction of the vector determined before as exemplary visualized for the first two faces. Each new node is immediately connected via vertices to form a triangle. The schematic code can be applied to compute triangles in a loop for the number of rows j and the number of points n in the upper lid edge.

Finally, the lid is added at each end of the trunk and exported in a separate surface file for further processing. This procedure is repeated iteratively until all trunks are prolonged in both heart sides.

5.3 Results

The introduced workflow was successfully applied on three whole heart geometries. The algorithm was able to reliably determine the desired fluid surfaces as exemplary visualized in figure 5.5: A left ventricle (LV) of a healthy heart geometry (A) as well as a heart geometry burdened by hypertrophic cardiomyopathy (HCM) (B-D) are included. While the green lids show the inlet surfaces, the outlet boundaries are marked in red color. Besides a HCM LV (D) also a right ventricle (RV) (C) and an exemplary time series (B) consisting of three time steps (diastole, systole and in between) is included.

Regarding computational resources, the trunk prolongation sub module takes approximately 0.5 s to compute all concerning trunk prolongations and export the surfaces. The time extraction for the whole heart cycle time series in the following module needs approximately

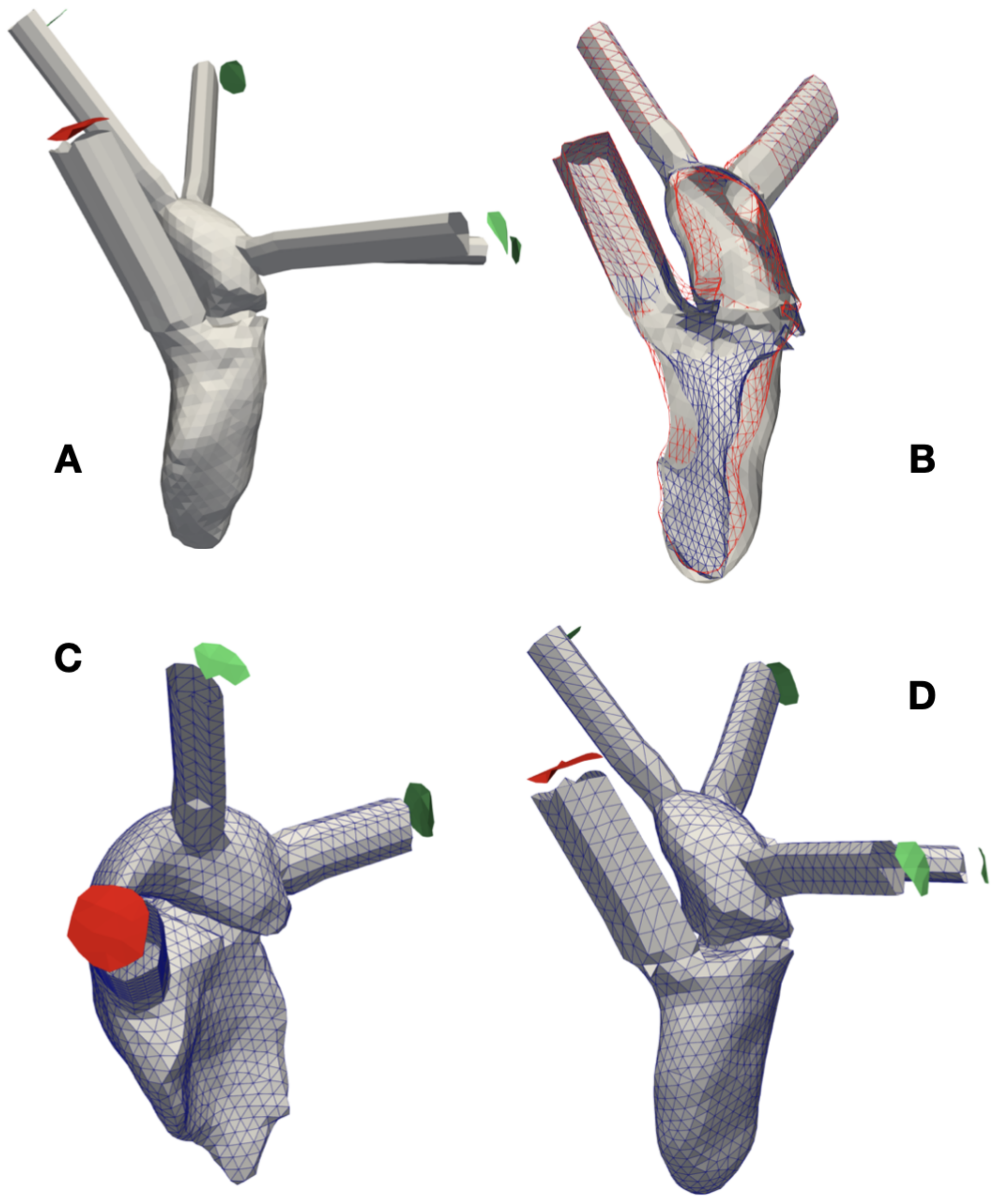


Figure 5.5: Resulting surfaces for healthy LV (A) and diseased case. B-D HCM with C LV, D RV and B time series of LV, smallest corresponds to systolic volume.

0.8 s for each geometry, such that a full heart cycle covered with 80 geometries is computed in below 70 s.

5.4 Discussion

The introduced routine to automatize the setup of the boundary conditions (BC) revealed a fast and successful creation of the constraining endocardial surfaces. The initial geometry, as well as the time series and the lids for the inlet and outlet surface are created in a reliable way. However, main drawback of the implemented procedure of cause is the high dependence on the mechanical mesh availability and quality. As the trunks are just a prolongation of the existing short trunk bases, potential errors would propagate across the whole trunk geometry. Thus, the accurate segmentation of these trunks is of high relevance, however challenging as the segmentation of the relevant regions is error-prone.

In addition, the trunks are simply prolonged as straight tubes, not based on any physiologic structural condition. This of course could be improved, if the concerning geometries would be delivered from measurement data. As this is not the case for the studied geometries, this limitation has to be taken in mind.

The partitioned structure of the algorithm also allows to model data acquired by other modalities than the elastomechanical simulation framework. Exemplary, measurement data may be considered. However, as the pipeline is implemented for the input data resulting of the elastomechanics simulation up to now, the methods to establish correspondence have to be adapted. Rigid alignment methods, such as the iterative closest point algorithm as well as non rigid-body transformations can be applied to establish correspondence. High demands on the alignment of the individual surfaces are required to ensure to obtain an adequate resulting closed surface mesh.

Chapter 6

Valve representation

6.1 Introduction

As they regulate the inflow into the ventricular chambers as well as the ejection of blood out of the ventricle, a central element in the computation of cardiac blood flow dynamics are the valves. Other than in the circulatory system (described in section 3.5), in the fluid simulations, all valves are represented by volumetric structures. Thus, a 3D representation of the geometry has to be modeled. In addition, the time dependent procedure of the opening a closing process has to be modeled. Depending on the chosen implementation of the valve, this temporal controlling during the heart cycle varies. This chapter provides an overview and a comparison of the different valve models utilized in the different simulation studies of this thesis. From the implementation point of view, four different models can be distinguished.

First, the existing implementation of a simplified planar mitral valve (MV) model published in [38] is introduced. This simplified model originally was assessed in an idealized left ventricle (LV) geometry. It proved to reproduce the characteristic asymmetric shape of the vortex ring to a sufficient extent. This valve representation is implemented based on a **porous zone** approach, as well as a simplified 3D valve geometry with only **two states** immersed in the immersed boundary method (IBM) grid. In the scope of this thesis, both realizations of this valve model are integrated into a more realistic framework of an individualized, realistic LV geometry.

In addition, two further representations of the MV have been implemented and evaluated in the scope of a student project [127]. On the one hand, the **Linköping valve** is a realistic 3D valve based on data measured in an animal heart. Based on biplane radiography with radiopaque markers that have been implanted surgically in a sheep heart, Ingels et al. measured the motion sequence of the MV [128]. On the other hand, the St. Jude medical valve is a **prosthetic valve**, that was implanted in a high number of patient since the late 1970 years [129]. Both 3D valves are assessed by immersing them in the IBM grid.

6.2 Methods

6.2.1 Porosity model

In literature, the porous zone approach was introduced to model really small structures that have characteristic edge lengths below the mesh resolution [130]. For such zones, the behavior of a classic sponge is taken as a prototype: While the direct convective flow is obstructed, permeable pores allow for a limited amount of flow to penetrate the membrane structure. The size of the pores determines the amount of flow across the valve and thus by introducing a permeability factor $k_p \in [0,1]$, this approach perfectly fits the demands in valve modeling. While the permeability is prescribed as a mean value for the whole valve, adjusting the permeability of the zone over time can be regarded as a simplified method to model the valve behavior. By the porous structure, not only the flow is decreased depending on the porosity. Also a pressure drop occurs.

To model this porous zone influences, a forcing term F_p is inserted into the Navier Stokes equation (NSE) (equation 3.8). The porosity model forces the velocity within the zone to be equal to the motion of the respective plane and adjusts each time step dependent on c_i:

$$F_p = -\frac{\mu}{\rho}\frac{\phi_p}{k_p} \cdot (u_i - c_i) - \frac{1.75\sqrt{\phi_p}}{\sqrt{150 k_p}} \cdot (u_i - c_i)\,|u_i - c_i| \ , \tag{6.1}$$

considering the density ρ, the kinematic viscosity μ and the porosity $\phi_p = 1$, which is kept constant. The valve plane permeability k_p varies between infinitesimally small (impermeable) and $k_\mathrm{p} = 1$ (permeable). As this valve representation is applied in chapter 14, the time course of the porosity k_p is shown in figure 14.3.

6.2.2 Model geometry and implementation

On the contrary to the porous zone approach that effectively models the valves based on spatially averaging all influences in the time dependent permeability factor k_p, the IBM solver offers the great benefit to enable a spatially resolved model of the valves. Thus, a 3D construction of the valve is possible and a higher level of physiological accuracy can be achieved. To be considered in the IBM solver framework, the geometries of the valve are prescribed by a stack of external surface files. Based on this time course for one full heart cycle, the geometry is updated at the correct point in time.

The first and most simple implementation is to again model the valve based on the planar MV model approach [38]. In this case, the opened and closed state are prescribed by external surface files and immersed into the grid, resulting in an additional indicator field Φ_{MV} for the MV. However, in this case the influence of the MV leaflets is neglected and a infinitesimally rapid opening process is assumed.

To increase accuracy and take benefit of the IBM implementation, more physiologic valves can be considered: With the Linköping valve and the prosthetic valve representation implemented in the scope of the student thesis, two time resolved motion sequences are modeled based on literature data. To fit these models to the individualized LV geometry evaluated in the context of this thesis, an algorithm was developed to fit the valve, located in an arbitrary coordinate system to a predefined position of the mitral annulus ring [127]: First, a plane is fitted through the MV geometry that will be transformed and all corresponding annulus vertices are projected into this plane. The same is done for the desired annulus ring position. In the subsequent step, a local Cartesian grid is created for each plane. Finally, the local coordinate systems of the valve geometry is transformed into the desired annulus position. All cells inside the outer shape of the valve are scaled with a factor based on the distance between the center point and the annulus vertices. With this process, the geometry of the valve is slightly changed. However, the aim of modeling the movement of the MV is achieved with only a small deviation.

To correctly reproduce the MV movement throughout the cardiac cycle from a temporal point of view, the cycle length of one heart beat is adapted in accordance with the simulation setup.

6.3 Results

For modeling the valves based on the porous zone approach, the aortic and mitral valve geometry are obtained from the mechanical solver framework. For the initial time step, their shape and position is shown in figure 6.1 with the original spacing between them.

In figure 6.2 the resulting geometry for the prosthetic valve, transformed into the individualized LV geometry is shown. Different configurations in time including the closed and the fully opened state, as well a position during the opening process are depicted.

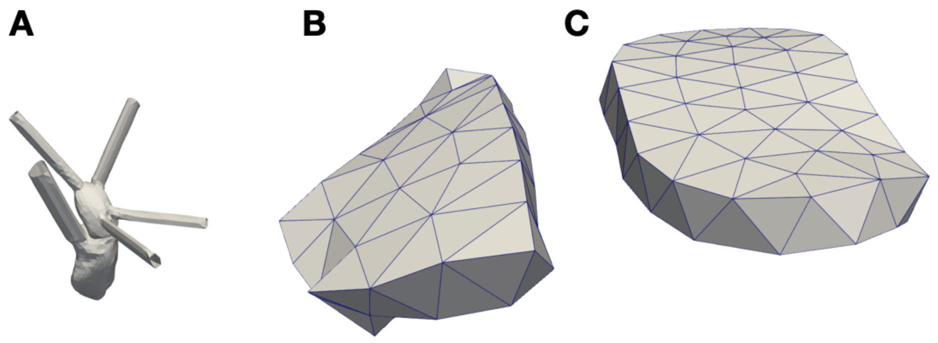


Figure 6.1: Geometry of the porous zones denoting the shape and position for the initial time step $t = 0$ s. The geometry is extracted from the mechanical solver results. The geometry of the left heart is shown for orientation reasons (A). The aortic valve (B), as well as the mitral valve (C) are shown.

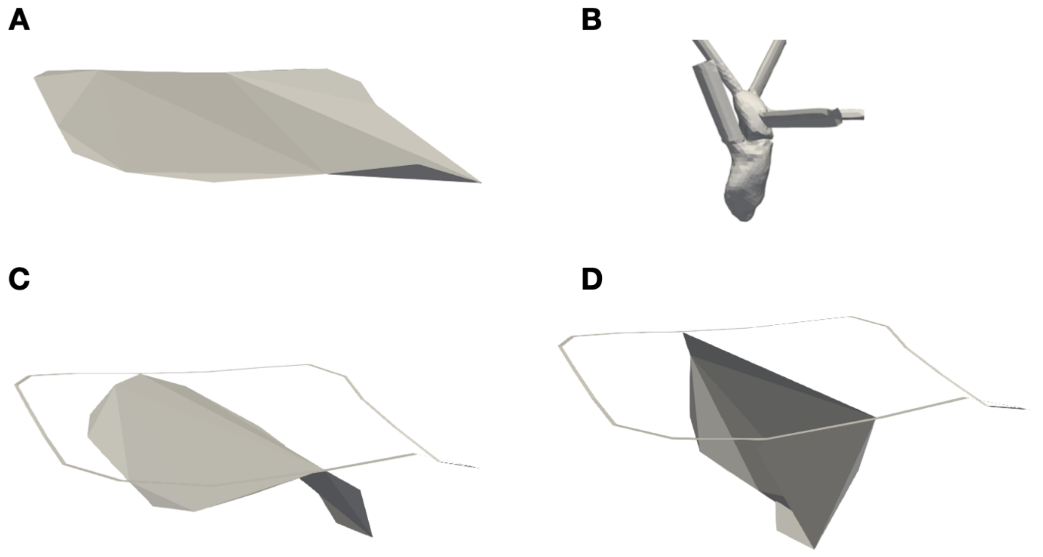


Figure 6.2: Geometry of the prosthetic valve based on the data set of the St. Jude prosthetic valve model [129]. The valve is adapted to fit the annulus position of the individualized left heart geometry, which is shown for orientation reasons in B. The valve is shown in the closed (A) and the fully open (D) state, as well as during the opening process (C).

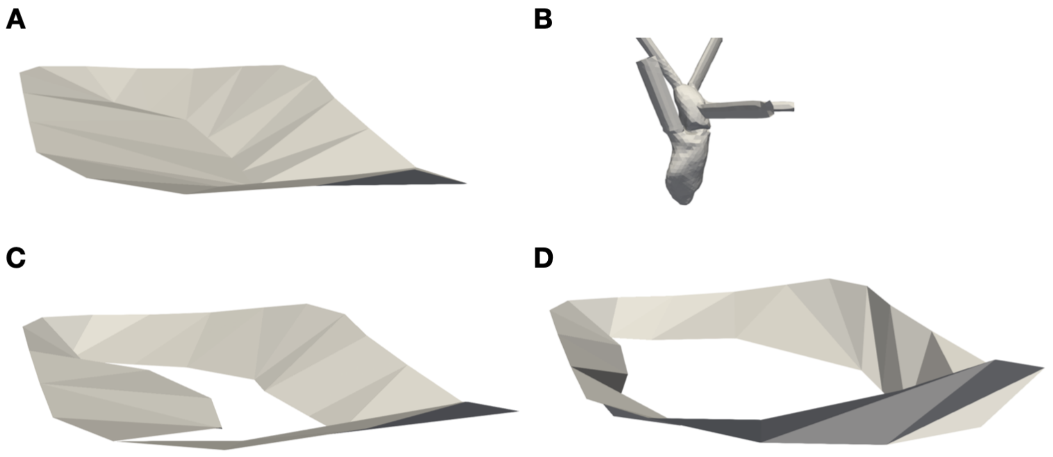


Figure 6.3: Geometry of the Linköping valve based on the measured animal data set [128]. Adapted to fit the annulus position of the individualized left heart geometry shown for orientation reasons in B. The valve is shown in the closed (A) and the fully open (D) state, as well as during the opening process (C).

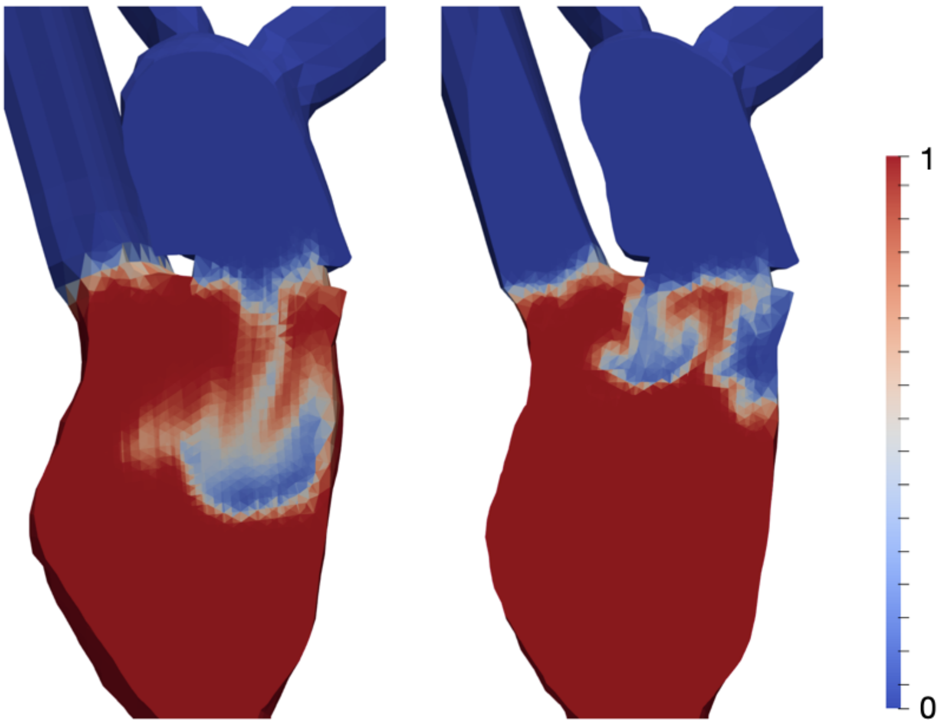


Figure 6.4: Scalar field ψ visualizing the flow patterns in the LV. Comparison of the Linköping valve (left hand side) and the prosthetic valve (right). Figure from [127].

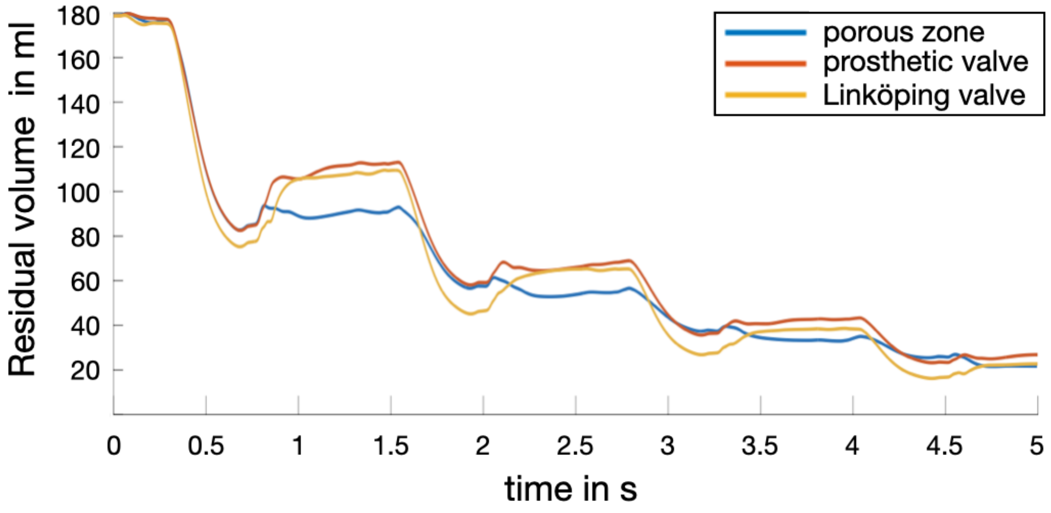


Figure 6.5: Time course of the residual volume V_{res} in ml for four full heart cycles. Comparison between the porous zone implementation, the prosthetic valve and the Linköping valve. Figure adapted from [127].

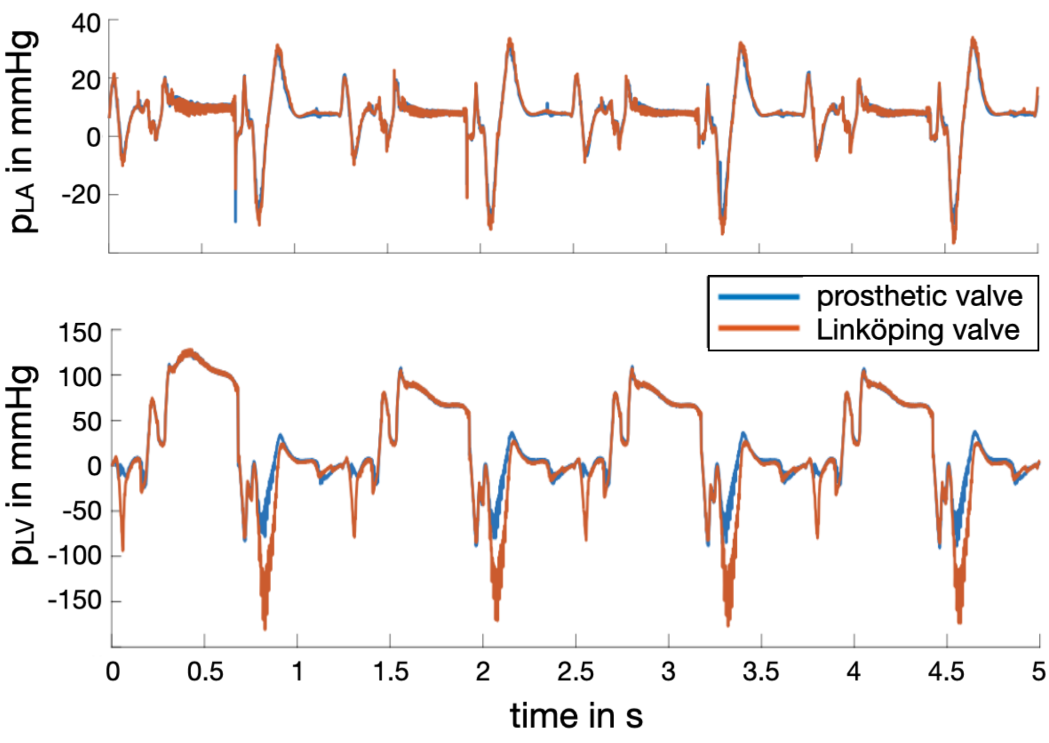


Figure 6.6: Time course of the pressure in the left atrium (LA) (top row) and the LV (bottom row) in mmHg. Comparison between the prosthetic valve and the Linköping valve. Figure adapted from [127].

The same holds true for the geometry based on the measurement data taken at the Linköping university. In figure 6.3 the resulting valve shape, transformed to fit the individualized LV geometry is shown. Different configurations in time including the closed and the fully opened state, as well a position during the opening process are shown.

To assess the resulting flow patterns, a passive scalar transport equation is implemented as will be introduced in section 14.2.7.2. The scalar ψ is considered as a measure of local concentration of blood that was initially located in the LV when the simulation started. Therefore, it is initialized with a value of $\psi(x,t_0) = 1$ for all cells located in the LV, such that the local washout fraction can be visualized.

Comparing the spatially resolved scalar field ψ in a long axis view of the left heart which is cut in two halves, the differences between the Linköping valve (left hand side) and the prosthetic valve (right) can clearly be seen in figure 6.4. Because in the Linköping data set blood from the left atrium (LA) enters in a central position and with a higher velocity magnitude, it penetrates deeper towards the MV. In comparison, in the prosthetic valve, atrial blood enters besides the central partition wall that to which the two artificial flaps are attached and subsequently spreads laterally.

The time course of the scalar field ψ is shown in figure 6.5. Comparing the two approaches reveals that in the Linköping data set higher variations occur. While at the beginning, the decrease is higher, in the plateau phase a similar value compared to the prosthetic valve is reached. This can be explained by the different inflow patterns visualized in figure 6.4. Compared to the porous zone approach, the residual volume resulting after four heart cycles is the same as in the Linköping valve. However, in the first three cycles, differences of up to 20 ml occur. These differences can again be explained by the different geometric implementation of the valves: As the porous zone approach averages the valve characteristics over the whole valve diameter, the significantly higher effective orifice area leads to a jet that penetrates deeper towards the apex. This has the consequence that in the first cycles more blood that initially started in the LV is ejected. Due to the periodic contraction of the endocardial surface walls and the induced mixing of the blood, after a few cycles the deviations are gone and an end state with low deviations ($\Delta V_{\text{res}} < 1$ ml) is reached.

The pressure curves for the LA and the LV are plotted over time for four full heart cycles in figure 6.6. A comparison between the two approaches reveals a high overall correspondence. In the LA (upper part of figure 6.6), the maximum deviation is approximately 11 mmHg in the first cycle, in the following cycles $\Delta p_{\text{LA,max}} = 5$ mmHg. In the LV a significant pressure deviation of approximately 100 mmHg occurs during the opening of the valve. This deviation is due to a variation in the timing: While the opening of the MV in the Linköping data set is slow at the beginning and then accelerates, the prosthetic valve has a linear opening process. Due to the slow opening, the opened orifice area is significantly smaller at the beginning of the opening process. To achieve the same volume flow, a higher negative pressure is necessary.

6.4 Discussion

The four valve models introduced in this chapter hold different characteristics as well as prerequisites and thus are convenient to fulfill different application purposes. While the most simple model, the porous zone approach only relies on a mean value for the whole valve area and does not consider the spatial geometry of the leaflets, it still reproduces the characteristic flow patterns. Due to the direct consideration as an external force in the NSE, it is the method of choice for the body fitted implementation. In the IBM solver framework, the spatially resolved leaflet geometries of the two alternative 3D implementations provide a higher level of detail. As the Linköping and the prosthetic valve are both representations of the MV, for the aortic valve representation still the planar valve model introduced by Daub et al. is in use [38]. As the volume that is ejected from the LV through the aortic valve flows in the prolonged aortic trunk, the spatially resolved flow patterns in this artery are of secondary importance, anyway.

Comparing the Linköping and the prosthetic valve, a direct comparison is impeded by the different geometric conditions prescribed by each of the models. During the opening procedure, the nonlinear behavior of the Linköping valve introduces pressure differences that have to be considered. This is due to a boundary condition of the solver: As the movement of the endocardial surfaces is prescribed, the volume flow through the valve is fixed. Thus, a difference in the valve diameter directly results in a pressure difference. However, when comparing this to the real heart under physiological conditions, this would be the same. Blood is sucked into the LV due to the relaxation of the endocardial wall and therefore an increase of the chamber volume results. This movement triggers the blood flow.

Considering the the long-term quantities of the residual volume which are of high relevance in terms of stroke risk assessment, the compared valve models show a high accordance in the time course of the residual volume. While after a small number of cycles, the deviations are gone, this implies that the choice of the valve geometry does not influence the long-time residual volume. Thus concerning this quantity, all valve models provide sufficient simulation results. However, especially when analyzing and comparing the models in the first cycles, this deviation due to the different model realizations has to be taken into account.

A comparison to literature values reveals a good accordance for the pressure conditions in both chambers [131, 132]. Small deviations of up to $\Delta p = -2.2\,\text{mmHg}$ occurred, but in the scope of the time course do not seem to have a big impact.

Chapter 7

Trunk length

To compute blood flow dynamics in the heart, in most research groups, the inlet and outlet pressures are described at the orifices. In case of the left heart side, this is at the pulmonary veins and the aorta. However, the veins and arteries are often not segmented from the medical images, as most tools do not offer an automatic segmentation at least for the pulmonary veins. In addition, during the acquisition of whole heart stacks, they normally are not included in the measured data set at all. For this reason and to allow for a straight tube vessel approximation that minimizes the influence of dynamic processes, the inlet and outlet trunks are commonly prolonged by straight tubes. The aim is that at the end of the rigid tubes and thus where the blood enters the left atrium (LA), the flow profile is fully developed and the flow patterns are not influenced by dynamic flow profile development processes. At the outer end of these trunks, pressure boundary conditions provided by the circulatory system model are prescribed.

In this study, we systematically investigate the influence of a variation of the trunk length on the blood flow in the heart chambers. The aim is to answer the question, how a variation of trunk length alters LA inflow conditions. Additionally, the effects on the global fluid dynamics results are assessed. The content of this chapter was presented at the iHeart "Modelling the Cardiac Function" 2022 conference [133].

7.1 Methods

Three simulation scenarios are implemented: For the reference case, all trunks are initialized with a reference length l_0, which is approximately 95 mm and corresponds to the 6-fold aortic diameter. Before this study, the length l_0 was used as the standard trunk length throughout the simulations. To systematically evaluate the influence of shortened trunks, the initial length was reduced by 25 % and 50 %, respectively. Beneath the standard length l_0, the two shortened versions with $0.5 \cdot l_0$ and $0.75 \cdot l_0$ result as visualized in figure 7.1.

To assess the resulting flow patterns, a passive scalar transport equation is implemented as will be introduced in section 14.2.7.2. The scalar ψ is considered as a measure of local

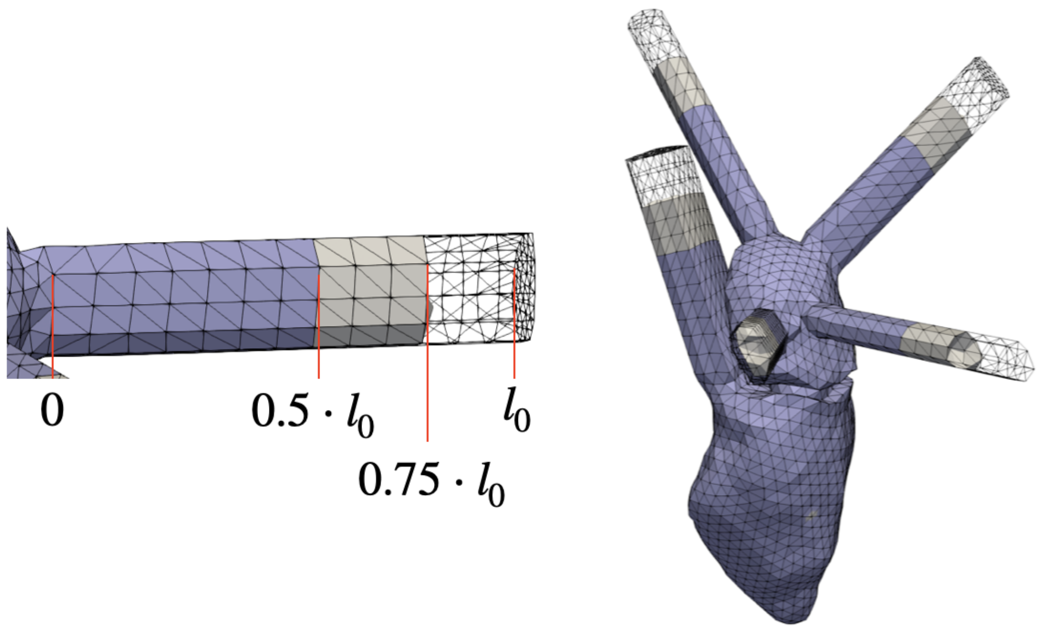


Figure 7.1: Resulting geometry of the trunks with varied length. Left: Exemplary pulmonary vein comparatively visualizing the the different trunk length $0.5 \cdot l_0$ (blue color), $0.75 \cdot l_0$ (gray color) and l_0 (black wireframe visualization). Right: Overall geometry of the left heart side as a superposition of the three trunk length variations.

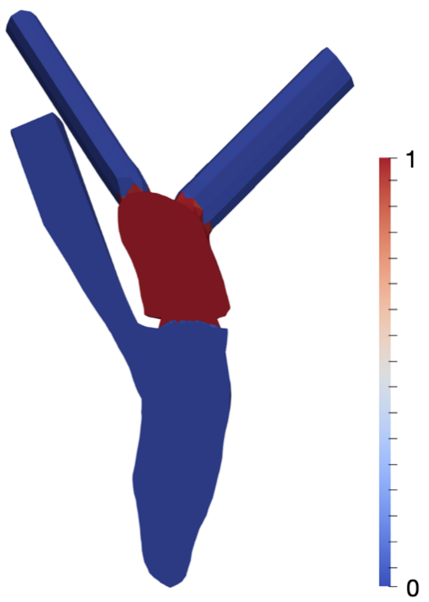


Figure 7.2: Initialization of the scalar field ψ. All cells in the LA are set to $\psi = 1$.

concentration of blood. It is initialized with a value of $\psi(x,t_0) = 1$ for all cells located in the LA, such that the local washout fraction can be visualized.

In comparison to other applications of the scalar transport equation throughout this thesis, in this case all atrial cells are set to a value of $\psi = 1$ at the initialization. Thus, not the ventricular cells are evaluated but with this choice, flow patterns in the LA directly at the junction of the pulmonary veins can be analyzed. The initial values of the scalar field ψ are visualized in figure 7.2.

7.2 Results

The resulting scalar field ψ during the atrial filling phase is shown in figure 7.3. In this relaxed state, blood from the pulmonary veins passively flows into the LA and mixes with the initial blood volume. While basically, the flow patterns visually are similar, deviations in the mixing procedure can be observed. The differences between the shortest and the standard trunk length are the smallest, only resulting in slight difference in the central area of the LA. In the scalar field of the $0.75 \cdot l_0$ trunk, an increased inflow through the right visualized pulmonary vein can be observed in the cut plane. A small circular vortex develops in the central region. In the wall region, as well as close to the mitral valve (MV) no changes occur during the first cycle filling phase.

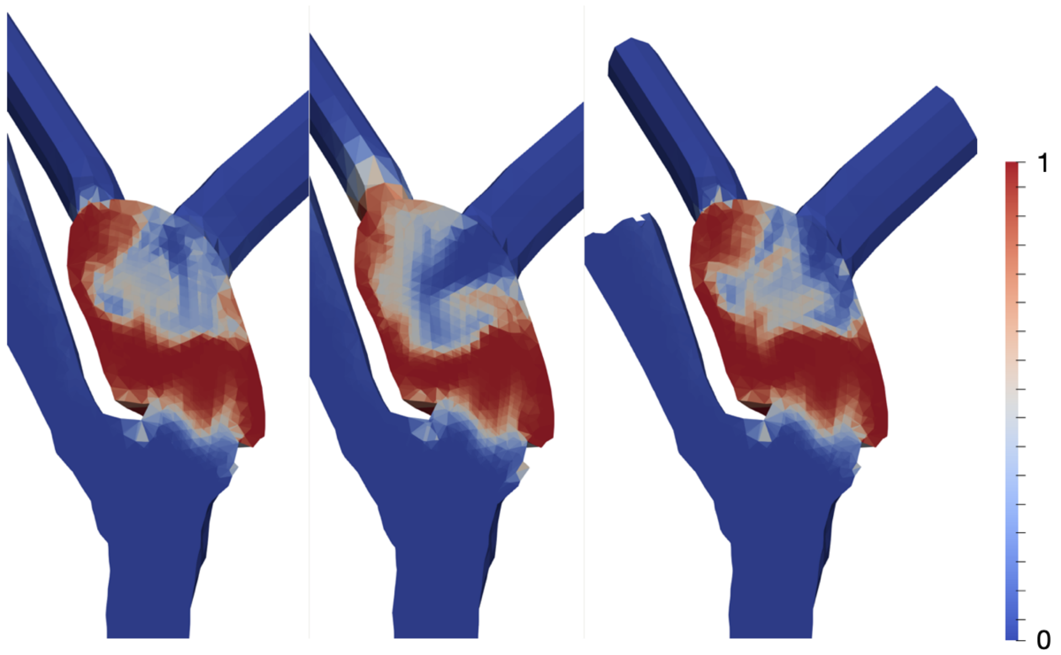


Figure 7.3: Scalar field ψ color-coded in a cut of the long axis view of the geometry to visualize the flow patterns in the LA. Time point $t = 0.7$ s during the atrial relaxation phase. Left: Reference geometry l_0, center: $0.75 \cdot l_0$ and right: $0.5 \cdot l_0$.

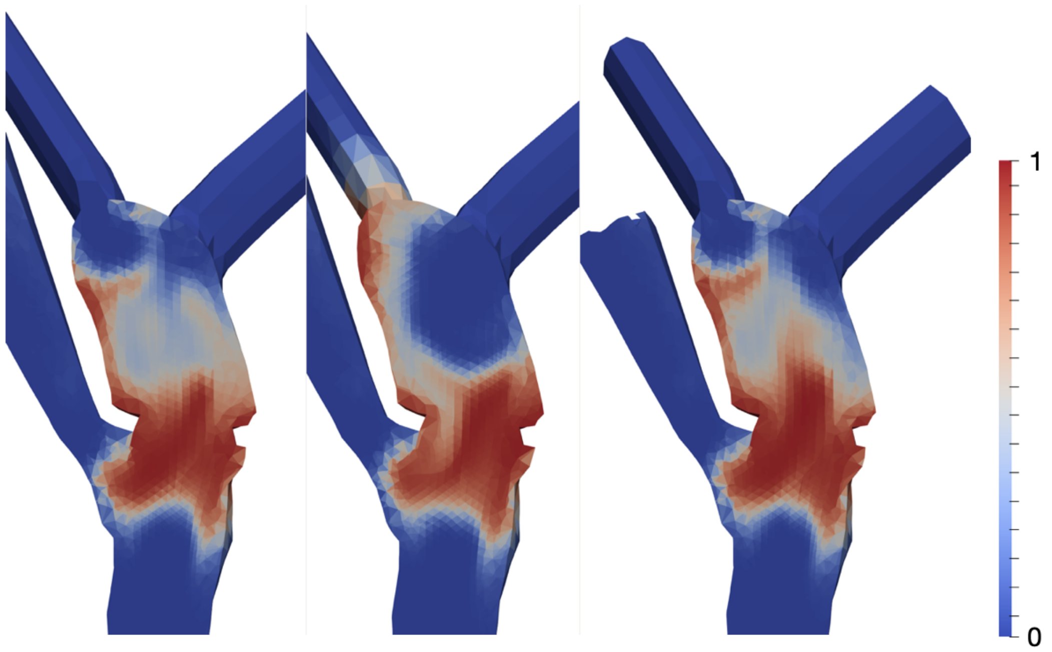


Figure 7.4: Scalar field ψ color-coded in a cut of the long axis view of the geometry to visualize the flow patterns in the LA. Time point $t = 0.83$ s during contraction of the atrial chamber. Left: Reference geometry l_0, center: $0.75 \cdot l_0$ and right: $0.5 \cdot l_0$.

In figure 7.4 the same setup is shown for a later point in time in the cardiac cycle. At $t = 0.83$, during the atrial contraction phase, blood is ejected trough the MV in the ventricular chamber. The transition of the blood volume that initially started in the LA into the left ventricle (LV) can be observed. While the overall pattern is again similar, the biggest amount of initial blood volume has already left the atrium during the first heart cycle. Quantitatively, the washout fraction of the LA is reduced by up to $1\,\%$. Visually, a moderate variation in the blood flow patterns can be observed, especially in the center of the LA. While in the shortest trunk geometry ($0.5 \cdot l_0$, right hand side), the amount of remaining blood volume close to the MV is the highest, in the center geometry, the inflowing blood volume has distributed directly in the visualized cut plane and forms a big area with blood that entered in the current cycle. A small fraction of blood that entered in the current cycle is directly ejected through the MV.

In addition, the time course of the LV pressure is plotted in figure 7.5 to evaluate a possible influence on the whole left heart simulation. The time course shows an concordant course for all three trunk length scenarios with deviations smaller than ± 2 mmHg for nearly the whole cycle. The only exception is the begin of the ventricular contraction at approximately $t = 0.7$ s. For the trunk shortened by $25\,\%$ of the standard length l_0, a drop in the averaged chamber pressure of up to ± 2 mmHg in the LA and ± 6 mmHg in the LV can be observed.

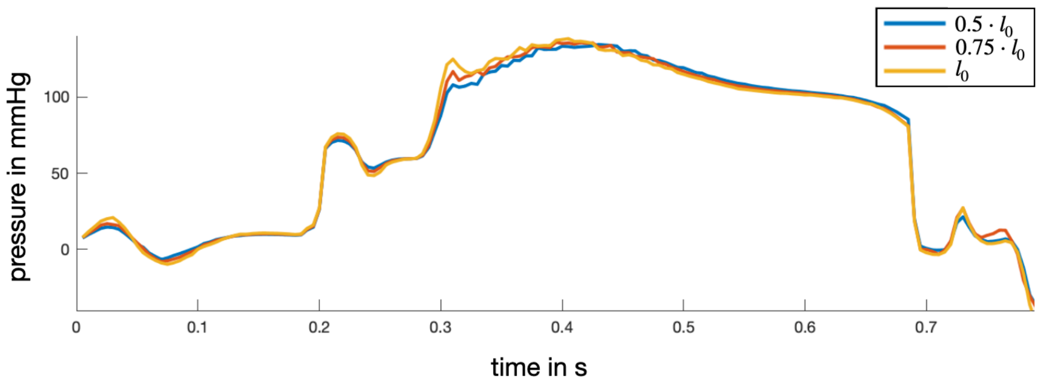


Figure 7.5: Time course of the LV pressure in mmHg for a full heart cycle. The three different trunk length scenarios are color-coded.

7.3 Discussion

Simulating the flow patterns in the LA for different lengths of the prolonged inflow and outflow trunks revealed only slight variations in the visual and quantitative evaluation of the flow patterns. While the washout fraction of the LA is reduced by a factor of up to 1% ($0.75 \cdot l_0$ compared to l_0), it can be argued, whether this low difference is of high relevance. However, as seen in the visual comparison, the flow patterns changed - even if this is not adequately reflected in the time course of the values. As the quantitative value is computed as the sum of ψ over all single cells in the LA, local deviations are neglected.

The differences in the LV pressure simulations can be explained by the shorter distance between the pressure inlet and the atrial chamber, as the pressures are prescribed at the end of each trunk. A comparison to the Hagen–Poiseuille equation, which links the pressure drop in a straight tube to the length of the tube, supports this hypothesis. Therefore, as the difference between inlet orifice pressure and the mean LA pressure is determined by the circulatory system model beforehand, the length of the trunks is interrelated to the choice of the circulatory system model parameters.

Coming back to the questions raised in the introduction, a variation in trunk length has only limited influence on the atrial flow patterns, as long as a minimum length is maintained. Flow patterns and pressure curves are changed slightly. Thus, for an optimal simulation result, the trunk length has to be chosen in accordance with the circulatory system parameters.

All in all, the results highlight the importance of accurate boundary conditions as well as an accurate interaction of circulatory system and fluid dynamics model. It also indicates that the optimal length of the trunk has to be computed individually for each geometry and most notably in accordance with the circulatory system parameters to obtain accurate flow results.

For further investigation, it has to be considered to which extent the curvature of a bend trunk influences the inflow conditions, as of course the veins and arteries in the human body do not form a straight tube geometry. In addition, it would be valuable to determine, to which extent flexible trunks would alter the flow patterns in the LA. This is especially of interest, as physiologically the aorta fulfills a Windkessel function to smooth the pressure edges of the pulsating flow.

PART III

SOLVER VERIFICATION

Chapter 8

Simplified geometry simulation

To verify the moving immersed boundary method (IBM) implementation and parameterization with respect to different aspects of its functionality, four simple either static or time-dependent test scenarios are set up and evaluated in this chapter. All verification steps are based on simulated data. The main focus of this chapter lies on 2D patch simulations, as results can be interpreted more comprehensively. In the last section the extension to a cubic grid is included to also emerge to the 3D application scenario.

While this introduction presents a short overview over the implemented test cases, it additionally defines the requirements that are proven in each test setup. In each discussion part, these requirements are picked up again.

In the first section, the setup of a flat plate scenario is introduced. The **static plate** is exposed to a constant flow. Applying the Blasius' solution, the aim is to verify the developing boundary layer versus the analytical solution. In the next section, the setup is time dependent: while the fluid is in rest at the initiation, the **plate** accelerates and then **moves** through the channel with a constant velocity. The symmetric displacement of the fluid and the resulting laminar boundary layer is verified. Thirdly, the inflation process of the heart is mimicked by a **cylinder** that **inflates** and deflates periodically. While the enclosed volume changes, the inflow and outflow of the fluid can be modeled. Therefore, it can be verified that the amount of blood that is prescribed analytically by the shape variation of the cylinder (prescribed by an analytical function) leaves and enters the channel as calculated (deduced from the change in the geometry). In addition, the scenario serves to prove that the enclosed blood does not leave the geometry throughout the wall. This is done on the one hand by demonstrating a constant flow through the channel, on the other hand the fluid volume leaving the patch through the input output boundary Ω_{io} outside the channel has to be inverse to it. The third requirement that can be investigated using this test setup relates to the way the wall movement is prescribed: on the one hand the distance field Λ can be computed by an analytical function, on the other hand it can be computed from external geometries. In this context, this section additionally aims to verify the conformity of both procedures. Finally, the 2D cylindrical scenario is extended to a 3D **inflating sphere** to also show the functionality on a higher dimensional grid. In this section, the aim is to verify that the 2D velocity profiles can be transferred to the 3D case and that no blood leaves the channel through the wall.

8.1 Static plate

8.1.1 Simulation setup

In 1904, Ludwig Prandtl introduced the fluid dynamic boundary layer theory as a specialized field of fluid mechanics. Therefore, he separated the flow in the proximity of a body in two parts and distinguished between the flow inside and outside of the boundary layer. While in the outer area friction is neglectable, inside the narrow region close to the wall friction takes a central role. The conditions inside the boundary layer allow for a simplification of the Navier Stokes equation (NSE) such that an analytical solution for them can be found [134].

This analytical solution is used in this section to verify the resulting boundary layer in an infinite channel setup. Therefore, an infinite plate in an infinite channel is set up. As visualized in figure 8.1 A and B, a thin plate in the center of the computational grid is prescribed by the IBM. Therefore, with a small distance from the left wall, a velocity $u = 0\,\mathrm{ms}^{-1}$ is prescribed for all plate cells. On the left wall itself, a constant flow of $u = u_x = u_\infty = 1\,\mathrm{ms}^{-1}$ is prescribed in positive x-direction (inside the channel). Upper and lower wall are prescribed as symmetry planes and thus do not influence the flow. The right wall is assigned a zero gradient velocity condition and thus serves as an outlet patch where the flow can leave the grid. To achieve a high resolution of grid cells, a $0.02\,\mathrm{m} \times 0.02\,\mathrm{m}$ grid with 400 cells in each direction is used.

The analytical solution for the NSE in the context of an infinite plate in an infinite channel is computed based on the Blasius solution. The thickness δ of the resulting boundary layer depends on the distance to the beginning of the plate: $\delta = \delta(x)$. It can be computed based on the following equation:

$$\delta(x) = \frac{5}{\sqrt{Re_x}} \cdot x \,, \tag{8.1}$$

while the Reynolds number computed as

$$Re_x = \frac{u_\infty \cdot x}{\nu} \tag{8.2}$$

and $u_\delta = u(y = \delta) := 0.99 \cdot u_\infty$, as well as the viscosity ν.

8.1.2 Simulation results

The resulting static indicator field Φ is shown in figure 8.1A, as well as in the close-up (B). The first two cells are not included to not force a zero velocity $u_x = 0\,\mathrm{ms}^{-1}$ in the plate while the boundary is set to the constant velocity u_∞. The velocity field u (figure 8.1C) reveals the expected laminar flow profile with a low velocity close to the wall and an unchanged flow velocity u_∞ outside the boundary layer. On the right hand side, a close-up of the flow profile is shown. The color bar is chosen such that only two discrete values for velocities above or below the limit of $u = 0.99 \cdot u_\infty$ result. While in black color the position of the static plate is denoted, the green line represents the analytically computed Blasius solution following equation 8.1. The thickness δ of the boundary layer matches the analytical solution with deviations below the grid size.

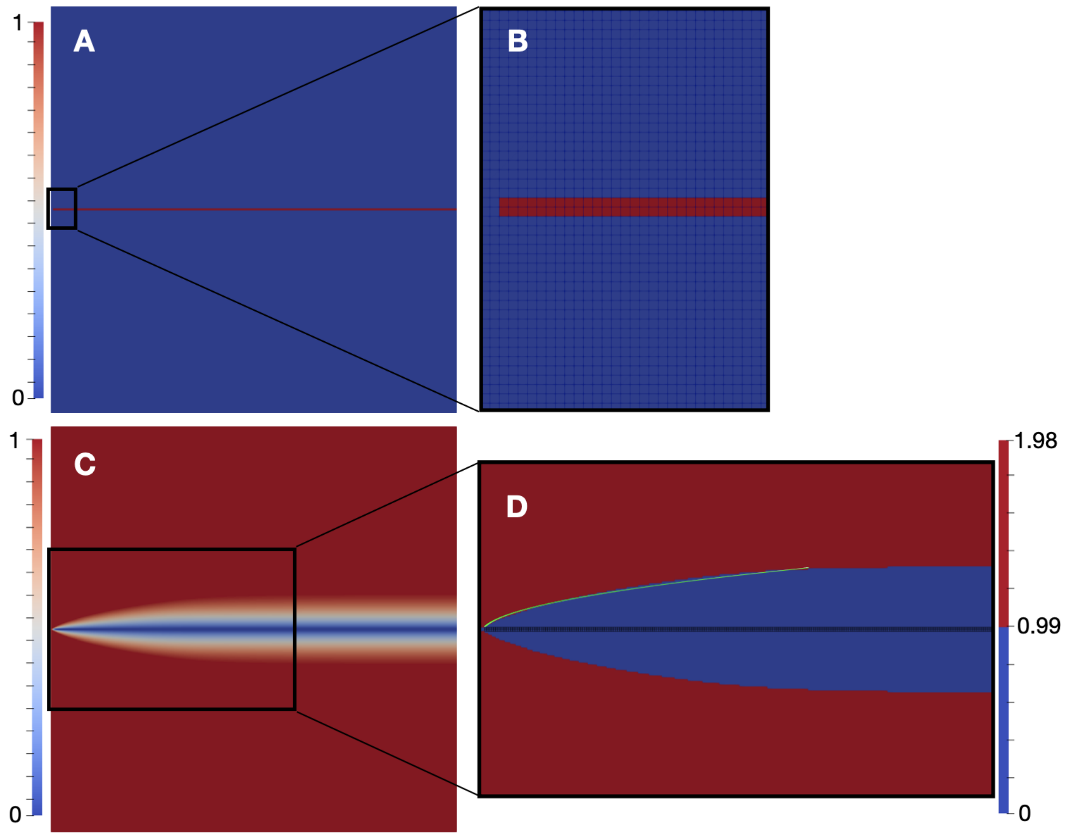


Figure 8.1: Top row: indicator field Φ that denotes the position of the infinite flat plate. Bottom row: resulting velocity field $\mathbf{u}$ in ms^{-1}. The right hand side figure delivers a close-up (D) of the area framed in (C) comprising the indicator Φ in black and the thickness of the boundary layer computed by the analytical Blasius solution in green.

To further investigate the boundary layer, the velocity u_x in flow direction is plotted over a perpendicular analysis line. While the x-coordinate is kept constant at $x = 0\,\mathrm{m}$, the velocity is plotted over the y-coordinate in figure 8.2. As expected, the velocity equals the prescribed flow velocity u_∞ throughout nearly the whole the grid. In the center of the grid ($y = 0\,\mathrm{m}$), where the plate is cut, a steep decay to the prescribed zero-velocity occurs. As this decay at the edge (in the proximity of the edge) of the plate is the region of interest, this area is enlarged in the right part of the figure. While the five black vertical lines mark the edges of the four cells that model the static plate, the velocity equals $u_x = 0\,\mathrm{ms}^{-1}$ at each center of the four grid cells. Due to the cell-to-point interpolation and the linear connection between the interpolated points, a velocity $u_x \neq 0$ occurs in the outer half of the outer cells. The comparison to the analytical Blasius solution depicted by the green line shows a good accordance.

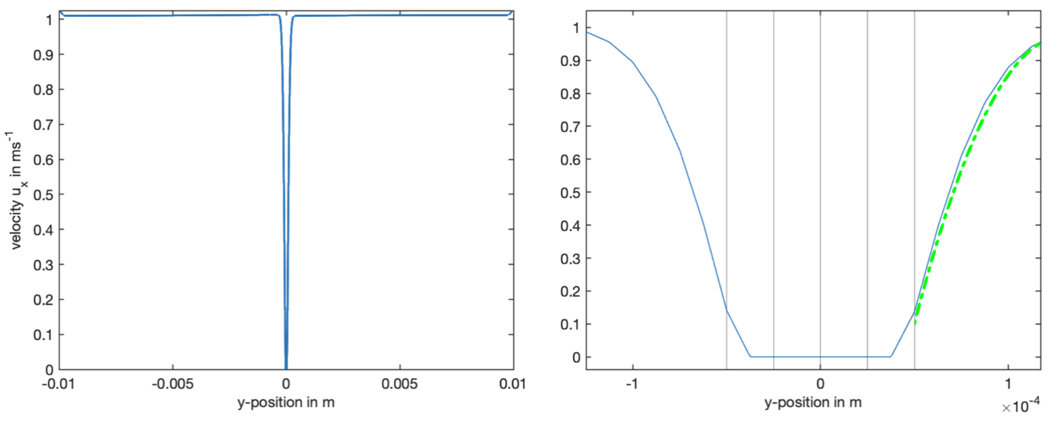


Figure 8.2: Velocity u_x in flow direction in ms^{-1} plotted over a line in y-direction ($x = 0$ m). The static plate is cantered at the position $y = 0$ m. On the right hand side, a zoomed view on the relevant area is included. The black vertical lines denote the position of the edges of the four cells that model the static plate. The green curve shows the analytic Blasius solution.

8.1.3 Interpretation and discussion

In the first simplified simulation setup, a static plate in an infinite channel which is flowed with a constant velocity was implemented. A comparison to the Blasius solution verified the flow profile to be in good accordance with the analytical solution for the boundary layer.

In the evaluation, it seemed that a velocity greater than zero occurred inside the solid plate. However, when assessing the velocities, it turned out to be an artifact introduced by the cell to point interpolation. Nevertheless, this artifact can be further reduced by an increase of the mesh resolution, as it will always be half the size of the edge length.

Apart from that, following the Blasius solution, the steepest gradient directly has to occur at the edge of the plate. However, when considering the zero-velocity on the edge of the plate, this is fulfilled in the current simulations.

8.2 Moving plate

8.2.1 Simulation setup

The second verification setup is similar to the first one. Again an infinite plate and an infinite channel are used. However, this time the fluid is in rest and the plate moves with a constant velocity (after acceleration ended). Therefore, this is the first dynamic verification setup and the moving IBM approach can be verified. The plate is modeled as a completely solid medium. While the plate thickness is 0.1 m, the prescribed plate velocity in the channel direction is $u_x = 0.005\,\mathrm{ms}^{-1}$. With the start of the simulation, the plate accelerates from rest until the velocity u_x is reached. Afterwards the constant velocity is applied.

8.2.2 Simulation results

The prescribed indicator field Φ is shown on the left hand side of figure 8.3. For three specific points in its position is shown in shades of red with increasing opacity. On the right hand side of figure 8.3, the resulting velocity field u_2 is plotted for the specific point in time t_2. Despite the angularly shaped plate, a resulting parabola-like flow profile can be observed. Even if the conditions compared to the first simulation setup are switched (now the wall moves in a resting fluid), a laminar boundary layer emerges (compare figure 8.1D).

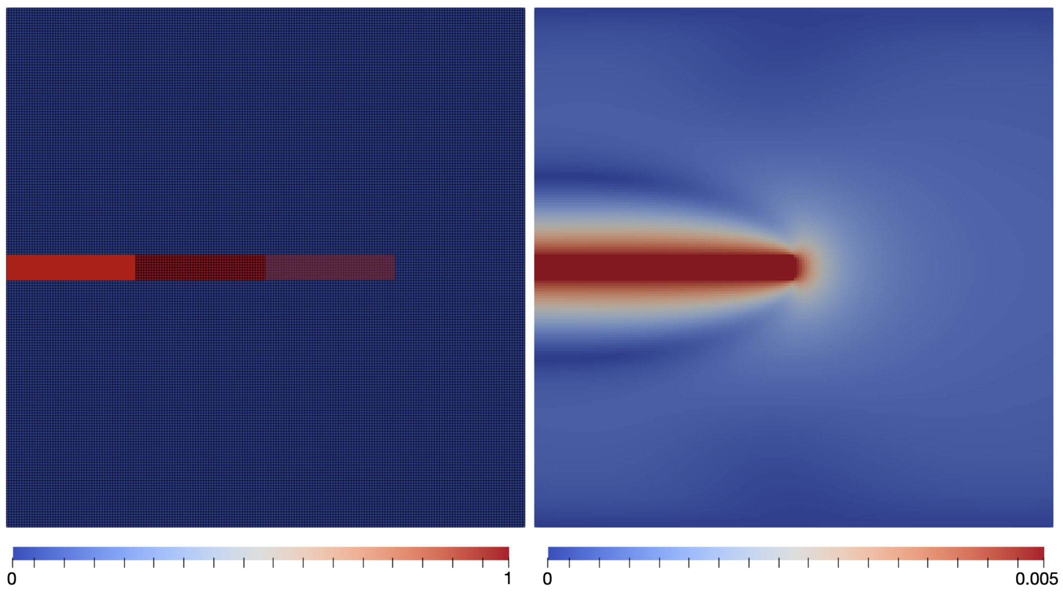


Figure 8.3: Left: Indicator function Φ denotes the position of the plate at three specific steps in time $t_1 < t_2 < t_3$. A movement in x-direction from the left edge of the grid towards the right hand side is prescribed. Right: Resulting velocity field $\mathbf{u}$ in m/s for time step t_2

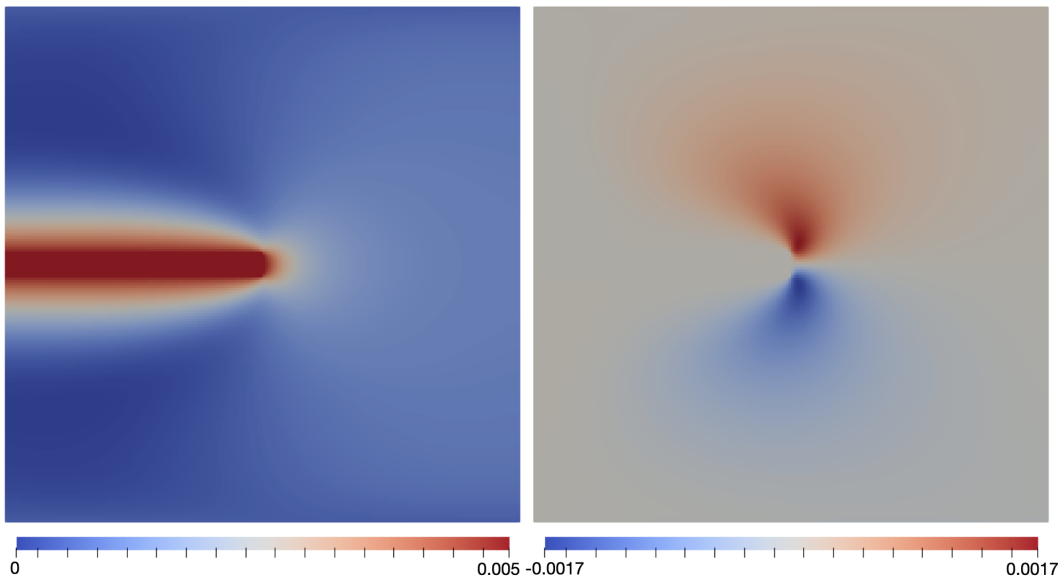


Figure 8.4: Velocity field $\mathbf{u}$ in $\mathrm{m\,s^{-1}}$ for the time step t_2 split up in x-component (left) and y-component (right) of the velocity vector.

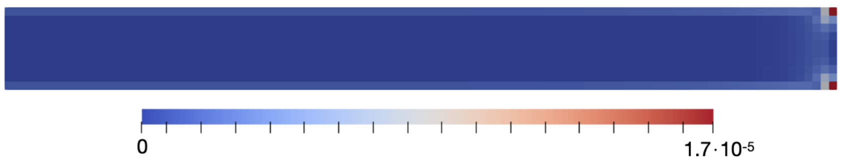


Figure 8.5: Difference of the resulting velocity field and the prescribed wall velocity $\Delta\mathbf{u} = \mathbf{U}_w - \mathbf{u}$ in $\mathrm{m\,s^{-1}}$ for the close-up view of the plate.

In figure 8.4 the components of the velocity vector are split up in x- and y-components. The x-component u_x of the velocity contributes the largest share, while u_y is only relevant at the front of the plate where the fluid is displaced. The results are fully symmetric.

In figure 8.5 the difference of the prescribed and the resulting velocity is shown. The maximal difference arises at the front end of the plate. However, in relation to the prescribed wall velocity the difference is smaller than $0.35\,\%$.

8.2.3 Interpretation and discussion

In the second simplified simulation setup, a solid plate moves in an infinite channel through a resting fluid. First of all, the simulation proved that the prescription of a movement by applying the wall velocity $U_{\mathrm{w},i}$ in the IBM setup is enabled. In addition, the prescribed velocity was achieved in all wall cells, exactly as prescribed. In this fully symmetric simulation setup, also the resulting flow field proved to show a symmetric behavior.

8.3 Inflating cylinder

8.3.1 Simulation setup

To further verify the IBM implementation and parameterization in a setup that allows for the enclosed volume to change and to accelerate and decelerate blood in a periodic manner, an inflating cylinder scenario is set up. This again covers a really simple scenario on a 2D patch (2 m x 2 m with the x and y range between -1 m < x,y < 1 m). A circular cylinder with radius $r = r(t)$ following equation 8.3 is located in the upper part of the quadratic patch around the center y_c. This cylinder is attached to a rectangular channel which connects the cylinder with the boundary of the simulation domain and thus serves as an in-/outlet channel that allows for blood being sucked in the cylinder (inflation) or ejected out of it (deflation). The geometry of the setup is illustrated in figure 8.6.

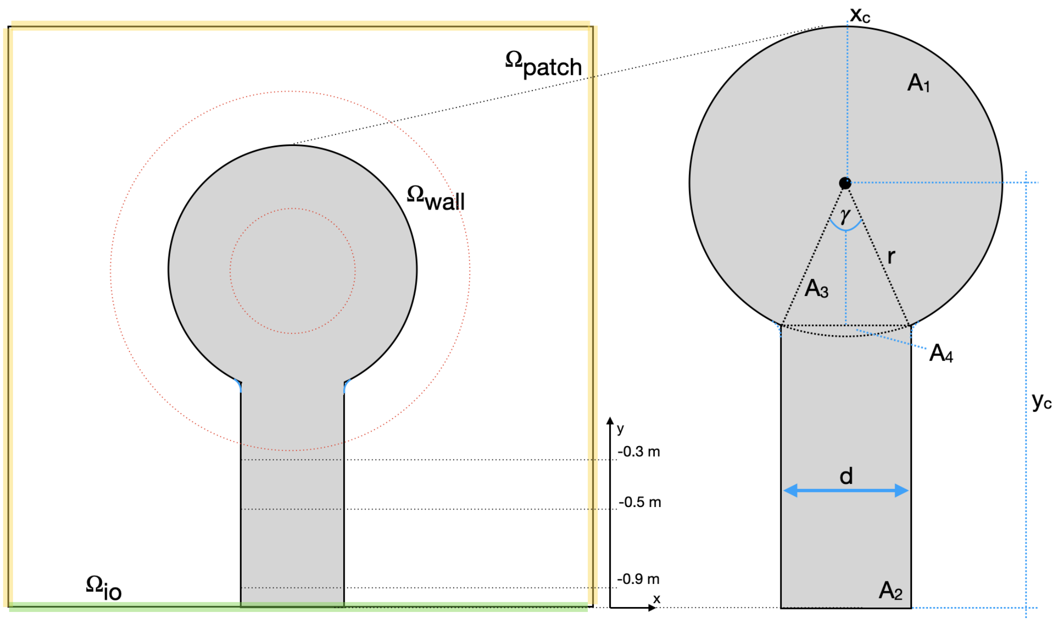


Figure 8.6: Inflating cylinder schematic. Left: 2D patch from top view, cylinder and attached channel. Red dashed lines denote minimum and maximum cylinder size. Right: Annotation of the geometric compartments and variables.

While on the three walls of the patch Ω_{patch} (left, right and top, marked in light yellow) a zero velocity boundary condition and a zero gradient pressure condition are applied, for the inlet respectively outlet boundary Ω_{io} (whole lower boundary, marked in light green) a zero gradient velocity is assumed and a pressure of 0 Pa is prescribed.

As the ground truth reference value, the geometric variation of the fluid surface over time can be computed analytically. The resulting surface A is calculated on the underlying simple geometries and in dependency of time. While A_1 is a circular surface and A_2 a rectangle, the overlapping area A_4 has to be subtracted (compare figure 8.6). That can be calculated based on the triangle A_3 and the circle segment $A_3 + A_4$ based on the following equations:

$$r = r(t) = r_0 + dr \cdot sin(\omega t) \tag{8.3}$$

$$\gamma = 2 \cdot arcsin(\frac{d}{2 \cdot r}) \tag{8.4}$$

$$A_1 = \pi \cdot r^2 \tag{8.5}$$

$$A_2 = (y_c - r) \cdot d \tag{8.6}$$

$$A_3 = \frac{d}{2} \cdot r \cdot cos(\frac{\gamma}{2}) \tag{8.7}$$

$$A_3 + A_4 = \pi \cdot r^2 \cdot \frac{\gamma}{2 \cdot \pi} \tag{8.8}$$

$$A = A_1 + A_2 - A_4 \tag{8.9}$$

The variables denote the initial radius $r_0 = 0.3$ m, radius change $dr = 0.15$ m, location of the center point $y_c = 1.2$ m above the lower channel edge, the channel width $d = 0.1$ m and the resulting angle γ. The surface change is given by the derivative $\frac{dA}{dt}$, the resulting surface flow into the channel can be obtained by multiplying with the patch height $h = 0.01$ m as

$$q = \frac{dA}{dt} \cdot h \; . \tag{8.10}$$

This flow value computed from the variation of the geometrical shape in the following serves as the **ground truth** value.

For the computation of the necessary indicator field Φ, a distance field is computed as introduced in chapter 4. In this geometrically simple scenario, the distance field Λ can be prescribed analytically for each cell center $\mathbf{x}_i = (x_i, y_i)$ based on the Euclidean distance (ED) to the circular wall:

$$\Lambda_i = \begin{cases} |ed_i - r(t)| & y_i > y_c \\ |min((x_i - x_c - 0.5d), (ed_i - r(t)))| & \text{else} \end{cases} \tag{8.11}$$

with

$$ed_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{8.12}$$

The wall velocity U_w for each point $\mathbf{x_i}$ in the directions x and y can be computed based on the cylinder radius. Velocities U_w are weighed by the sine and cosine function to account for the circular structure. Therefore the angle γ that each point encloses with the x-axis is computed by equation 8.15. This method to prescribe the moving wall is based on and therefore referred to as an **analytical function** wall prescription.

$$r = r(t) = r_0 + dr \cdot \sin(\omega t) \tag{8.13}$$

$$\frac{\partial r}{\partial t} = dr \cdot \omega \cdot \cos(\omega t) \tag{8.14}$$

$$\gamma = tan^{-1}\{\frac{y_i - y_{c,i}}{x_i - x_{c,i}}\} \tag{8.15}$$

$$U_{w,x} = \frac{\partial r}{\partial t} \cdot \cos(\gamma) \tag{8.16}$$

$$U_{w,y} = \frac{\partial r}{\partial t} \cdot \sin(\gamma) \tag{8.17}$$

Alternatively, as introduced in chapter 4, the location of the wall can be prescribed by external surface files. This way of prescription is additionally used for this simulation setup to enable a comparison between both procedures. A time series of 300 geometries is supplied for one periodic inflation and deflation cycle. It is referred to as the **external file** prescription.

8.3.2 Simulation results

The results for this simulation setup are presented in a triply divided structure: the solution with the analytically prescribed wall motion (equations 8.11 to 8.17) is visualized on the left hand side (A) of the figure. In the center (B), the results obtained by the prescription of the wall movement by external surface files is shown. The image on the right hand side (C) shows the difference between both methods. Most of the figures focus on the prescribed boundary and deliver a close-up of the wall region, as the main differences arise there.

In figure 8.7 the resulting distance field is shown. While the distance increases in the normal direction of the cylinder as expected, the deviations between the two prescription procedures are low. With a maximum value of $|\Delta\Lambda| = 0.005\,\text{m}$, the difference between both methods is lower than half the edge length. The average deviation throughout the wall's center line cells is in an even lower range and decays with increasing distance from the center line.

The resulting indicator field Φ is shown in figure 8.8. While the basic structure is the same, the deviations observed in the difference $\Delta\Lambda$ of the distance fields (see figure 8.7C) result in a deviation of the indicator field of up to 6 % (figure 8.8C). The number is that high due to the sharp decay of the cosine function weighing the distance field (equation 4.3). Comparing $\Delta\Lambda$ and $\Delta\Phi$, the corresponding high value cells can be observed in both figures.

Finally, the resulting velocity field u is shown in figure 8.9. While the basic structure is the same, close to the prescribed wall deviations of up to 5 mm/s occur.

Figure 8.10 shows the velocity field u for the whole simulation grid. The upper row shows the inflation of the cylinder with the fluid entering through the channel and spreading across the two sides of the cylinder. During the deflation process an outflow can be seen in the lower row. The deviations again occur on a small scale of less than 1 mm/s in average. While during the inflation process, deviations mainly occur in the enclosed volume, during

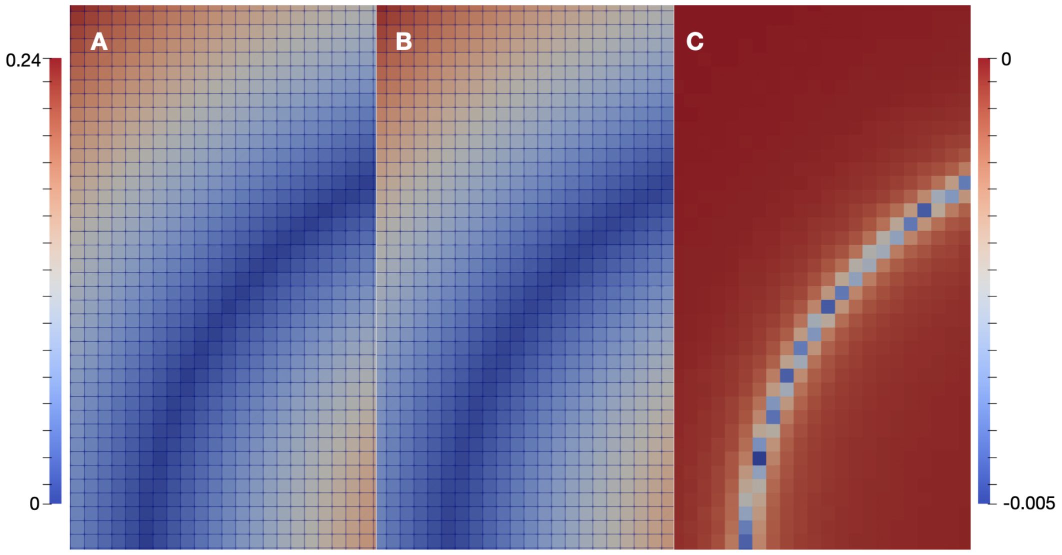


Figure 8.7: Distance field Λ in m. Analytic function (A) and external file (B) prescription as well as their difference $\Delta\Lambda = \Lambda_A - \Lambda_B$.

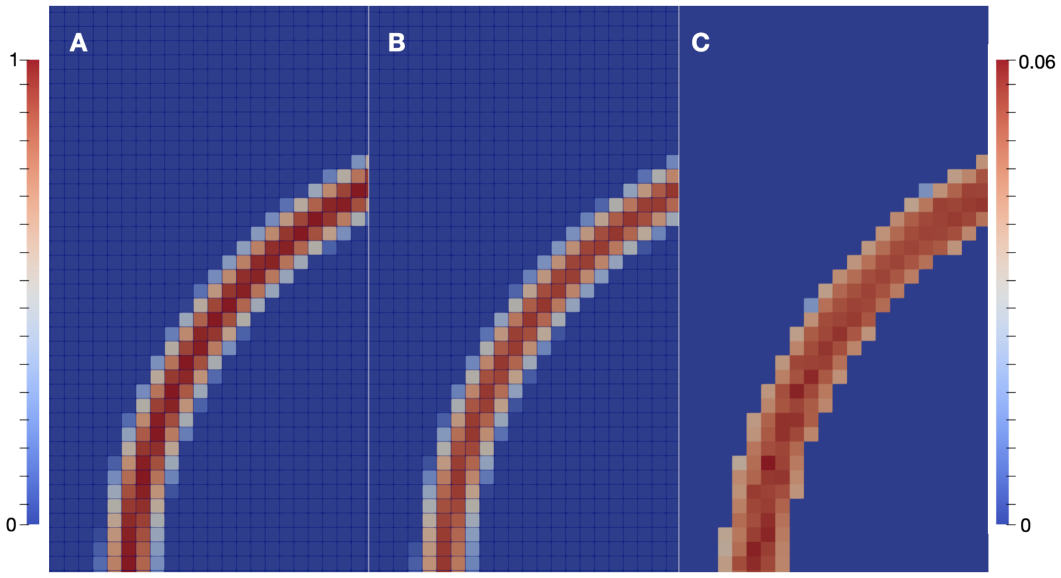


Figure 8.8: Indicator field Φ (dimensionless). Analytic function (A) and external file (B) prescription as well as their difference $\Delta\Phi = \Phi_A - \Phi_B$.

deflation they mainly occur in the channel and the wall area. However, in relation to these velocities, the deviation of below 1 % does not seem to be significant.

The velocity vectors in figure 8.11 show a behavior as expected. While during rapid inflation (A) the highest velocities occur in the channel due to the fluid sucked in, in the

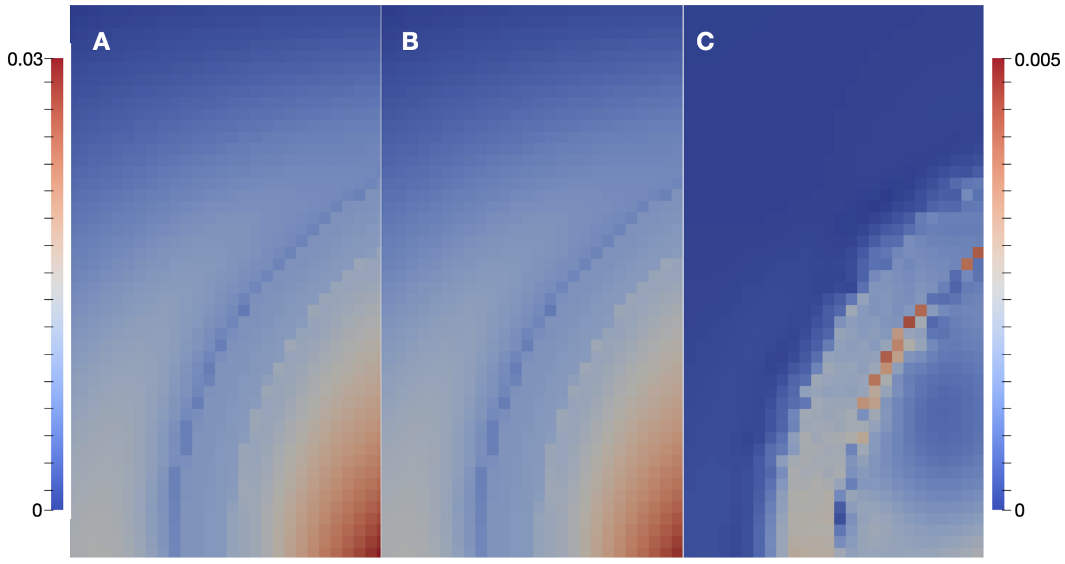


Figure 8.9: Velocity field u in m/s. Analytic function (A) and external file (B) prescription as well as their difference $\Delta u = u_A - u_B$.

outside area close to the cylinder wall the external fluid is pushed away in a circular manner. When inflation retards, maximum velocities arise in the cylinder itself before the velocity inverts due to the transition to the deflation process. Due to the boundary conditions (BC) for Ω_{wall} the external volume leaves the grid at the lower end Ω_{io} during inflation. The same or the exactly inverse behavior can be seen in the other figure parts. Even if the figures may mistake because not every velocity vector is plotted (except for C), the results are perfectly symmetric. This is proven in (C), which shows exactly the same time step as (D) with the only difference that all velocity vectors are depicted.

To additionally evaluate the velocity field over time and to account for flow leaking through the prescribed walls, the flow through the channel is analyzed over time. In figure 8.12 the flow through the channel is plotted over time for three cut planes perpendicular to the channel. The location of the slices perpendicular through the channel can be seen in figure 8.6. The planes are close to the inlet and outlet of the channel, as well as in between. Additionally, the flow is integrated over the positive and negative flow areas. In the same way, the sign of the integrated volume distinguishes between an inflow and an outflow through the channel, where negative values correspond to an outflow of blood. While the flow nearly perfectly matches the solution calculated based on the underlying geometry (following equation 8.9), also the integration of the flow delivers convincing results: the deviation of $0.035\,\text{cm}^3\text{s}^{-1}$ is small compared to the overall integral. As another verification step, the fluid volume leaving the computational grid outside the channel is summed up. While the movement of the cylindrical wall during the inflation process not only sucks blood into the enclosed surface, the same amount of volume has to be displaced from the outside area. Following the boundary conditions for the walls, the volume leaves the grid at the lower

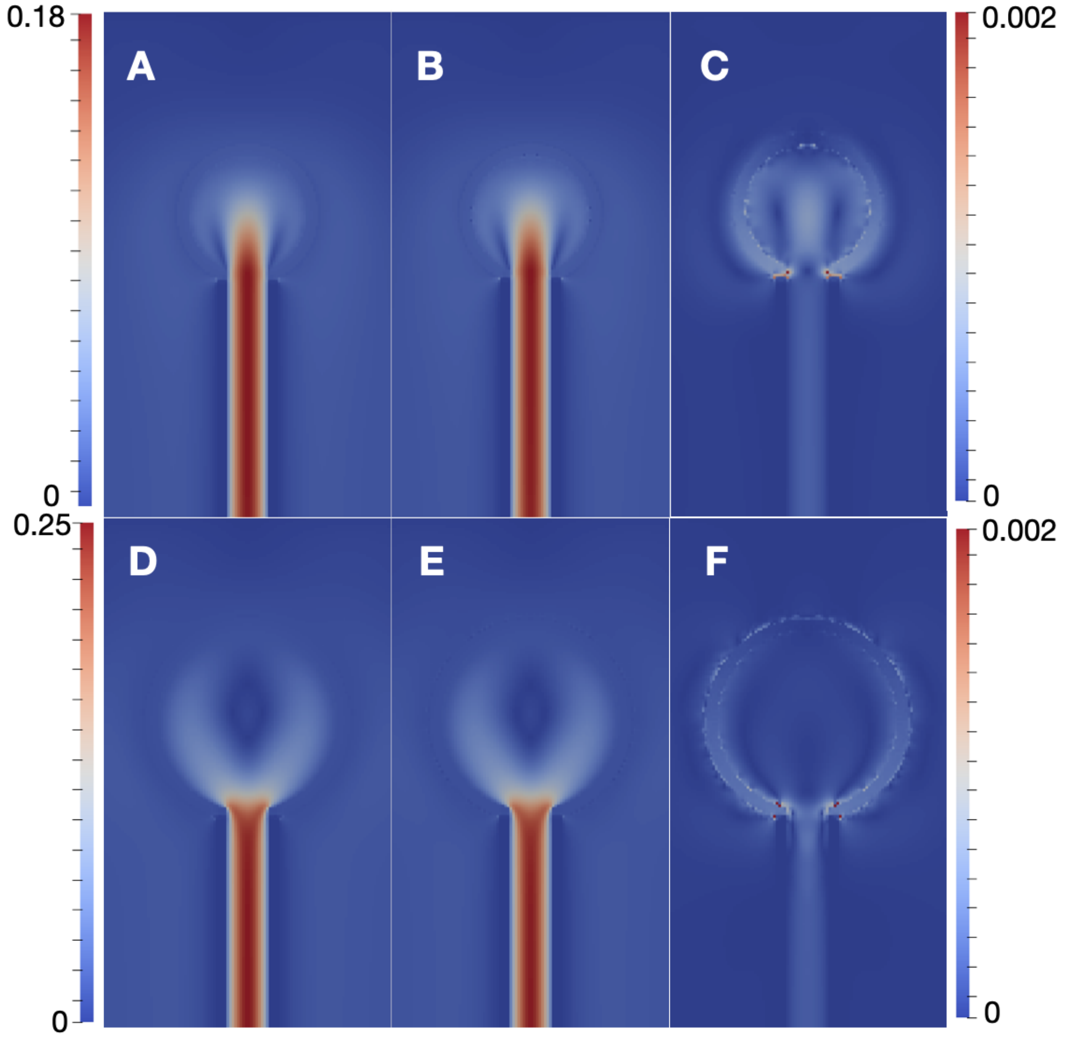


Figure 8.10: Velocity field u in m/s for the whole simulated grid. Analytic function (A) and external file (B) prescription as well as their difference $\Delta u = u_A - u_B$. The upper row is extracted during inflation of the cavity, while the lower row shows the deflation process.

wall Ω_{io} of the grid. For mass balance reasons, the amount of fluid leaving Ω_{io} outside the channel has to be inverse to the amount entering through the channel. Figure 8.12 shows the flow outside the channel in green color and verifies that this condition is fulfilled throughout the whole time series.

Computing the volume difference obtained by the integration of the flow and using the averaged flow over the last two corresponding cycles (annotated in figure 8.12) delivers

$$\frac{50.303\,cm^3s^{-1} - 50.268\,cm^3s^{-1}}{50.268\,cm^3s^{-1}} = \frac{0.035\,cm^3s^{-1}}{50.268\,cm^3s^{-1}} = 6.96 \cdot 10^{-4} < 0.07\% \tag{8.18}$$

Deviations in this order of magnitude are neglectable compared to other influences.

In figure 8.13 the comparison between the analytically and the externally prescribed wall is shown. For both scenarios, the flow through the cylinder is analyzed in three different

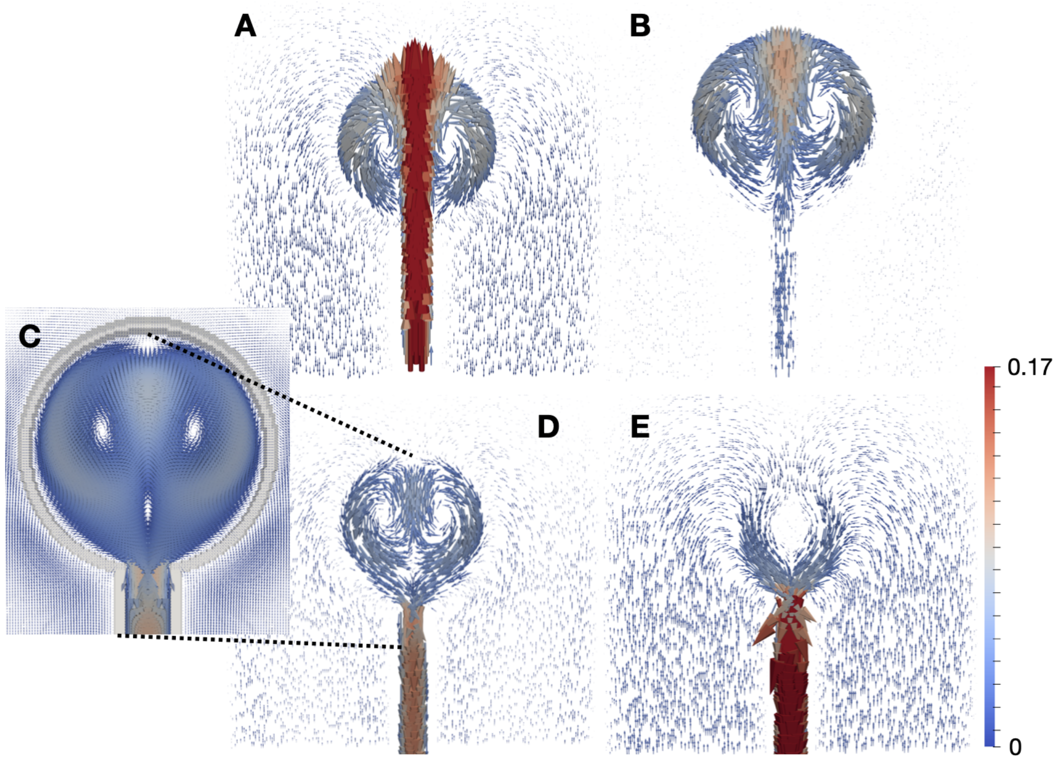


Figure 8.11: Vector plot representing the velocity u across the whole grid in ms^{-1} in different phases. The upper row shows the inflation process (fast inflation (A) and retarding inflation (B)), while the lower row figures are taken during deflation (beginning deflation (D) and rapid ejection (E)). On the left hand side (C) a zoom view is shown in which a velocity vector is included for each single cell.

heights (in relation to the inlet surface, for the exact position in the grid see figure 8.6A) to ensure that no fluid is lost throughout the channel. The different heights do not imply any difference, as the lines fully overlap. The deviation between the analytic and the externally prescribed geometry is so little that it can hardly be recognized.

In the last simulation setup with focus on the inflating cylinder geometry, the shape of the prescribed wall is optimized. Up to now, the round cylinder and the rectangular channel are simply assembled side by side to one another. This results in a square edge at the transition from the cylinder to the channel. Thus, the fluid has to flow around a sharp corner when entering the cylinder and recirculation may be supported. Therefore, a smooth transition from the channel geometry to the circular cylinder shape is introduced. The resulting indicator field is shown in figure 8.14B. The influence on the velocity field is shown in the two bottom rows of figure 8.14. On the left hand side of the figure, the standard geometry is shown, while on the right the results for the optimized shape are depicted.

It can be seen, that during deflation (center row in figure 8.14) higher velocities occur in the transition area from the cylinder to the channel if the sharp edges are used (C). Especially, the acceleration of the ejected fluid occurs already at a higher position in the channel. With the smoothed transition area also the resulting flow field is smoothed. The flow follows the

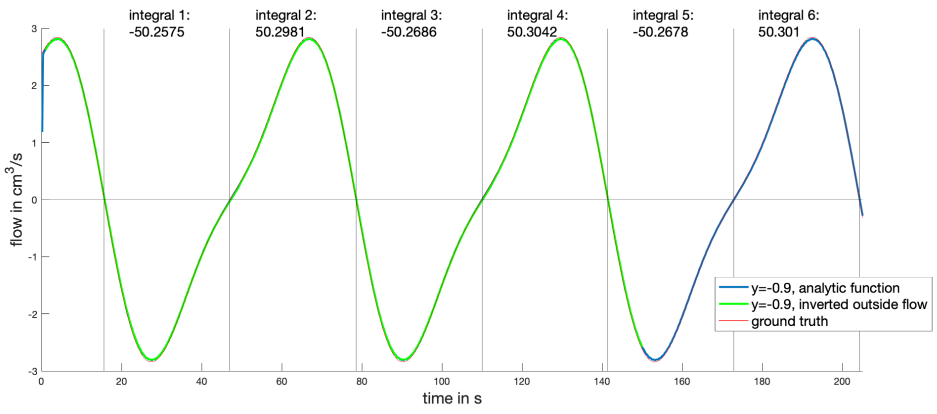


Figure 8.12: Time resolved flow resulting in the channel in cm^{3}s^{-1}. The numbers above the time course denote the flow integrated over time for a plane perpendicular to the channel inflow and outflow, located close to the lower end of the channel ($y = -0.9$ m). While positive signs of the integrated volume denote an inflow in the cylinder, the negative sign indicates an outflow.

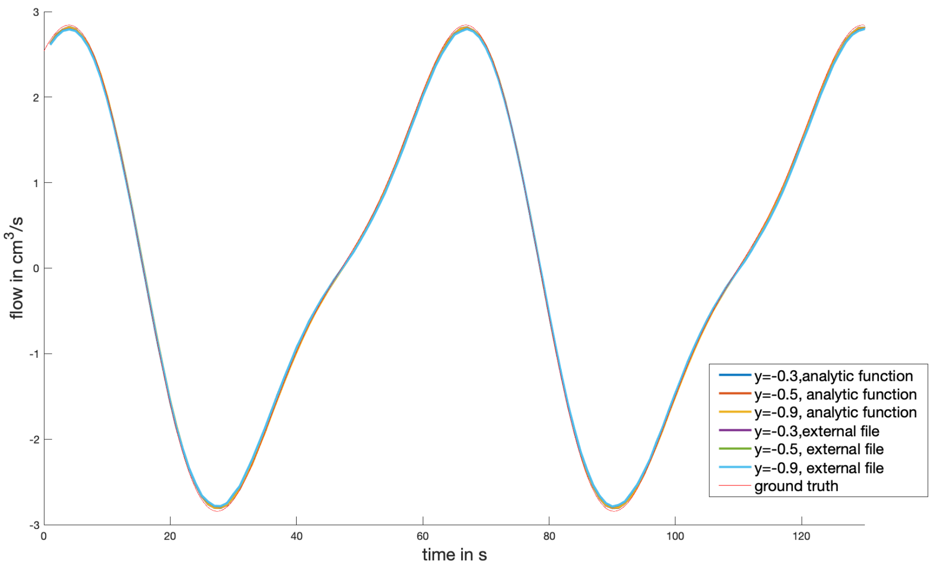


Figure 8.13: Time resolved flow resulting in the channel in cm^3s^{-1}. Comparison of different cut plane heights along the cylinder and for the analytic function and the external file prescription geometry.

shape prescribed by the wall. During inflation the same phenomenon can be observed in a less strong occurrence. While the fluid enters through the channel, the spread throughout the cylindrical shape starts with a minimal offset in height (E). The transition from channel to cylinder geometry is more smooth in the optimized wall (figure 8.14F).

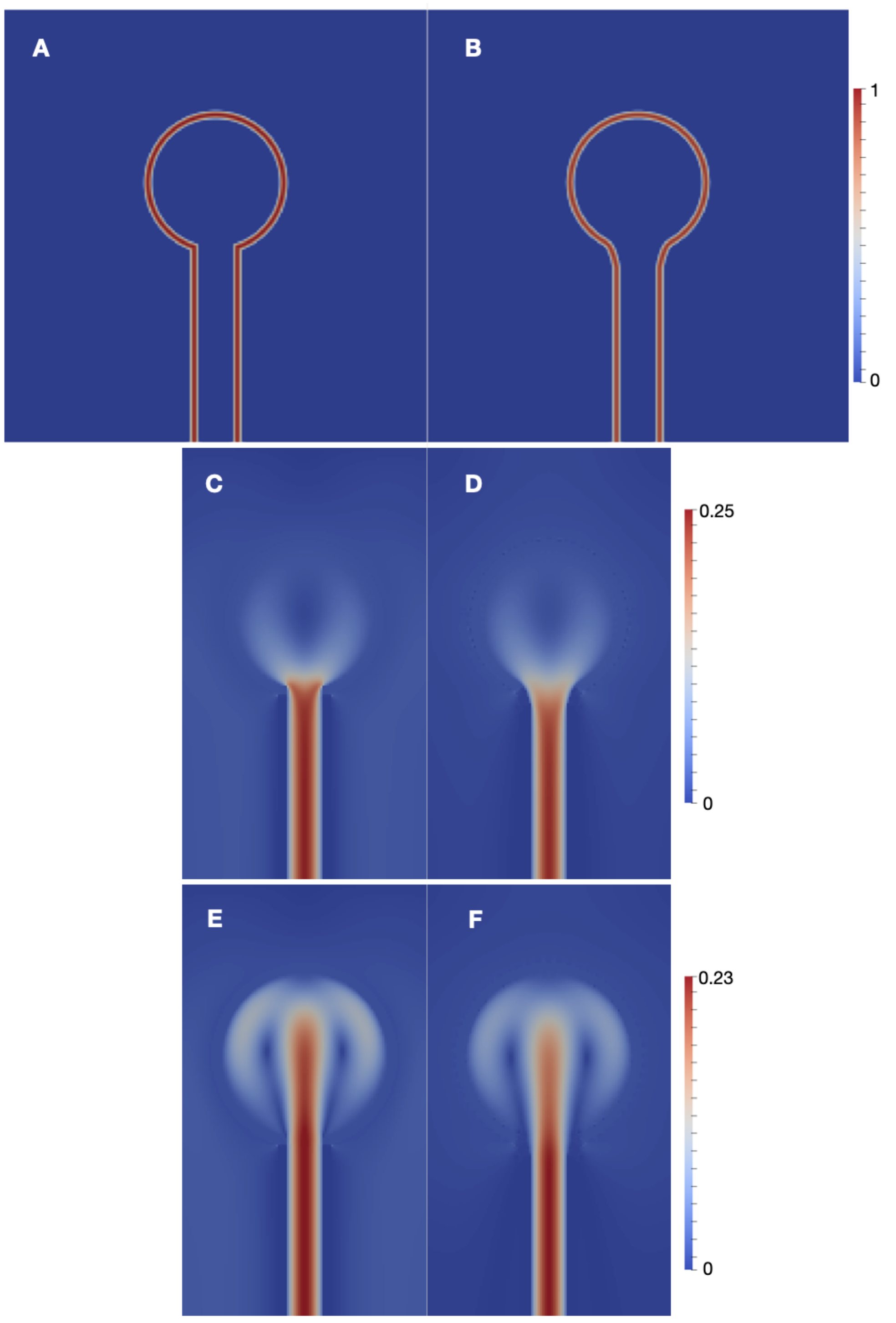


Figure 8.14: Upper row: indicator, lower and center row velocity in $m\ s^{-1}$. Comparison of round and edge corners.

8.3.3 Interpretation and discussion

Simulating a periodically inflating and deflating cylinder geometry, a first simplified 2D scenario simulated under similar conditions as required by the deforming left ventricle (LV) in 3D cardiac simulations is evaluated. While the deformation of the fluid boundary triggers an inflow or an outflow of fluid through the attached channel by the change of the enclosed volume, the simulation setup showed that blood is sucked into the cylinder (or pushed out of it) in a periodical manner as expected. If no deformation of the geometry is prescribed, the flow also comes to a still-stand. Thus, the basic functionality of the heart to drive blood through the circulatory system like a mechanical pump is fulfilled. Also the resulting flow patterns show the expected behavior: The maximum flow velocity is reached in the channel during deflation as well as during inflation phase due to the lower width compared to the cylinder. A laminar flow profile with a decay towards the channel walls in x-direction can be observed. Additionally, during the inflow of the fluid a half-circular distribution towards the wall occurs from the top of the cylinder towards the lower ends, while the wall moves outwards. On the left and the right hand side of the inflow jet, a stasis area occurs on the horizontal center line of the cylinder, as the fluid continues to flow in y-direction due to mass inertia. During deflation, the stasis area is located in the center of the cylinder as the fluid is symmetrically pushed into the channel due to the movement of the constraining walls.

This simulation scenario proved, that no flow leaves the geometry throughout the walls. Thus, a major requirement towards the application of the IBM solver for cardiac modeling is fulfilled.

Additionally, the simulation results clearly show that the deviations between the analytic and the externally prescribed geometry are only present on a really small scale. These deformations are induced by the mesh resolution of the constraining surface file and could be further decreased by a decrease of the edge length in the prescribed geometry. Another possibility would be to increase the grid mesh resolution, as the deviation between the cell center and the prescribed geometry can be decreased further. However, the deviations between the two methods are already below the mesh resolution in every time step and thus neglectable compared to other mesh resolution errors. Therefore, a further harmonization of the compared wall prescription methods does not yield any further benefit.

In both cases, with a wall prescribed by the analytical function as well as a wall prescribed by the external files, the resulting flow meets the ground truth flow nearly perfectly at each point in time of the periodic cycle. Thus, the simplified simulation setup of the 2D inflating cylinder proves the basic fulfillment of all relevant requirements for modeling a moving wall by the IBM approach.

8.4 Inflating sphere

8.4.1 Simulation setup

This simulation setup is the next step in dimension compared to the inflating cylinder geometry presented in 8.3: The circular 2D cylinder geometry that models the blood cavity is extended to a 3D shape. While the inflating cylinder is replaced by a sphere with the same radius r, the channel that supplies the inflow and outflow is replaced by a cylinder. This cylinder is the 3D equivalent of the channel and covers the radius r_c. The underlying basic grid is extended to form a cube with equal sides. The mesh is utilized in a block structured implementation with a refinement of second order (one cell is divided in $4\times$ 4 = 16 cells) in the whole internal area. The grid orientation is set in correspondence with the 2D case: To ensure comparability, the cylinder height is furthermore oriented in y-direction. The circular diameter of the cylinder is thus located in the x-z plane.

In this geometrically simple setup, the ground truth solution for the expected flow again can be calculated analytically by the variation of the geometrical surface. The volume of the inflating sphere, as well as the volume of the flow supplying cylinder have to be summed up.

$$V = \frac{4}{3}\pi r^3 + \pi(r_c - \frac{w}{2})^2 \cdot (y_m - r) - V_{r,h} \tag{8.19}$$

In the overlapping region, the volume of the spherical segment $V_{r,h}$ is subtracted based on the height h of the segment.

$$V_{r,h} = \frac{\pi}{3} \cdot h^2 \cdot (3r - h) \tag{8.20}$$

$$h = r - \sqrt{r^2 - r_c^2} \tag{8.21}$$

The resulting flow in the channel can be computed based on the derivative $\frac{\partial V}{\partial t}$.

Switching to a 3D sphere causes the volume change and thus the expected velocity to change proportional to the third power of the radius ($q \propto r^3$). If the cylinder radius r_c is chosen to be equal to the half channel width ($0.5d$), the velocity in the cylinder ($q \propto r_c^2$) will exceed the velocity in the inflating cylinder test case scenario. This is not intended, as the Reynolds number would increase (re>100) and turbulent influences may occur. Therefore, the radius r_c of the cylinder is adapted, such that the expected flow is in the range of the 2D inflating cylinder scenario. This also allows a comparison of the resulting velocity profiles between the 2D and the 3D case. The radius r_c is calculated based on the resulting cylinder surface A_c. The velocity u_c through the cylinder is prescribed. As the main geometrical changes appear in the inflating sphere itself, the overall volume can be approximated by the sphere volume to simplify the computation of the expected flow (only for the radius r_c calculation, not for the ground truth).

This results in:

$$V \approx \frac{4}{3}\pi r^3 = \frac{4}{3}\pi(r_0 + dr \cdot sin(\omega t))^3 \tag{8.22}$$

$$\frac{\partial V}{\partial t} \approx 4\pi(r_0 + dr \cdot sin(\omega t))^2 \cdot dr \cdot \omega \cdot cos(\omega t) \tag{8.23}$$

$$u_c \approx \frac{\partial V}{\partial t} \cdot \frac{1}{A_c} \tag{8.24}$$

For the initial time step $t_0 = 0$, in which the maximal flow through the channel occurs (due to $\cos(\omega t_0) = 1$), the velocity u_c results as

$$u_c \approx 4r_0^2 \cdot dr \cdot \omega \cdot \frac{1}{r_c^2} \ . \tag{8.25}$$

Assuming a flow $u_c \approx 0.25\,\text{m/s}$ to allow comparability with the 2D case, the radius of the cylinder can be computed based on equation 8.25 as $r'_c = 0.15\,\text{m}$. However, it has to be taken into account that this is the necessary effective radius which must be available to the flow. Due to the thickness of the impermeable wall cells located inwards from the wall center line (half the wall thickness as found in section 4.3), the effective value has to be multiplied with the correction factor resulting from table 3.2:

$$r_c = \frac{r'_c}{76.7\,\%} = \frac{0.15\,\text{m}}{76.7\,\%} = 0.2\,\text{m} \ . \tag{8.26}$$

The radius r of the sphere is adjusted to always meet the cylinder width at least, such that

$$r(t) = \max(r(t), r_c) \ . \tag{8.27}$$

The wall position and the wall velocity U_w are again prescribed by an analytical function in this simulation setup. The underlying distance field Λ is calculated based on the following equations.

Therefore, the ED introduced in equation 8.12 is extended in the third dimension for each cell center $\mathbf{x}_i = (x_i, y_i, z_i)$:

$$\text{ed}_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2 + (z_i - z_c)^2} \ . \tag{8.28}$$

The distance Λ results in analogy to equation 8.11 under consideration of the condition in equation 8.27:

$$\Lambda_i = \begin{cases} |\text{ed}_i - \max(r(t), r_c(t))| & y_i > y_c \\ |\min((\sqrt{(x_i - x_c)^2 + (z_i - z_c)^2} - r_c), (\text{ed}_i - r(t)))| & \text{else} \end{cases} \tag{8.29}$$

The indicator field Φ is calculated from the distance field Λ as previously, following equation 4.3. The wall velocity vectors are assigned based on the change of the radius $r(t)$ as introduced in the 2D case in equation 8.13 also considering the extension to the third dimension:

$$\gamma = \tan^{-1}\{\frac{y_i - y_{c,i}}{x_i - x_{c,i}}\} \qquad \gamma_1 = \tan^{-1}\{\frac{x_i - x_{c,i}}{z_i - z_{c,i}}\} \tag{8.30}$$

$$\gamma_2 = \tan^{-1}\{\frac{z_i - z_{c,i}}{x_i - x_{c,i}}\} \qquad \gamma_3 = \tan^{-1}\{\frac{y_i - y_{c,i}}{z_i - z_{c,i}}\} \tag{8.31}$$

$$U_{w,x} = \frac{\partial r}{\partial t} \cdot \cos(\gamma) \cdot \sin(\gamma_1) \tag{8.32}$$

$$U_{w,y} = \frac{\partial r}{\partial t} \cdot \sin(\gamma) \cdot \sin(\gamma_3) \tag{8.33}$$

$$U_{w,z} = \frac{\partial r}{\partial t} \cdot \cos(\gamma_3) \cdot \sin(\gamma_2) \tag{8.34}$$

8.4.2 Simulation results

In figure 8.15 the cubic grid is cut in x- and z-direction through the center point of the grid.

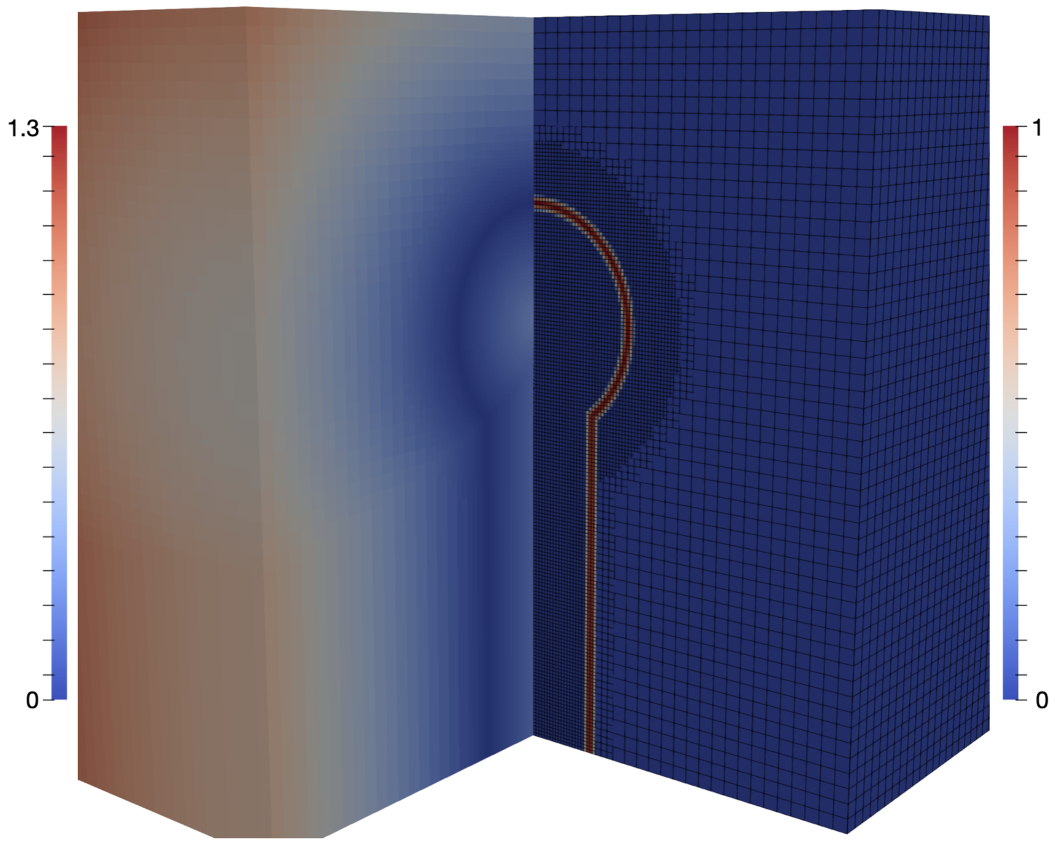


Figure 8.15: 3D grid with cut planes in x and z direction. Left: distance field Λ in m. Right: indicator field Φ computed from the distance field.

On the left hand side of the grid the distance field Λ is plotted, while on the right hand side the resulting indicator field Φ is shown. While the grid edges are visualized in the right part, the refined area can be seen. Distance and indicator field match the 2D case with the exception that the 3D channel is wider, as it was prescribed by the analytical function.

In figure 8.16 the cubic grid is clipped along the z-axis and depicts the indicator field with the cell edges being visible. In addition, in x-direction the indicator field with a threshold value of $\Phi > 0.2$ is shown, resulting in the half-spherical reddish shape. While the inner cells are dark red, with a rising distance from the wall's center line, the indicator value decreases. Finally the prescribes wall velocities U_w are depicted by vectors on top of the spherical surface. All in all, the fully symmetric results reproduce the 2D case in 3D as expected.

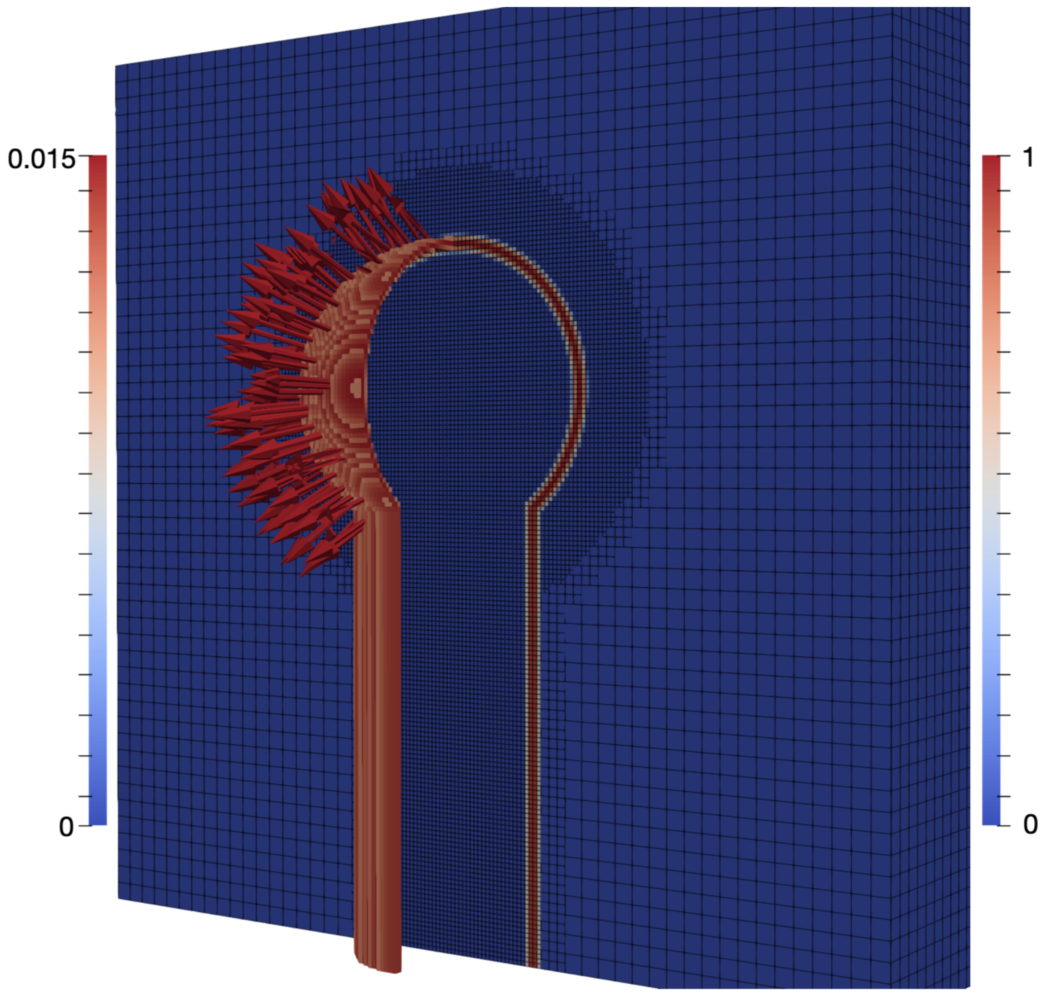


Figure 8.16: 3D grid with cut planes in x and z direction. Indicator function Φ plotted with underlying grid edges in z-plane (right hand side). In the x-plane, the thresholded indicator ($\Phi > 0.2$) field resulting in a cut spherical red shape is shown. Additionally, the resulting vectors of the prescribed wall velocities are shown.

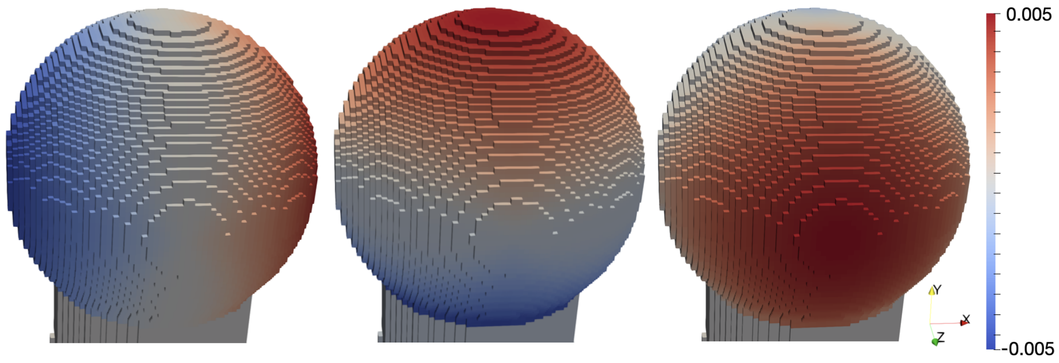


Figure 8.17: Prescribed wall velocity U_w magnitude, split up in the three spatial directions $U_{w,x}(left), U_{w,y}(center)$ and $U_{w,z}$ (right) in ms^{-1}. For easier comparability, all grid cells are thresholded such that $\Phi > 0.8$ and the same view angle is kept for all directions.

The wall velocities U_w as computed in equation 8.34 are visualized on the geometry in figure 8.17. For better comparability, only the wall cells ($\Phi > 0.8$) are visualized. The maximum velocities of $U_{w,max}$ are reached for the highest distances in the respective direction - either in positive or negative direction. In central areas (e.g., for a point $x_i \approx 0$) the resulting velocity in the corresponding direction equals zero ($U_{w,x} = 0$). The resulting fields again are completely symmetric.

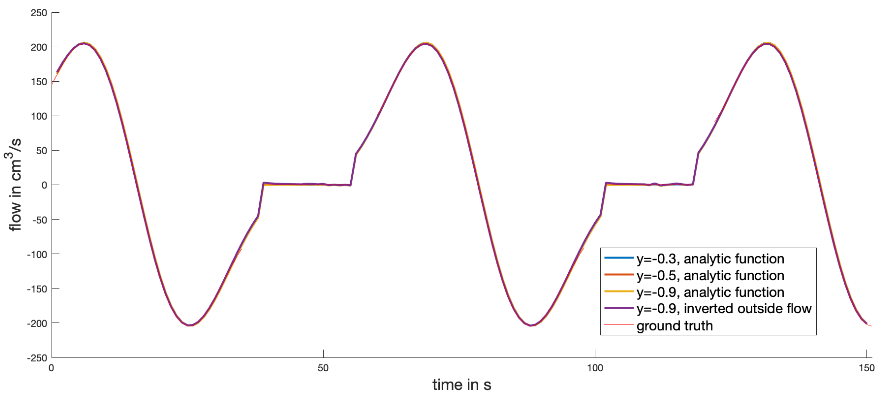


Figure 8.18: Time course of the flow through the cylinder in cm^{3}s^{-1}. While positive values denote an inflow of fluid through the cylinder into the sphere (inflation process), negative values correspond to an outflow. The flow is plotted in different heights from the channel end (compare 2D inflating cylinder scenario) to evaluate the loss throughout the cylinder walls. All values are compared against the computed ground truth (equation 8.19) and the inverted flow leaving the computational grid outside the channel.

The time course of the flow through the supplying cylinder is shown in figure 8.18. While positive values denote an inflow into the sphere (during inflation), negative signs arise in the deflation process when blood is ejected out of the sphere. Flat areas (no flow) denote the phases, in which the minimum sphere radius (equal to the cylinder radius, $r(t) = r_c$) is reached due to the constraining equation 8.27. The three different heights located close to the front and the end of the cylinder, as well as in between (compare also the 2D inflating cylinder case in figure 8.6) are evaluated. While no deviation at all can be seen in the comparison of the three evaluated heights, no fluid leaves the cylinder through the wall. Therefore, the impermeability of the wall is proven. Additionally, the comparison with the ground truth value computed based on equation 8.19 delivers accordant results. Small deviations at the start of the simulation vanish after the first full cycle. Thirdly, the comparison with the fluid volume that leaves or enters the inlet and outlet surface Ω_{io} outside of the cylinder is drawn (purple line). While this flow is inverted compared to the flow entering and leaving through the cylinder, both magnitudes agree. This again proves correctness of the inflow and outflow volumes as well as the impermeability of the wall.

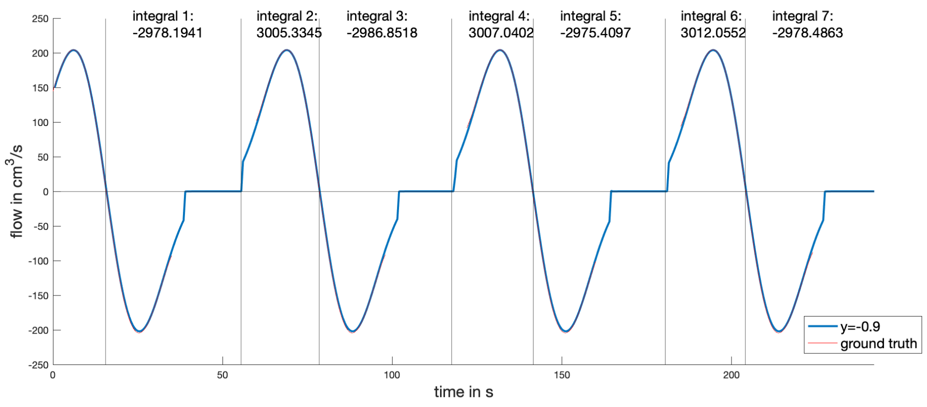


Figure 8.19: Flow through the attached channel for a series of four cycles. In addition, integration of the flow in the compartments.

In addition to this, the integration of the flow through the cylinder is evaluated in the last two cycles of figure 8.19. Comparing the inflow volume to the outflow volume shows an agreement also in the 3D inflating sphere scenario. The deviation between the volumes is in a low range and smaller than 1.2 %.

Finally, the velocity field u of the inflating sphere scenario is compared with the 2D inflating cylinder scenario (left hand side) in figure 8.20. The resulting flow patterns are of the same basic appearance, while at specific locations varied effects can be observed. Comparing the begin of the inflow phase (top row), the velocity in the 3D scenario (B) is higher. This is the case, even if the supplying cylinder is enlarged compared to the 2D channel and is due to the faster volume change in the 3D scenario. However, the expected

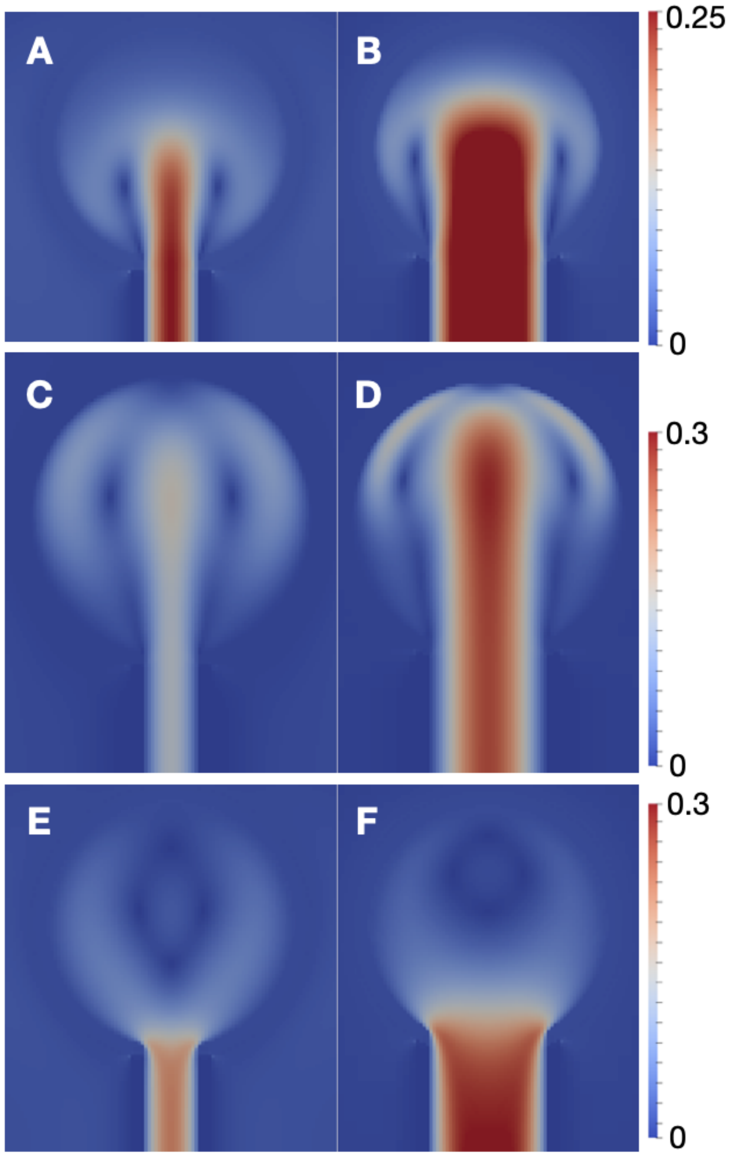


Figure 8.20: Comparison of the velocity field u in the 3D inflating sphere scenario (right hand side) to the 2D inflating cylinder scenario (left hand side). Three specific points in time are shown exemplary: during the begin of inflation (top row) and towards the end of the inflation process (center row), as well as during deflation (bottom row).

pattern with the maximum velocity in y-direction that leaves the supplying cylinder and then spreads in a circular manner towards the wall is retained in the 3D scenario. This holds true also for the end phase of the inflation process (center row of figure 8.20). While the velocity magnitude in the 3D case is higher than in the 2D case, areas with nearly resting flow occur close to the maximum velocity inflow area. During deflation (bottom row) maximum flow velocities can again be observed in the outlet cylinder (F) and channel (E). The fluid ejected in that phase is in both scenarios accelerated directly in the wall area and thus a resting fluid

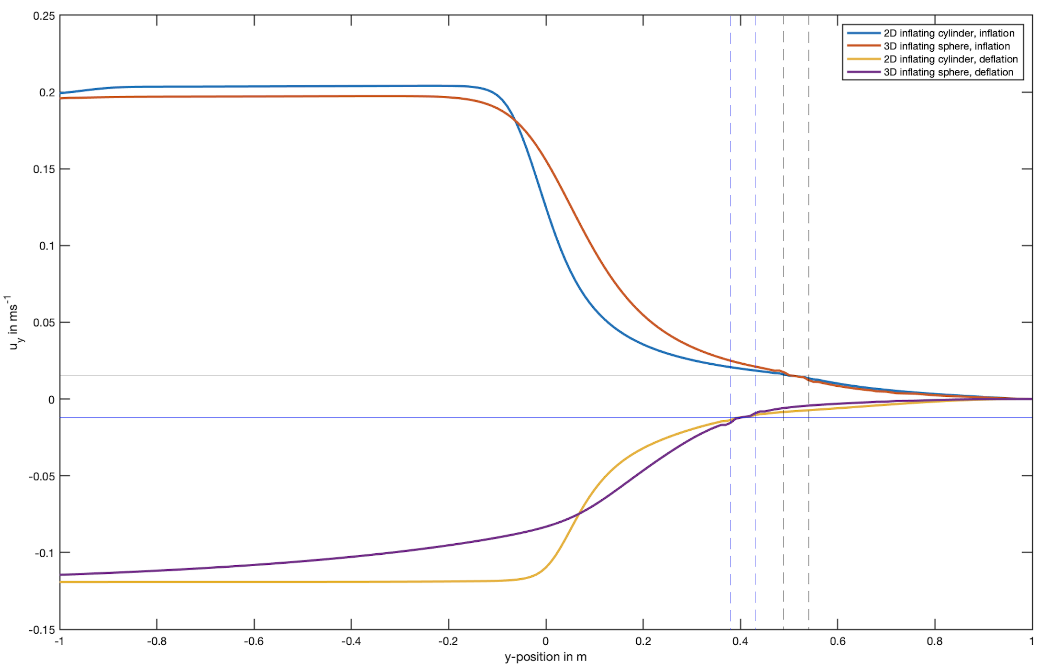


Figure 8.21: Velocity u_y plotted over the spatial coordinate in y-direction starting from the bottom of the computational grid ($y = -1$ m). The x-coordinate is set to $x = 0$ m, as well as the z-coordinate $z = 0$ m in the 3D scenario. Comparison for two different points in the cycle (deflation and inflation), as well as for the 2D scenario and the 3D scenario. The dashed vertical lines denote the position of the prescribed wall (inflation in black, deflation in blue color). The horizontal line denotes the prescribed velocity in the same color than the wall position.

region forms in the upper part of the deflating sphere, respectively cylinder. In the deflating sphere (F), the region is smaller due to the higher speedup of the fluid.

For a direct comparison, the fluid velocity u_y in the y-direction is plotted over a line ($y \in (-1, 1)$ m, $x = 0$ m and $z = 0$ m) in figure 8.21. The location of the prescribed wall is denoted by dashed vertical lines, while the prescribed velocity in this region is visualized by the solid horizontal lines for the inflation process (black color) and the deflation process (blue color). On the right edge of the figure, in all cases the prescribed velocity at the edge of the computational domain $u_{\Omega_{\text{patch}}} = 0$ ms^{-1} is met. The prescribed wall velocities U_w are perfectly met in the center of each wall. Towards the ends of the prescribed walls, where the prescribed indicator Φ already decays, an increase (respectively decrease) in velocity can be observed. After a transition phase through the 2D cylinder or the 3D sphere volume a steady flow through the 2D channel respectively the 3D cylinder occurs. It can again be confirmed that the flow magnitudes vary, while the overall basic structure corresponds the expected profiles in both cases.

8.4.3 Interpretation and discussion

In the inflating sphere scenario, the 2D inflating cylinder scenario was successfully extended to the 3D grid. While the flow patterns observed in the 2D setup have been reproduced in the 3D simulations, a good accordance with the expected flow computed from the geometric volume changes was proven. The integration of the in- and out-flowing fluid volume showed a good accordance and no flow left the geometry through the prescribed moving walls.

However, the loss of the effective volume that is introduced by the prescription of the wall velocity has to be considered.

Chapter 9

Grid independence study

9.1 Methods

To ensure that the chosen grid resolution does not falsify any results, its influence on the simulated quantities is assessed in this chapter. Therefore, the previously introduced 3D setup of the inflating sphere scenario is analyzed regarding varied mesh resolution. The underlying cubic computational grid is kept constant at the size of $2\ \text{m} \times 2\ \text{m} \times 2\ \text{m}$. The basic resolution for all scenarios is 50 cells in each grid direction, resulting in a basic edge length of 4 cm. In the relevant region of the sphere, this basic edge length (refinement level 0) is refined. In each refinement step, the concerned cells are divided in two cells in each coordinate direction, resulting in $2^3 = 8$ refined cells per initial cell. The resulting edge length and the total number of cells for the three investigated refinement stages are listed in table 9.1. The setup previously evaluated in section 8.4 corresponds to refinement level 2 and is referred to as the standard resolution. Accordingly, refinement level 1 is a coarser mesh and refinement level 3 a more fine mesh.

refinement level	cell division	edge length in cm	number of cells	number of nodes
1	8	2	190,128	203,233
2	64	1	722,100	757,458
3	512	0.5	4,924,144	5,048,498

Table 9.1: Quantitative mesh metrics for the three refinement stages of the 3D inflating sphere scenario. In one refinement step, each concerning cell is divided in $2^3 = 8$ cells.

9.2 Results

To ensure comparability to previous result sections, the 3D grid is always clipped centrally along the z-axis ($z = 0\,\text{m}$). In all result figures, in which quantities are plotted across the inflating sphere geometry, the standard case (refinement level 2) is visualized in the center column, while the coarse mesh (refinement level 1) is shown on the left and the fine mesh on the right hand side.

The resulting grid resolution for the three refinement stages, as well as the refined area is visualized in figure 9.1, highlighted by the dark blue cell edges. In addition, the resulting indicator field Φ is color-coded in this figure. In the coarse mesh, only two neighboring cells form the wall of the supplying cylinder (in x-direction). Thus, a cosine-shaped decay of the indicator field as prescribed by equation 4.3 is not possible: the maximum value of $\Phi = 1$ is not reached in the cylinder wall. In the sphere wall region, Φ does not fulfill the requirement of two diagonally connected cells with an indicator field value $\Phi > 0.5$. Thus, flow may leave through the wall.

In comparison, the fine grid on the right hand side enables a higher resolution of the cylinder wall cells: In x-direction 8 cells model the wall and enable for a graded indicator field. On the one hand, this allows a more detailed modeling of the wall. On the other hand it would allow for a reduction of the wall thickness w prescribed in equation 4.3. However, in comparison to the standard resolution, the indicator field Φ of the fine grid does not reveal significant shape differences.

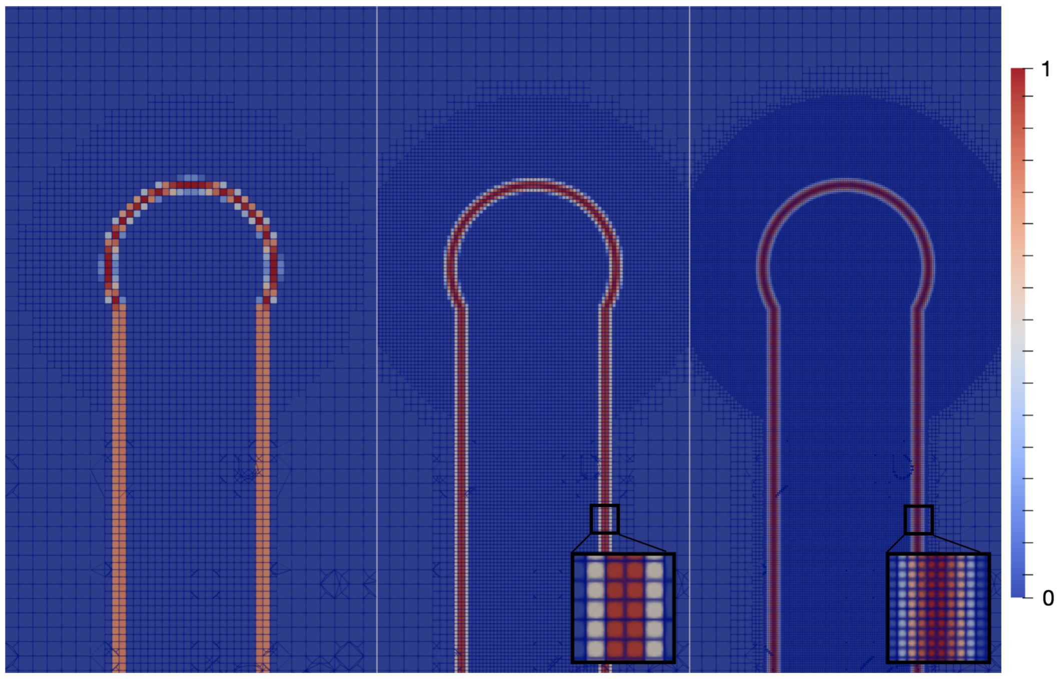


Figure 9.1: Resulting indicator field Φ for the three refinement stages. Left: coarse mesh, center: standard mesh, right: fine mesh. In the background, grid size and refinement area are highlighted by the dark blue cell edges. In the respective right corner, a close-up of the cylinder wall indicator field is shown.

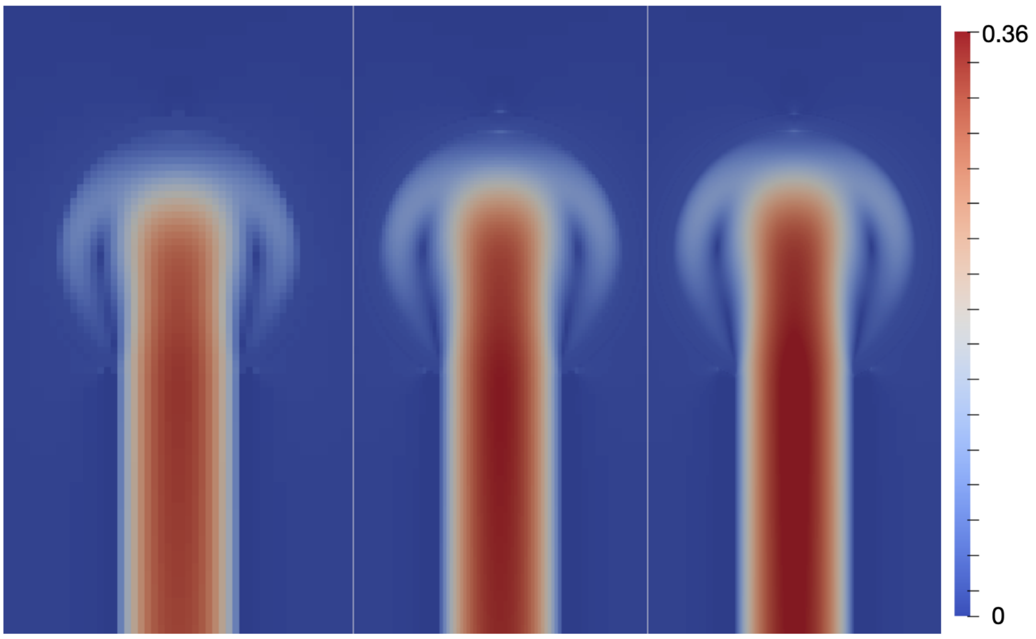


Figure 9.2: Resulting velocity field u for the three refinement stages in m s^{-1}. Specific point in time during inflation. Left: coarse mesh, center: standard mesh, right: fine mesh.

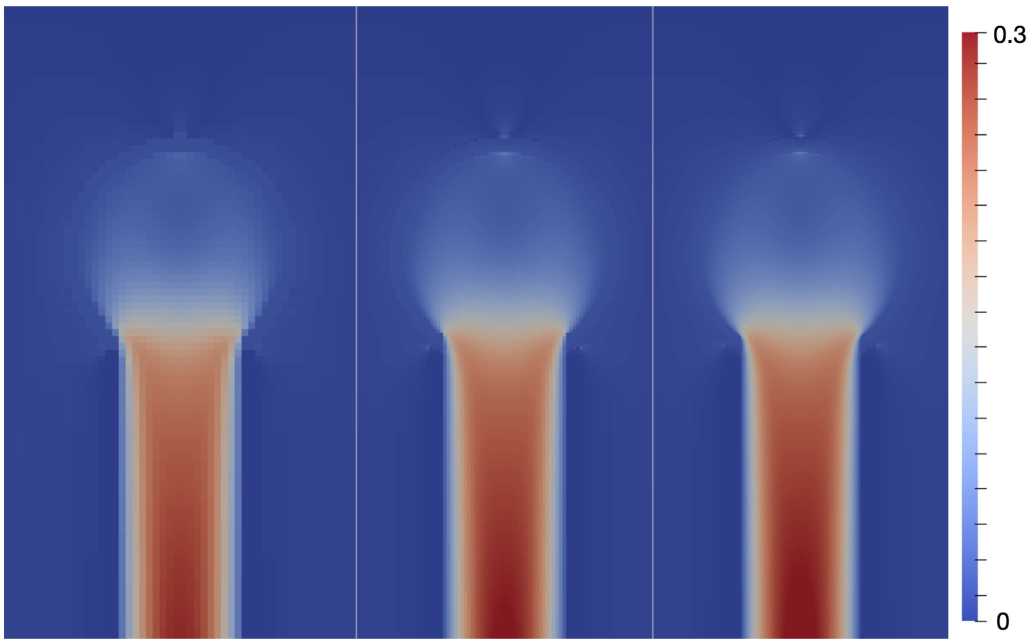


Figure 9.3: Resulting velocity field u for the three refinement stages in m s^{-1}. Specific point in time at the beginning of the deflation phase. Left: coarse mesh, center: standard mesh, right: fine mesh.

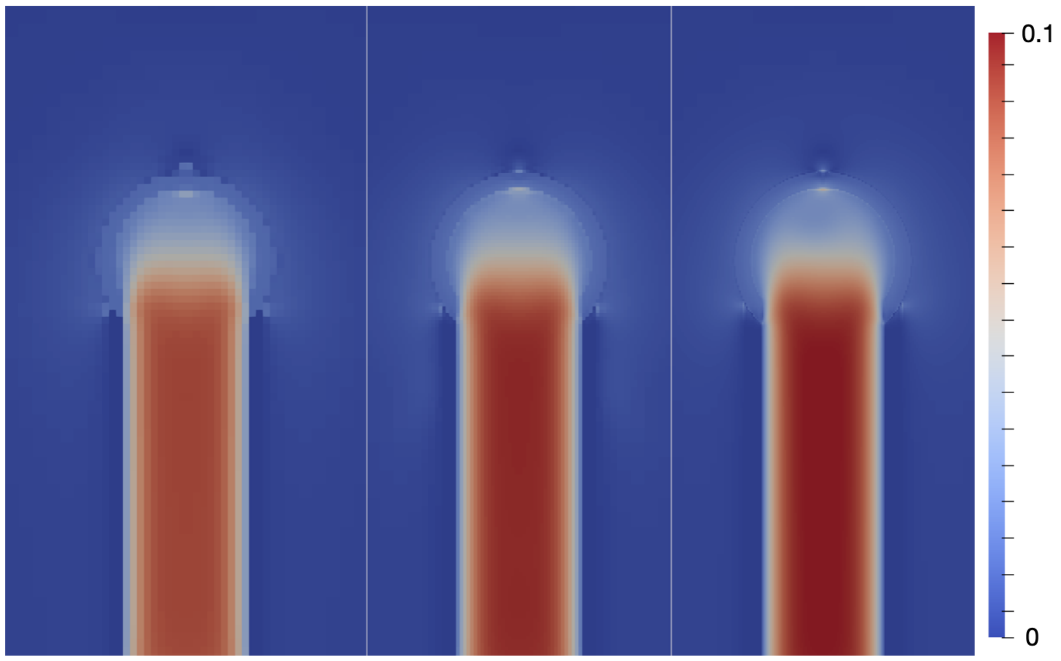


Figure 9.4: Resulting velocity field u for the three refinement stages in ms^{-1}. Specific point in time at the end of the deflation phase. Left: coarse mesh, center: standard mesh, right: fine mesh.

In figure 9.2, the resulting velocity field is plotted for the corresponding cut plane during the inflation of the sphere geometry. Between the standard and the fine geometry, almost no difference can be observed. While the jet seems to penetrate deeper into the sphere area of the fine grid simulation, a horizontal line shows the contrary and confirms the same penetration distances. This phenomenon also affects the coarse mesh result. However, in this case additionally the low resolution contributes to this effect. The velocity in the coarse mesh is slightly reduced compared to the other cases. The overall shape of the fluid patterns with the spread towards the sphere wall is obtained in all grids.

Figures 9.3 and 9.4 show the velocity fields during the deflation of the sphere. Again, all grid resolutions show the same overall flow patterns as expected. However, close to the end of the deflation phase (figure 9.4) the velocity in the coarse mesh is slightly reduced compared to the other cases. Between the standard and the fine grid resolution, no difference is observed.

Finally, the time course of the flow through the cylinder is shown in figure 9.5. While all lines are symmetric, only the standard grid resolution matches the ground truth solution perfectly. In the coarser mesh, the computed flow rate is slightly increased. On the other hand, the flow rate is slightly decreased in the finer mesh. As discussed in the following subsection, this may occur due to a variation in effective cylinder diameter. The plateau phases in which no flow leaves or enters the cylinder are due to the minimum sphere radius (equal to the cylinder radius, $r(t) = r_c$), caused by the constraining equation 8.27 as explained in section 8.4.

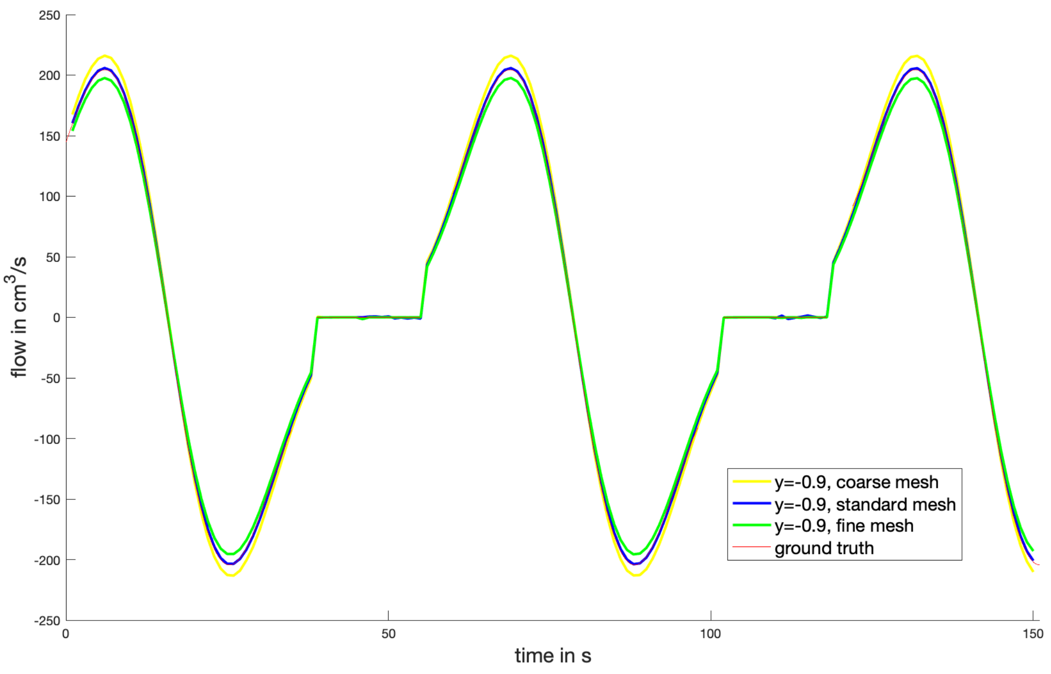


Figure 9.5: Time course of the flow through the cylinder for the coarse, standard and fine mesh resolution. For comparison, the geometrically computed ground truth solution is shown in red color (overlaps with the standard mesh).

To conclude this study on the grid resolution influence, the consumed computational resources are listed in table 9.2. All simulations were carried out on the bwUniCluster (2.0) to use the parallel computation infrastructure. In each case, a division in $4 \times 4 \times 4$ blocks was applied, resulting in a parallel computation using 64 cores. Simulation duration includes two and a half deflation and inflation cycles (as shown in figure 9.5), up to $t = 150$ s. The listed job wall clock times include the decomposition and reconstruction of the grid.

refinement level	number of cells	wall clock time in hh:mm:ss
1	190,128	00:18:16
2	722,100	00:54:33
3	4,924,144	09:42:54

Table 9.2: Consumed simulation resources for coarse, standard and fine mesh resolution.

While the standard mesh grid including 720,000 cells is computed in below one hour, the fine grid results needed the ten fold amount of computational time. The coarse mesh was simulated completely in below 20 minutes.

9.3 Discussion

Concerning the visual comparison of flow patterns and the resulting velocity fields, no significant difference is observable while comparing the three mesh refinement stages. Even while evaluating the coarse grid, which revealed significant differences in the resulting indicator field Φ in figure 9.1, only slight variations in the velocity magnitude can be observed. However, comparing the flow rates over time (figure 9.5), a systematic offset of up to $5\,\%$ can be observed between the three grid refinement stages. These deviations may be caused by the variations in the effective diameter of the supplying cylinder geometry. While the indicator field Φ is computed from the very same distance field Λ in all scenarios, the refinement level has an influence on the resulting indicator field distribution and also the resulting radius of the channel $r_{\mathrm{c,eff}}$. Even if the parameter w is kept constant while computing Φ based on equation 4.3, the center of the coarser mesh grid cells may already be outside the assumed threshold w. Thus, in the coarser grid, the higher edge length leads to a wider channel diameter, resulting in a lower absolute velocity but a higher flow. The wall is effectively thinner. This phenomenon arises the other way around in the fine mesh and explains an increased flow rate while the velocity magnitudes are the same. Because the wall thickness is already considered in the computation of the geometrical ground truth solution, it matches the standard grid resolution best.

Summing up, mesh resolution did not play a key role for the resulting flow patterns in the investigated scenarios. With all considered resolutions, global effects were successfully reproduced. However, when comparing absolute values, this has to be done with caution, taking into account the variation in effective volume deviations and geometrical diameters resulting in increased or decreased volume flow rates. Concluding, the finer mesh did not reveal a benefit over the standard mesh resolution. As the standard mesh requires less than one tenth of fine grid computation time and way less memory, the standard resolution is preferred compared to a finer grid.

Chapter 10

3D heart simulations

As motivated in the introduction of chapter 4, one of the aims of this thesis project is the assessment of a fluid dynamics solver based on the immersed boundary method (IBM) to model the fluid dynamics in the human heart. While up to now, only simplified scenarios have been evaluated, the 3D geometry of the left part of the human heart is assessed in this chapter. Therefore, blood flow dynamics are simulated in a realistic 3D heart geometry obtained from measurement data. The assumed boundary conditions and prescribed geometric surfaces are obtained from the mechanical simulation framework. Fluid and pressure boundary conditions (BC) are prescribed based on the simulation results of the circulatory system. To verify the results, a comparison to the body-fitted mesh representation is drawn.

10.1 Methods

Simulations are carried out on a 3D cubic grid of the size $0.225\,\mathrm{m} \times 0.19\,\mathrm{m} \times 0.245\,\mathrm{m}$ with a basic edge length of approximately 6 mm. The fluid boundary is prescribed as in the scenarios evaluated before and immersed into this grid. To accurately model the areas of interest with a high enough resolution, the cells in the wall region, as well as in the valve region are refined. As the wall moves throughout the cardiac cycle, the diastolic and the systolic geometry are both used to define the refinement area. This process ensures that no additional dynamic remeshing is necessary and still all cells in the wall region are fine enough. The fluid boundary is prescribed by a series of 250 external stl geometries which model the endocardial wall movement throughout the whole cardiac cycle. The geometry consists of a single closed surface to model the whole left side of the heart. It comprises the left ventricle (LV) and left atrium (LA) endocardial walls, as well as the outer hull of the mitral valve (MV). In addition, the pulmonary veins, as well as the aorta are included with the prolonged trunks. For modeling the IBM scenario, the length of the trunks is increased such that the surface of the cubic mesh grid is cut. All necessary underlying geometries result from the mechanical simulation framework implemented by Fritz et al. [110]. The applied external surface file exemplary is shown for the initial state (end diastole) in figure 10.2.

Following the procedure suggested in chapter 4, the distance field Λ is computed based on the Euclidean distance (ED) between each cell center and the constraining geometry. The indicator field is computed based on equation 4.3 assuming a wall thickness of w=0.004 m.

Beneath introducing the simulation setup and results, one aim of this chapter is to compare the fluid dynamics solver based on the IBM to the previous body-fitted implementation. Taking into account the findings of chapter 8, for a comparison between both solvers, the cells constrained by the wall velocity prescription have to be considered, as they do not contribute to the flow any more. Thus, instead of the full enclosed volume, the effective volumes have to be aligned between the two scenarios. To enable a comparison, the prescribed endocardial surface wall is inflated in normal direction. After ensuring that all face normal vectors n_f point outwards of the geometry, for each grid point i, the point normal vector $n_{\mathrm{p},i}$ is computed as the average of all face normal vectors n_f adjoining the point. The shifted point position x_i can be computed from the initial position $x_{0,i}$ as

$$x_i = x_{0,i} + c_\mathrm{inf} \cdot n_{\mathrm{p},i} \ , \tag{10.1}$$

under consideration of a constant scaling factor c_inf which by default is chosen to meet the half wall thickness $c_\mathrm{inf} = 0.5 \cdot w$.

As the length of the trunks necessarily had to be increased to cut the edge of the computational domain, the prescription of the pressure at the inlet and outlet surface can no longer be applied. Based on the findings in chapter 7, instead of the pressure boundary conditions, the adequate flow is prescribed at the inlet and outlet surface. Therefore, the flow simulated by the circulatory system is prescribed in normal direction of the inlet and outlet surface. The computed normal vectors are visualized in figure 10.2. In addition, the outlet pressure is prescribed at the end of the aortic trunk.

Both valves are represented by 3D geometries and immersed into the grid as introduced in chapter 6. For the MV, the Linköping valve is assumed while for the aortic valve the 3D planar valve model is used.

For a simplified comparison, as well as the assessment of the effective volume which is not constrained by prescription of the wall velocity, a mapping field κ, which can take three discrete values and is also referred to as denoting the inside cells, is introduced:

$$\kappa = \begin{cases} 1 & \text{cell inside the prescribed wall and } \Phi = 0 \\ 0.5 & \text{cell inside the prescribed wall and } \Phi > 0 \\ 0 & \text{else} \end{cases} \tag{10.2}$$

Thus, all cells with $\kappa = 1$ contribute to the effective volume, while the cells with $\kappa = 0.5$ have the velocity of the prescribed wall. If a threshold filter is applied to hide all cells with a value of $\kappa > 0.5$, the remaining cells correspond to the compared body-fitted implementation.

To assess flow patterns as well as to evaluate the fraction of the initial blood volume that remains in the LV, a passive scalar transport equation is implemented as will be described in section 14.2.7.2. The scalar ψ is considered as a measure of local concentration of blood. It is initialized with a value of $\psi(x,t_0) = 1$ for all cells located in the LV.

10.2 Results

In the first subsection, the resulting geometries, as well as the scalar fields necessary to compute the wall movement (distance Λ, inside κ and the indicator field Φ) are shown. In the two following subsections, the comparison between the IBM solver results and the existing body-fitted mesh approach is carried out.

10.2.1 Resulting geometry

In figure 10.1, the cubic computational grid is visualized. On the right hand side of the figure, the endocardial surface in the diastolic state is shown in red color. It determines the maximum LV size and thus forms the outer limit for refinement. The smallest LV volume results during systole, which is the point in the cardiac cycle that corresponds to the green endocardial surface. All cells around and in between these two layers are refined by a factor of 2, resulting in 64 cells per original cell. The MV position is visualized in orange, while the aortic valve is colored purple. The region around both valves is refined with a factor of 3, resulting in 512 cells per initial cell, which corresponds to an average edge length of approximately 0.75 mm. The cell edges shown in dark blue color show the refined cells. All in all, the model comprises 741,345 volume elements.

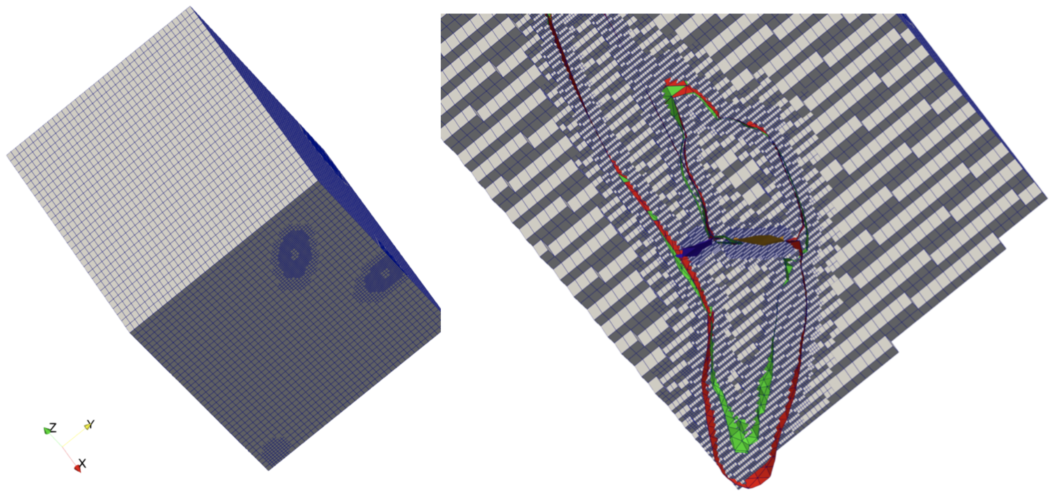


Figure 10.1: Grid for the IBM simulations of the left heart side. Left: 3D cubic grid with the edges visualized in dark blue color. Right: Cut in long axis direction through the 3D grid. The cell edges visualize the refined areas, while in red and green color the surfaces used for the refinement are depicted. The prescribed endocardial surface in the diastolic state (red) denotes the maximal size of the LV, while the systolic state (green) results in the smallest volume. The MV position is illustrated in orange, while the aortic valve is colored purple.

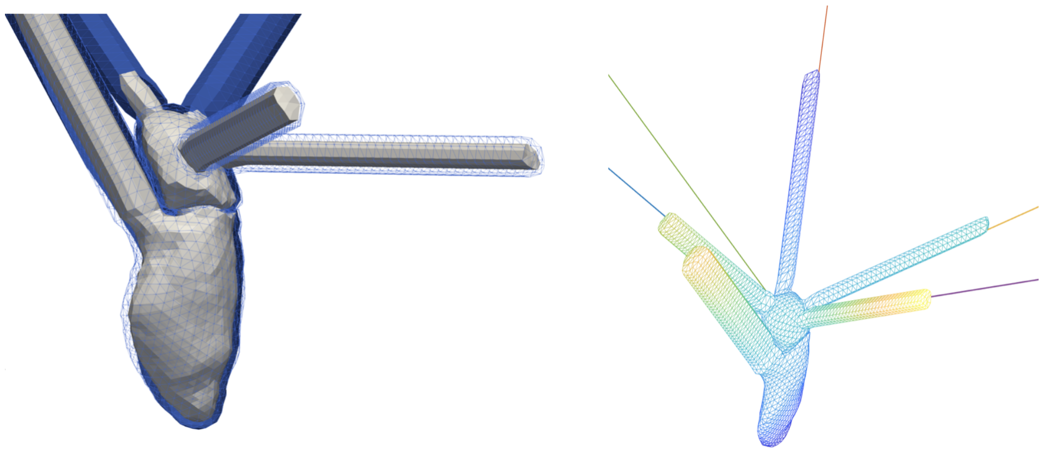


Figure 10.2: Prescribed endocardial geometries which are immersed in the cubic grid. Left: Results of the inflation of the left ventricular geometry. While the initial geometry is shown in light gray color, the inflated geometry is shown in dark blue (in the front part as a wireframe, in the back in surface representation). Right: The inlet and outlet normal vectors are depicted by lines through the origin of the grid.

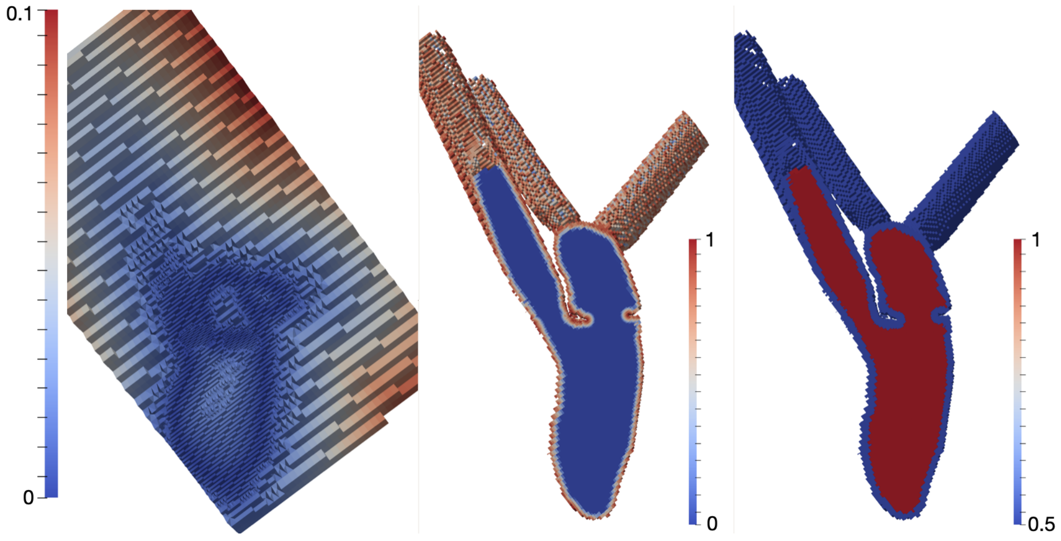


Figure 10.3: Resulting scalar fields for the first time step ($t = 0.01$ s): On the left hand side the distance field Λ in m is mapped on the grid that is cut in long axis direction. On the right hand side, the inside field κ is plotted for all cells with $\kappa > 0$. In addition, the geometry is clipped in long axis direction to distinguish the wall cells from the effective volume. In the center, the indicator field is plotted for all cells that fulfill the condition $\kappa > 0$.

Figure 10.2 shows the original (light gray color), as well as the resulting inflated endocardial surface geometry for the wall prescription. On the right hand side of the figure, the resulting normal vectors for the inlet and outlet surfaces are depicted.

Figure 10.3 provides an overview of the resulting scalar fields which are relevant to model the moving boundary wall. On the left hand side, the distance field Λ denotes the ED of each cell center to the closest point or vertex in the constraining endocardial surface. On the right hand side, the introduced scalar field κ marks all cells enclosed by the prescribed wall. While all cells with a sufficient distance to the wall contribute to the effective volume and are not constrained by any velocity prescription, they are marked with an inside value of $\kappa = 1$. Cells that have a prescribed wall velocity are marked with $\kappa = 0.5$. By applying a filter that excludes all cells outside the moving wall ($\kappa = 0$), the shape similar to the body-fitted approach is extracted. In the center panel, the indicator field deduced from the distance field is plotted. As expected, it shows a decay from the outside ($\Phi = 1$) towards the inside ($\Phi = 0$) of the geometry.

10.2.2 Geometry comparison

In this section, a comparison between the resulting geometries is included.

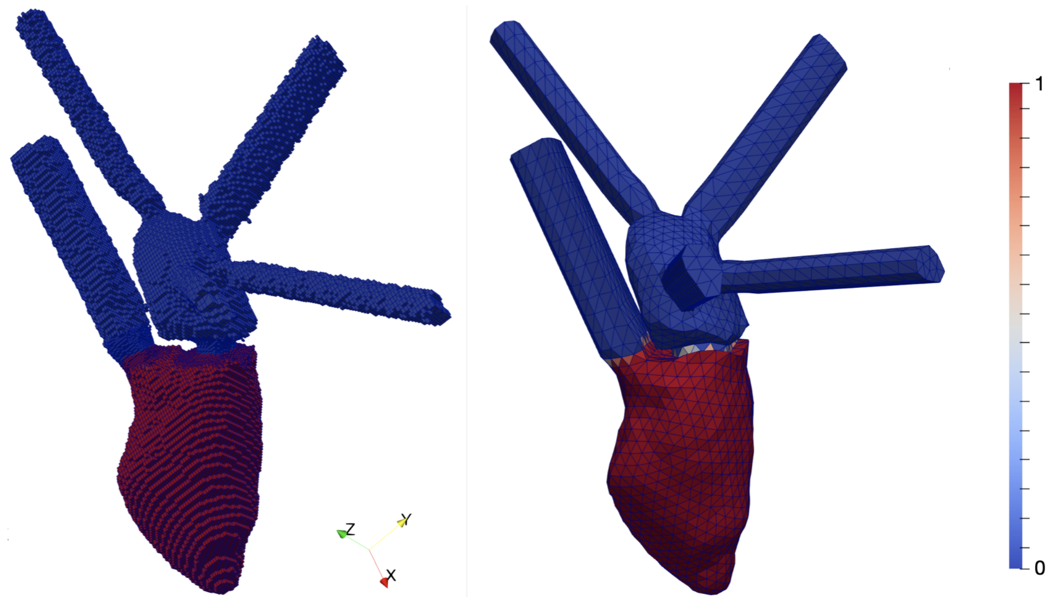


Figure 10.4: Comparison of the resulting IBM geometry, filtered by $\kappa > 0.5$ (right), to the body-fitted mesh formulation. Color-coded is the initial distribution ($t = 0.01$ s) of the scalar field ψ, which marks all cells located in the LV with $\psi = 1$.

In figure 10.4 the filtered ($\kappa > 0$) IBM mesh is visualized on the left hand side. It visually corresponds to the body-fitted mesh on the right hand side. A small deviation can be seen in the pulmonary vein on the left (closest to the aortic trunk): As the scalar fields are computed in the first time step, the geometry visualized here corresponds to the first exported time step

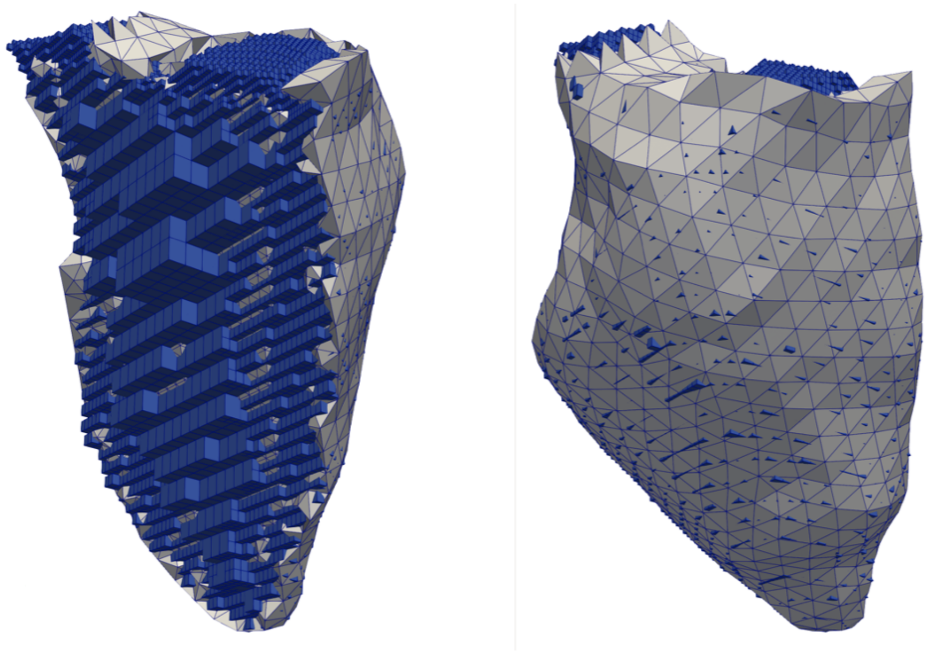


Figure 10.5: Comparison of the resulting LV geometries cut along the long axis, in front and back perspective. The IBM result which is shown in blue color is compared to the gray body-fitted cells.

at $t = 0.01$ s. The color coded initialization value for the scalar field ψ, as well as the shape of the ventricular chamber also look concordant.

However, a more detailed view of the LV is shown in figure 10.5, which offers a long axis cut through the two superposed ventricular chambers from the front and from the back view. The IBM geometry is visualized in blue, while the body-fitted mesh is colored in light gray. While an overall accordance between both geometries can be confirmed, many blue edges penetrate the gray surface. As the smooth body fitted shape is approximated by the cubic cells in the IBM approach, this behavior is intended. However, considering that the volume should fully be approximated by the cubic cells, the penetration should even be higher. In addition, in the region around the valves, the blue cubic cells are not included up to the edges of the endocardial walls.

An integration of the volume of all cell located in the LV confirms this observation: In the diastolic state, the IBM models yields a volume of 165 ml in comparison to 184 ml of the body-fitted mesh. This offset remains more or less constant and results in volumes of 125 ml and 144 ml respectively in the systolic state.

10.2.3 Fluid quantities comparison

Finally, the fluid quantities comprising the velocity u, the pressure p and the flow patterns based on the scalar field ψ are evaluated. To simplify the comparison, the body-fitted data is always plotted on the left hand side of each figure, while the results obtained based on the IBM model are represented on the right.

Figure 10.6 visualizes the flow patterns computed based on the passive scalar transport equation. In both geometries, a clear jet towards the apex can be observed with a clear focus on washing out the upper two thirds of the chamber. While in the body fitted mesh, a circular vortex forms in the center of the chamber, a considerable fraction of initial blood is retained in this region. In the IBM mesh, a more straight jet towards the apex can be noticed, whereas it does not penetrate as deep towards the apex as in the body-fitted case. Due to the movement of the wall, in the IBM case the fluid close to the wall is not in complete rest any more. In the aortic trunk, the resulting laminar flow profile can be seen quite well in the IBM geometry.

Similar flow patterns are obtained during the systolic state of the second heart beat, which is shown in figure 10.7. While in both scenarios, a major part of the upper chamber volume is ejected, in the body-fitted case a vortex-like structure causes a region with a low washout in the upper center part of the LV. In the IBM case, the upper part of the volume that initially started in the LV is ejected completely. For both scenarios, already in the second cycle a high fraction of the ejected volume has a value of $\psi < 50\,\%$. This confirms a high rate of the direct flow component, which quantifies the amount of blood that enters and leaves the LV in the current heart beat.

The explained behavior can also be observed in the time course of the residual volume in figure 10.8: While in the IBM scenario a high fraction of the initial blood volume leaves the LV in the first beat, this fraction decays in the following cycles. In the end, a significantly higher amount of residual volume remains in the apex region. On the contrary, in the body-fitted approach the ventricle is further washed out each cycle with only a small V_{res} remaining after four heart beats. The increase in residual volume is not caused by a regurgitated flow, but by a change in the volume of the considered cells due to the valve plane movement.

The resulting velocity fields for both implementations are visualized in figure 10.9 during diastole and in figure 10.10 for the systolic state. The color bar is the same for the two compared mesh implementations to enable a comparison. In the diastolic state, the IBM approach shows the maximum velocity in the MV region. During the passive filling of the LV, due to the small opening orifice of the Linköping valve discussed in section 6.3, a high velocity occurs. In comparison, in the body fitted mesh a more homogeneous distribution of the flow can be visualized with only an increased velocity at the edge of the MV region. In the IBM case, also velocities in the same order of magnitude occur in the LA and LV. However, the magnitude is increasing towards the valve and not as homogeneous as in the body-fitted case.

The same phenomenon can be observed in the systolic state. Again, the maximum velocity in the IBM scenario occurs in the valve region due to the small valve diameter. The laminar flow profile in the aortic trunk can be observed again. In the body-fitted mesh, the highest velocities occur along the center line of the aortic trunk. Velocities magnitudes around the half of the maximum value occur all over the ventricle, while in the IBM case this is only the case for the upper part.

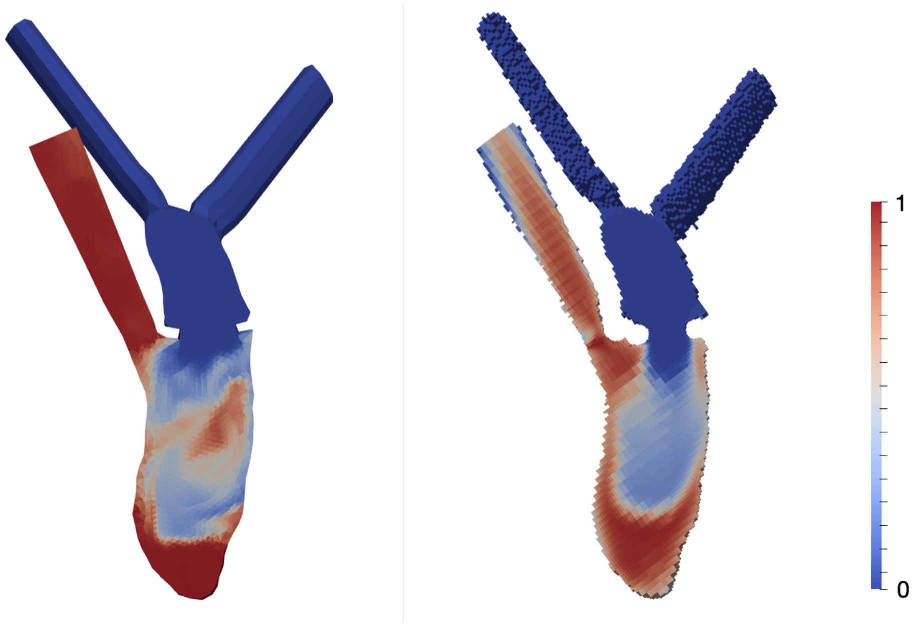


Figure 10.6: Scalar field ψ visualizing the flow patterns in the LV depicted in the diastolic state after the first heart cycle ($t = 1.35\,\mathrm{s}$).

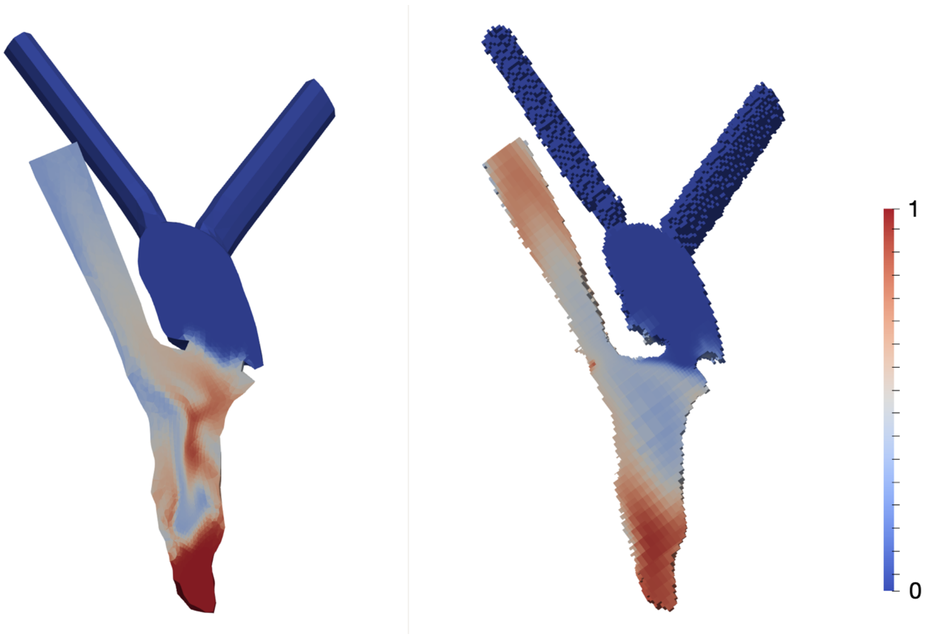


Figure 10.7: Scalar field ψ visualizing the flow patterns in the LV depicted in the systolic state of the second heart cycle ($t = 1.85\,\mathrm{s}$).

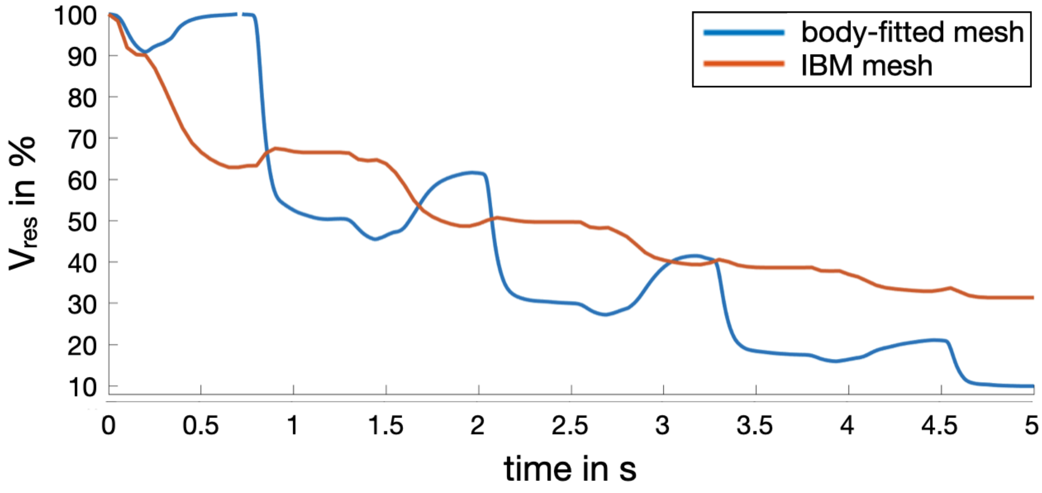


Figure 10.8: Residual volume V_{res} in % plotted over the time for four consecutive heart beats. Comparison of the body-fitted (blue color) and the IBM approach (red).

Comparing the pressure in the two different scenarios delivers the most interesting insight of this comparison. On the first glance, the pressure distribution is similar, as the pressure in the aortic trunks and the ventricular chamber are the same. After a pressure drop over the MV, the pressure in the LA is lower. However, this visual correspondence is only achieved by an adaptation of the color bar. This reveals a shortcoming of the immersed representation of the valve leaflets: As the velocity of the valve is prescribed by an external forcing term of the Navier Stokes equation (NSE) (compare equation 3.8), the areas before and after the valve are decoupled. Thus the pressure drop that occurs across the valve can not be resembled.

Considering the computational cost, the body-fitted mesh consumed a wall-clock time of 114,546 s = 31.8 h ≈ 1.5 d for the simulation of four full heart cycles on a single core processor (i5 processor, 3.7 GHz). In comparison, with 117,701 s = 32.7 h, the IBM approach needed roughly the same computational time without any optimization of the code yet. However, the IBM implementation offers the great benefit that it can be solved on the cluster machines in parallel mode. Using an amount of 64 parallel processors, the four heart cycles were simulated in approximately 5.5 h including decomposition and reconstruction of the mesh.

10.3 Discussion

The application of the computational fluid dynamics (CFD) solver based on the IBM approach showed a good basic functionality to compute the fluid dynamics in the human heart. However, taking into account the variation in the absolute values of the pressure, the

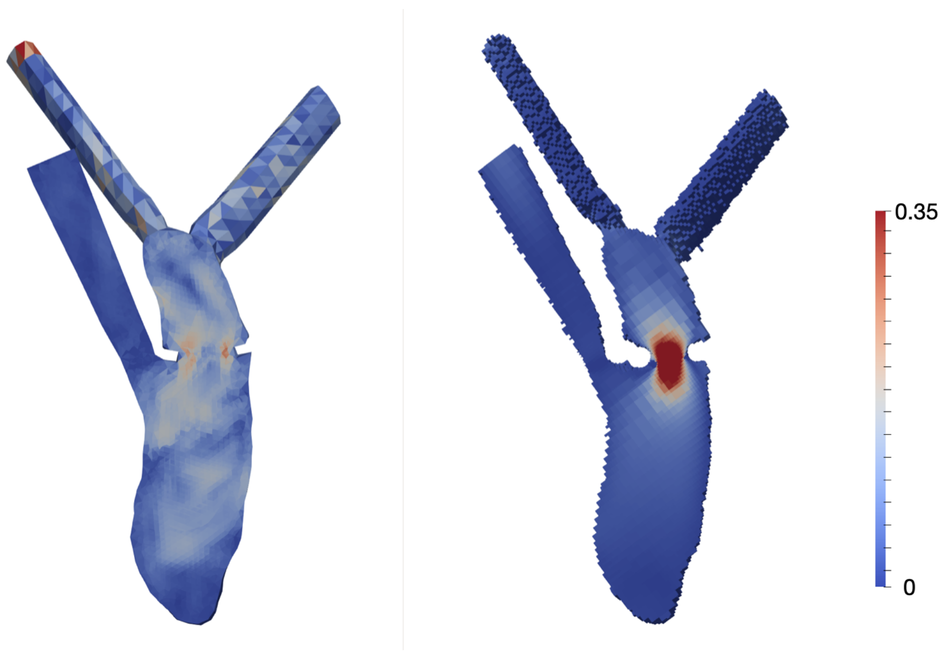


Figure 10.9: Magnitude of the velocity field u in m s^{-1} in a long axis cut of the left heart depicted in the diastolic state after the first heart cycle ($t = 1.35$ s).

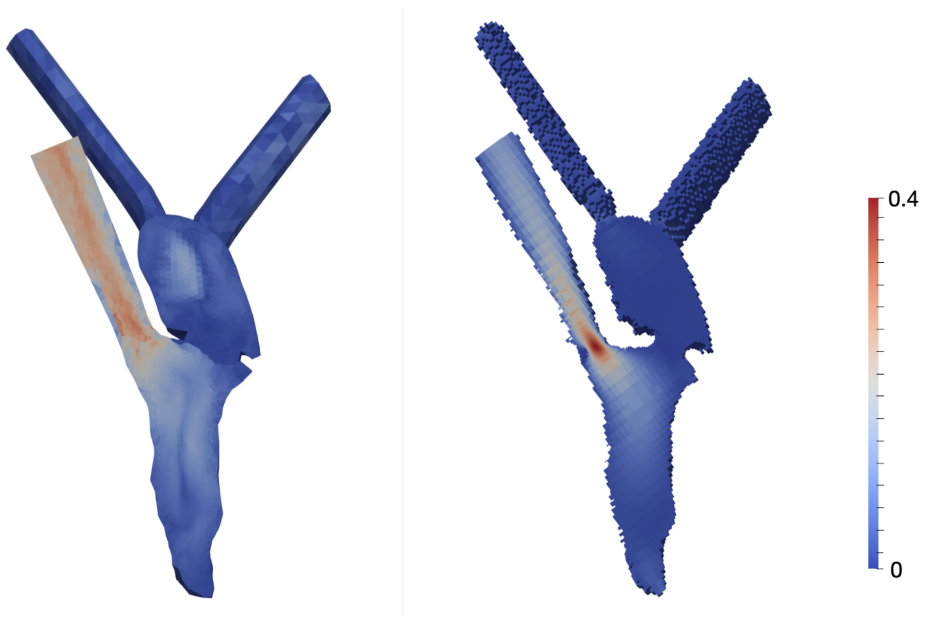


Figure 10.10: Magnitude of the velocity field u in ms^{-1} in a long axis cut of the left heart depicted in the systolic state of the second heart cycle ($t = 1.85$ s).

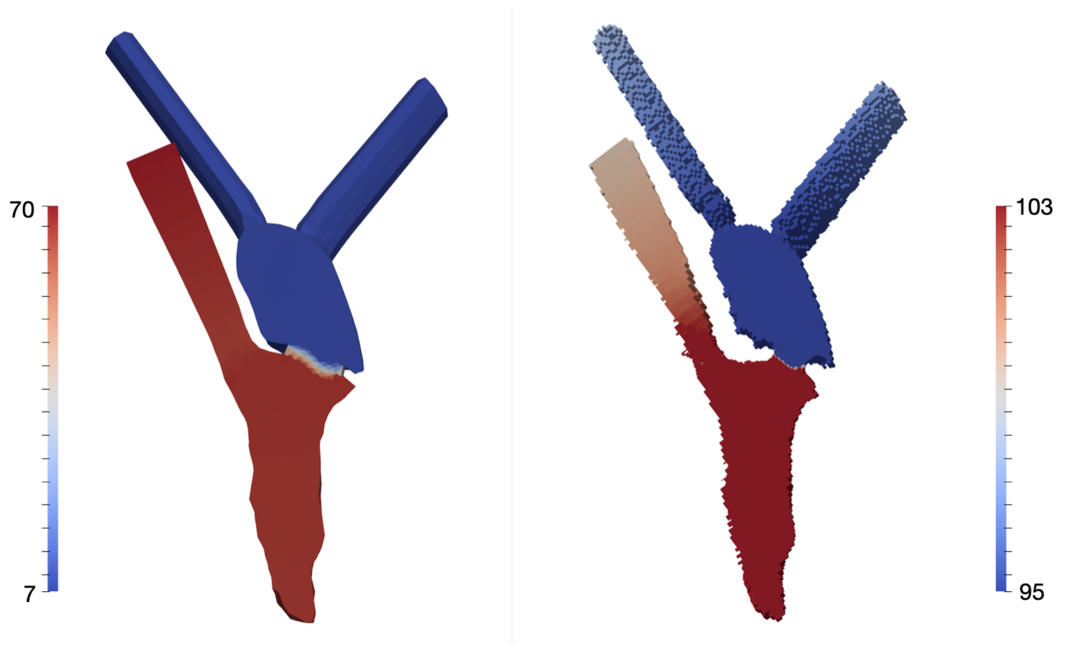


Figure 10.11: Pressure field p in mmHg mapped on the clipped left heart geometry. Systolic state of the second heart cycle ($t = 1.85$ s).

simulations revealed the high influence of the choice of the BC. With the use of the 3D resolved valves immersed into the computational grid, the pressure drop that was modeled by the porosity of the porous zone approach is not considered any more. To obtain accordant absolute pressures, this pressure drop has to be modeled again. One option would be the external prescription of the absolute pressure values in the proximity of the valves. However, this would again have a high influence on the flow patterns, impose an additional quantity that has to be deduced from the circulatory system results in correspondence with the valve diameter and thus increase the algorithm vulnerability. On the other hand, an alternative would be to additionally prescribe the pressure at the inlet surfaces. As at least one valve is always closed throughout the cardiac cycle, this would not over-constrain the simulation but allow for a prescription of the pressure at both sides of the currently closed wall. However, this again highlights one limitation of the light coupling strategy. While such influences are considered in every single time step in a strongly coupled approach, in unidirectionally coupled solvers all data has to be prescribed by the mechanical simulation results and the circulatory system data beforehand. The missing exchange of geometrical deformation in a bidirectionally manner imposes the main drawback of light coupling approaches. Due to the choice of the valve implementation, a higher number of parameters have to be prescribed in unidirectionally light coupling strategies.

The representation of the valve geometry showed to have a big impact on the flow patterns in the ventricular chamber. Even if this could have been expected, it has to be taken into account when evaluating the results. Especially the flow patterns and the location of local velocity maxima are influenced. Nevertheless, the comparison to the body-fitted approach revealed a basic accordance.

In the context of the determination of the geometry, a reduced LV volume occurred due to the approximation of the smooth wall by cubic cells in the wall region. This offset in the resulting volume has to be considered. While it can be corrected by a further inflation of the geometry, it needs to be ensured that the space between the prolonged trunks is big enough and does not lead to intersections. Alternatively, the wall thickness can be further decreased, resulting in an increase of the effective volume of the LV. To prevent from introducing holes in the wall by this process, the wall cells could be refined an additional stage.

All in all, the simulations based on the IBM implementation showed promising results, revealing limitations of the algorithm that can be overcome. The prescription of the moving wall geometry turned out to work quite well: Even if the prolonged trunks are close to one another, while one of them simulated inflow dynamics and the other one simulated an outflow, they did not interfere.

PART IV

ASPECTS OF VALIDATION

Chapter 11

Insights from measured data

11.1 Motivation and aim

The initial aim of this sub project was to close the loop of implementing, verifying and validating a fluid dynamics solver based on the immersed boundary method (IBM) approach. Based on the measured endocardial deformation, the necessary boundary conditions can be extracted and flow boundary conditions can be prescribed at the inlet and outlet surfaces. After simulating the results based on the measured conditions, a direct quantitative comparison between the two flow fields was intended. As 4D flow magnetic resonance imaging (MRI) is the method of choice in this context, a set of MRI measurement data was acquired from a healthy volunteer. However, while the flow measurement data was not captured with a sufficient resolution, the segmentation of the correct planes was challenging. The evaluation of a 3D resolved velocity vector field was not possible. In addition, the long axis slice of the four chamber view crops the left atrium (LA) centrally. Therefore, even a manual segmentation of the whole left side of the heart was not possible. Due to these circumstances, only a 3D geometry segmentation of the left ventricle (LV) as well as the path lines of the flow in a cut plane through the LV were assessed for a qualitative comparison between in-silico and in-vivo data.

11.2 Flow measurement data

In the context of this project a complete MRI measurement data set was recorded at the Theresienkrankenhaus in Mannheim. Measurements comprise electrophysiology, mechanical and fluid dynamic data and were taken from a 29-year-old male healthy volunteer. Informed consent was provided. An overview on the most important global quantities is given in table 11.1, whereas the resulting mechanical movement of the endocardial surfaces (cine images of the 2 chamber view, as well as short axis slices) for some characteristic points in the cardiac cycle is attached in appendix C.

Heart rate	LV EDV	LV ESV	LV EF	Blood pressure (arm)
62.5 bpm	186.4 ml	83.1 ml	55.4 %	120/70 mmHg

Table 11.1: Global measured quantities.

The complete measurement data set comprises the following recordings: For global alignment and organ positions, a static whole torso image is acquired during diastole. Static whole heart slices are acquired during diastole in short and long axis view to create the 3D mechanical model of the heart. To capture the endocardial surface movement throughout the cardiac cycle, cine-movies are recorded in 2-, 3- and 4-chamber view. For further mechanical validation, strain evaluation can be deduced from these images. 4D-flow measurements throughout the whole heart were acquired during a full heart beat for fluid solver validation.

To finalize the electro-mechanic-fluid data set, a full 12-lead-ECG signal was recorded additionally for several minutes in rest and in exercise condition.

Evaluation of clinical data was done in cvi42 software (Circle Cardiovascular Imaging, Calgary, Canada, Release 5.14.2) mainly relying on the modules for functional analysis (short and long axis based), the strain and the 4D flow module. Automatic segmentation based on the included artificial intelligence algorithm was applied to receive a 3D geometry of the LV (endocardial and epicardial) and the right ventricle (RV) over time.

11.3 Methods

To segment all relevant surface geometries, the artificial intelligence engine included in the clinical evaluation tool was applied.

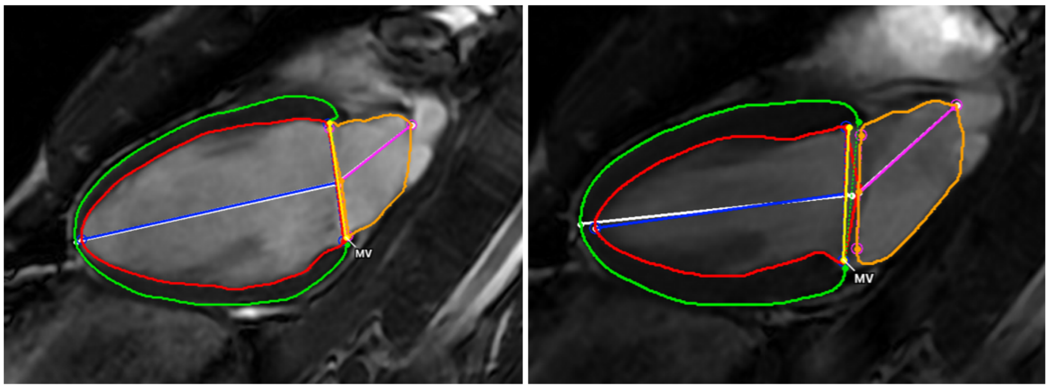


Figure 11.1: Automatic segmentation of chamber contours for the left heart side. The endocardial LV surface is colored in red, while the epicardial surface is marked in green. The orange color denotes the shape of the LA, the yellow line denotes the position of the mitral valve (MV). Left: diastolic state. Right: systolic state.

Based on the available measurement data set, a time series of the LV geometry is extracted. It can be used for a visual comparison of the shape variation throughout the cardiac cycle and delivers a basis for future model creation from measurement data.

Finally, the flow in a long axis cut of the whole heart geometry is assessed to visually compare the resulting pathlines. Based on these visualizations, at least a qualitative comparison to the acquired measurement data is enabled.

11.4 Results

The clinical evaluation tool reliably determined the LV endo- and epicardial surfaces in most of the slices and time steps. An exemplary screenshot for the segmented diastolic and systolic geometry is included in figure 11.1.

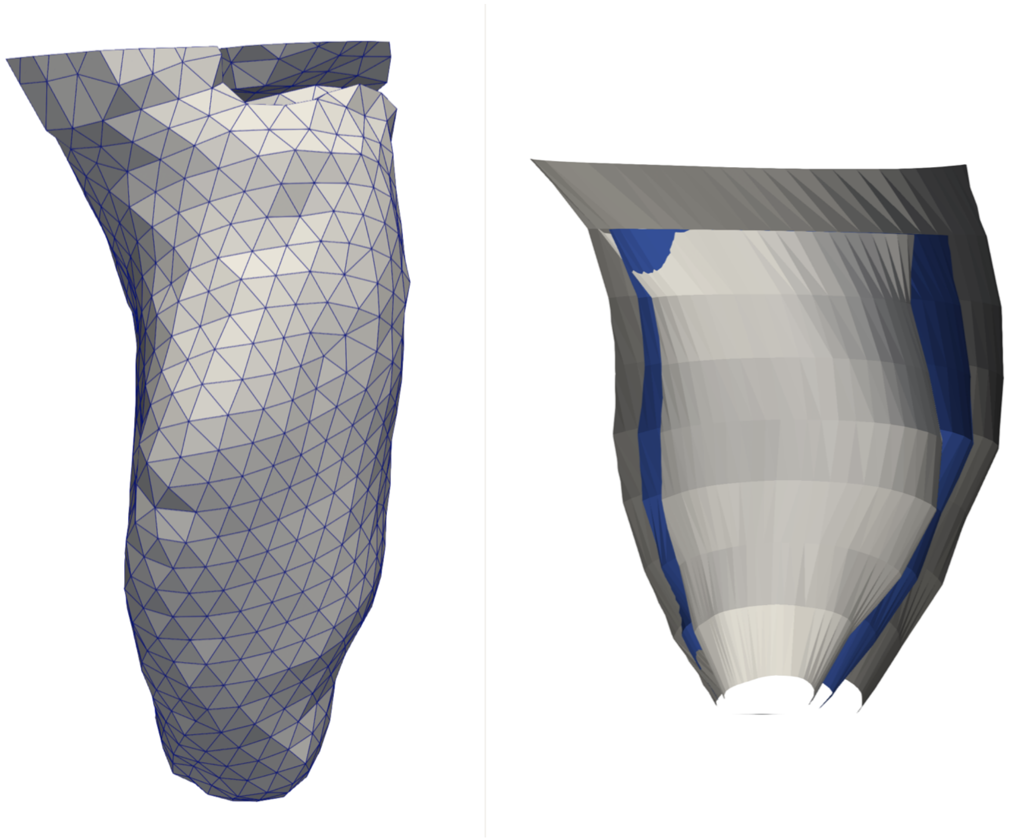


Figure 11.2: Right: Surface files representing the geometry of the LV segmented from measured data. While in the diastolic state, the maximum LV volume results in the largest endocardial surface, during the systolic state the lowest volume results. The blue surface models the endocardial surface for a point in time between the two extremes. On the left hand side, the left endocardial wall of the LV of the geometry used for all simulations is shown for comparison.

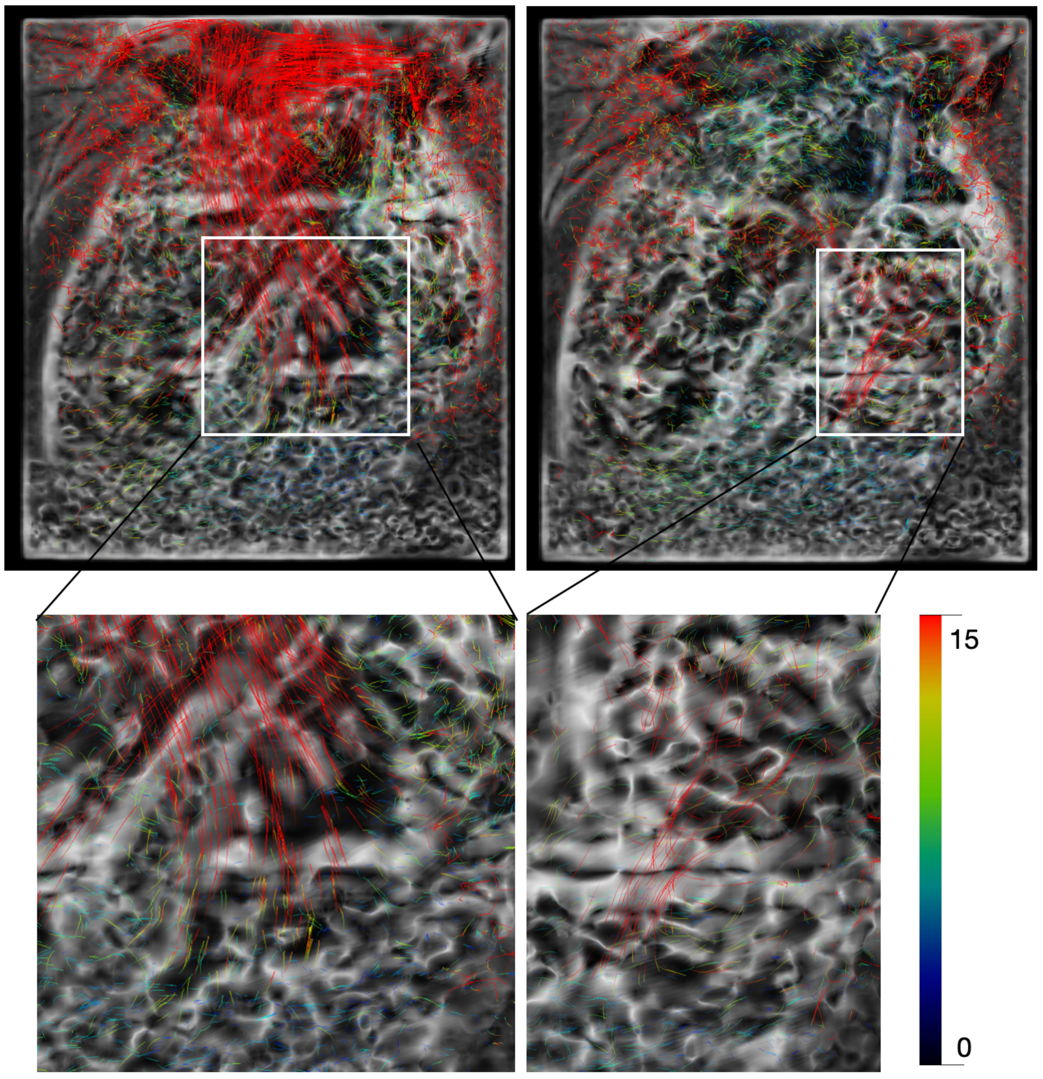


Figure 11.3: Pathline representation indicating the velocity field in ms^{-1} for a long axis cut trough the whole heart for different points in time. Left: Ejection of ventricular blood in the aortic trunks through the systole. Right: Flow across the mitral valve (MV) during the relaxation of the LV.

In addition to the endocardial surfaces, the long axis height of the ventricle is evaluated to denote the longitudinal strain, which results in -16 %.

The time series of the LV endocardial surfaces in included in figure 11.2. While diastolic and systolic geometry yield the highest, respectively lowest LV volume, a time step in between is shown in addition. Compared to the in silico geometry used for the simulations so far, a less high, but more wide geometry results from the segmentation. However, the bend structure that forms a smooth transition into the aortic trunk region is clearly visible. Considering the time course, the dilatation towards the right hand side of the image is higher.

This also makes sense, as on the other side, the right ventricle is attached and thus constrains the motion.

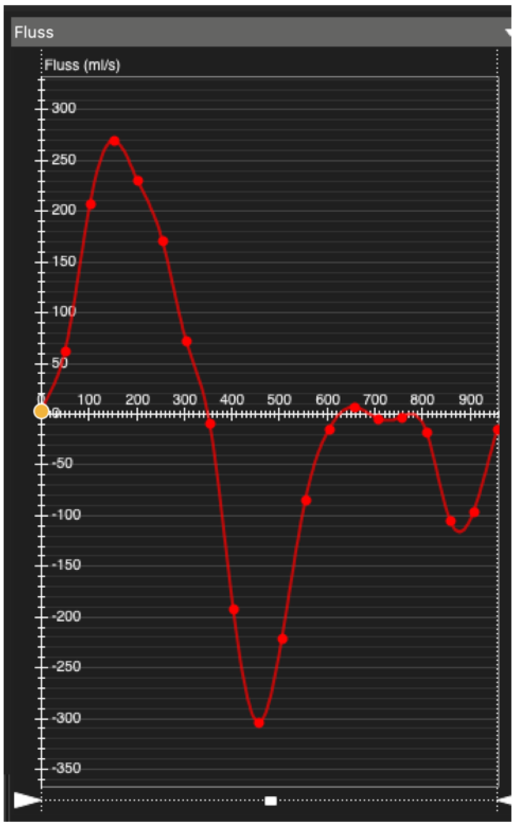


Figure 11.4: Time course of the flow trough a cut plane in ml s^{-1} comprising aortic flow (first positive wave) and mitral flow (first negative wave).

The most interesting insights were delivered by the pathline representation of the blood flow velocity in a long axis cut of the whole heart. In the systolic state visualized in the left part of the figure, the inner volume of the aortic trunk can clearly be assessed. In addition, the close up view reveals a smooth and bent transition of the blood leaving the ventricular chamber towards the aortic valve. This corresponds quite well to the velocity fields that have been observed in the simulations deduced in the projects of this thesis, exemplary visualized in the bottom row of figure 10.10. However, to a great extent, this resulting velocity field is already indicated by the shape of the geometry. In the diastolic state, the flow across the mitral valve (MV) can be assessed. The pathlines depicting the flow enable to deduce the diameter of the opened valve orifice area. In the close up of the valve zone, first a straight jet through the MV can be observed which after a few centimeters drifts apart towards the lateral direction. However, as the flow velocity decays after this point, it can not longer be distinguished from the omnidirectional background flow. Thus the screenshot does not allow

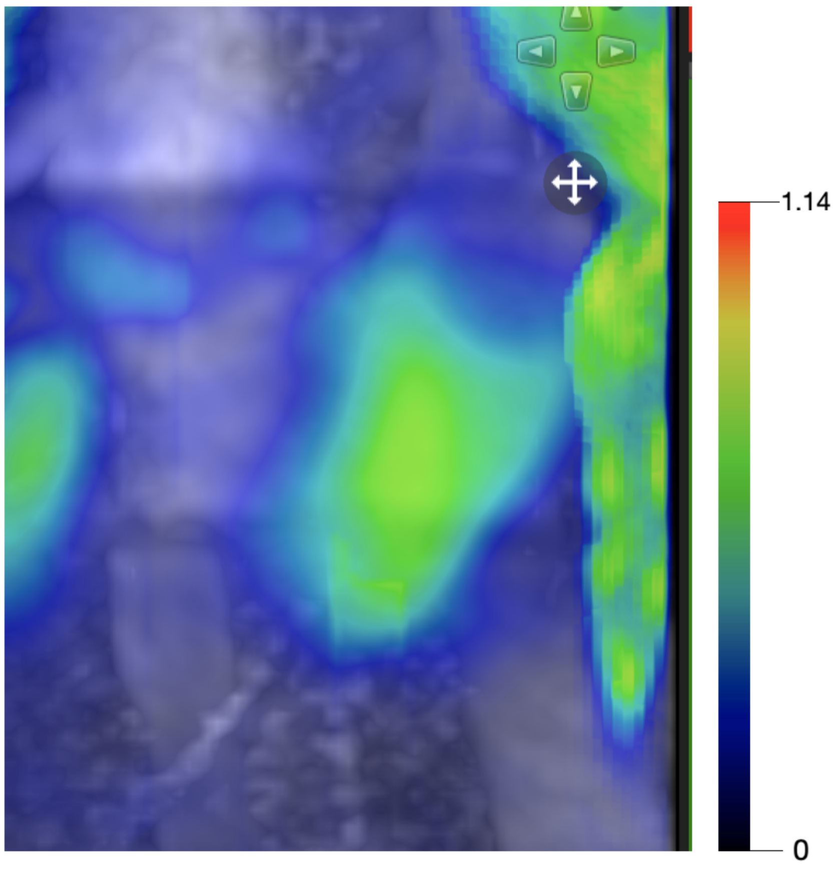


Figure 11.5: Wall shear stress in Pa for a long axis cut through the whole heart. The central green area reveals the shape of the LV.

for the evaluation of the penetration depth towards the apex. However, assessing the relevant valve diameters offers a possibility of adapting the valve parameters. In the systolic phase, the high density of pathlines in the aortic trunk denotes a high flow rate.

Figure 11.4 provides a temporal statistics of the flow rate in a specific plane located perpendicular to the long axis view in figure 11.3. However, in this plane as well the aortic trunk resulting in the first positive wave in the figure, as well as the mitral valve resulting in the first negative wave, are included. As the time course starts with the aortic ejection, the time course in the measurement data does not start from the diastatic state. However, this is due to the system being triggered on the r-peak of the electrocardiogram (ECG). The second negative wave therefore is caused by the atrial kick.

Finally, the wall shear stress analysis is shown in figure 11.5 comprising the same long axis view as previously applied. With an increased wall shear stress rate, the geometry of the LV chamber can be assessed.

11.5 Discussion

Summing up, the assessment of acquired measurement data delivered first promising insights in the validation based on 4D flow MRI data. A basic accordance with the qualitative data can be confirmed. However, as up to now only qualitative comparisons were deduced, the explanatory power of this statement is bounded. In future studies relying on a sufficient resolution of the flow data and a correct acquisition of the short and long axis geometrical data, a more thorough statement can be deduced. The basic functionality of the method for the application in hemodynamic evaluations was confirmed in [135].

In addition, parameters like the MV plane displacement or the segmentation of the blood volume can be assessed. Acquiring a full multi-physics data set would still deliver additional insights and also allow for a validation of the coupling procedures.

PART V

FLUID DYNAMICS IN THE DISEASED HEART

Chapter 12

Mitral valve regurgitation

In a first application scenario, an in silico model is utilized to compute the fluid dynamics in a heart suffering from mitral valve regurgitation (MVR). Three characteristic severity stages are artificially introduced to a mitral valve model to systematically evaluate influences with focus on vortex formation and ventricular washout.

The content of this chapter is adapted from the conference publication [39] under the Creative Commons Attribution 4.0 (CC BY 4.0) international license. Results are also based on the student theses [136, 137].

12.1 Introduction

MVR is of increasing prevalence and currently the second-most frequent valvular heart disease in Europe [82, 138]. As explained in detail in section 2.4.1, this phenomenon is caused by a diseased mitral valve (MV) that is not closing properly. Thus, blood regurgitates from the left ventricular chamber into the left atrium (LA). Using the implemented fluid solver based on the Finite Volume (FV) method in combination with an adaptation of the valve model, altered blood flow in the left heart can be simulated to assess qualitative and quantitative measures non-invasively. In this context, the visualization of different flow patterns and simulation-based blood flow component analysis is a powerful tool to reveal disease-induced flow alterations. These are clinically relevant because the medical doctor has to decide at what stage of the disease a surgery is mandatory.

So in this chapter, first an in silico fluid dynamics simulation is executed to simulate the time course of the retrograde jet throughout the heart cycle. With a systematic investigation of the the impact of different MVR severities, the qualitative impact on ventricular flow patterns, as well as the quantitative impact on regurgitation fraction, wash-out and flow volumes are revealed. Secondly, three different positions of the valvular leakage position are assessed to investigate the deviations between the central and two eccentric retrograde jets. In addition, alterations of the different blood flow components are quantified in this chapter. A visualization tool based on a particle tracking algorithm is provided to quantitatively and visually evaluate altered flow patterns.

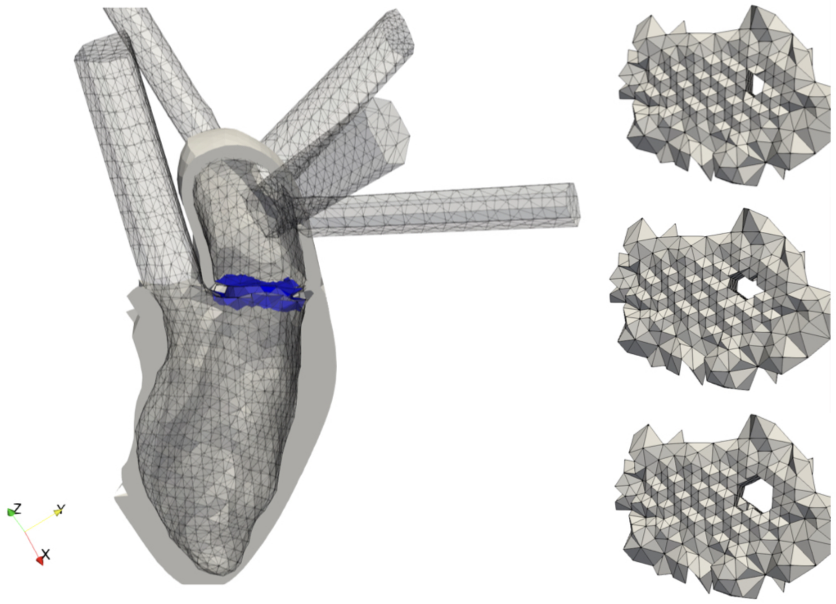


Figure 12.1: Left: Finite element model of the left heart. Mitral valve zone marked in blue. Right: Porous zones that model mitral regurgitation with severity mild, moderate and severe from top to bottom. Figure adapted from [39], published under CC BY 4.0 license.

12.2 Methods

12.2.1 Basic model generation

The basis of this model is a patient-specific healthy heart geometry represented by a body-fitted mesh that was created from magnetic resonance imaging (MRI) scans and discretized using finite volume techniques. Focusing on the mitral valve, the model comprises the LA and left ventricle (LV) as well as the prolonged trunks (compare section 7), as visualized in the left part of figure 12.1.

Unidirectional coupling of cardiac continuum mechanics to the fluid dynamics solver is established by considering the endocardial surface movement obtained from the mechanical simulation framework [120] and the pressures resulting from the closed-loop circulatory system model [25] as boundary conditions in the fluid solver. The Arbitrary Lagrangian Eulerian (ALE) formulation is used [139]. To resemble flow characteristics, all heart valves are represented by a planar model based on a porous material law [38]. For an idealized heart geometry, Daub et al. [38] already showed, that the planar MV model was able to reproduce flow fields that show physiological behavior. The Darcy-Forchheimer law [140] is used to quantify the pressure drop across the MV by adjusting the permeability of the porous zone. Extending this approach, the diseased valve is modeled based on the porous zone

MVR severity	**mild**	**moderate**	**severe**
EROA in mm^2	<20	20-39	≥ 40
Regurgitation volume in ml	<30	30-59	≥ 60

Table 12.1: MVR severity according to European Society of Cardiology guidelines [82]. Classification based on effective regurgitation orifice area (EROA) and regurgitation volume (V_{re}).

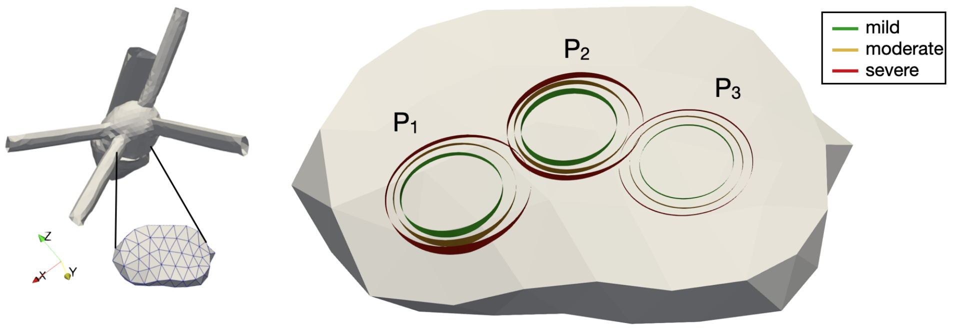


Figure 12.2: Geometry of the porous zone modeling the MV. Position of circular holes, which model the regurgitation orifice area. Different colors highlight the three severity stages: Mild MVR in green, moderate in orange and severe MVR colored in red. P2 denotes the central position, while the location of the eccentric nodes is labeled P1 (left) and P3 (right). LV geometry for orientation on the left.

implementation in this study. The variation of this zone's permeability over time models the valve characteristics in a compact way and is controlled depending on flow.

By solving the Navier Stokes equation (NSE), the pressure distribution and the velocity field across the left part of the heart geometry are computed. Blood is modeled as an incompressible Newtonian fluid, with density $\rho = 1055\,\frac{kg}{m^{-3}}$ and dynamic viscosity $\nu = 0.004\,\frac{kg}{m \cdot s}$.

12.2.2 Disease model

The starting point of the disease model is the planar MV model of a healthy heart geometry. Its outer surface is applied as the boundary of the porous zone implementation in the FV method that was described in section 6. To resemble the pathology, a circular hole is added in medial direction. By including holes of different diameters in the posterior leaflet of the MV, mitral regurgitation with different severities is modeled by different effective regurgitation orifice areas according to the European Society of Cardiology guidelines for primary MVR [82]. The hole diameter for mild, moderate and severe MVR and the corresponding regurgitation volumes can be found in Table 12.1. While for mild and severe MVR the referenced limits are chosen, for moderate MVR the mean value of $EROA_{moderate} = 30\,mm^2$ is chosen.

To additionally evaluate the influence of different hole positions, three different locations are investigated. Besides the central hole (P2), two eccentric positions in the left (P1) and right (P2) side of the anterior leaflet are simulated. The three locations combined with the three severity stages are shown in figure 12.2.

12.2.3 Scalar transport evaluation

To evaluate the fraction of the initial blood volume that remains in the LV, a passive scalar transport equation is implemented as will be described in section 14.2.7.2. The scalar ψ is considered as a measure of local concentration of blood (e.g. in the LA), that initially started in the LV and is calculated following equation 14.7. Initialization is the same as in figure 14.2 with the initial value $\psi(x,t_0) = 1$ for all cells located in the LV and $\psi(x,t_0) = 0$ for all others.

12.2.4 Blood flow components analysis

Introduced by Bolger et al. [141], the separation of blood flow in four different flow components is a common method to quantify component fractions and to visualize trajectories. The blood volume starting in the LA is composed of the direct flow (DF) that directly leaves the LV in the same cycle and the retained inflow (RI) that remains in the LV during the concerning cycle. The blood component that is already located in the LV at the beginning of the heart cycle and then ejected through the aorta is named delayed ejected flow (DEF), whereas the volume remaining in the LV is the residual volume V_{res}. Considering the back flow (BF) that regurgitates through the MV as a fifth component, the visualization of the components can be seen in figure 12.7.

The flow components are mathematically computed based on the volume relations at the end of the systole t_{ES}, at the end of the diastole t_{ED} and at the initial point in time t_0. Therefore, the definition of two auxiliary quantities is the most descriptive way: At first, the blood that is located inside the LV at the begin of the evaluation ($t = t_0$) is labeled V_v. This quantity changes over time and is computed by integration:

$$V_{\mathrm{v}}(t) = \int\int\int_{V_{\mathrm{endo}}} \Psi(x,t)\, dV \; . \tag{12.1}$$

The same way, blood that is located outside the LV at the begin of the evaluation is labeled V_o.

$$V_{\mathrm{o}}(t) = \int\int\int_{V_{\mathrm{endo}}} (1-\Psi(x,t))dV \tag{12.2}$$

With these two volumes, the blood volumes of the flow components can be computed as following:

$$\begin{aligned} V_{\text{rev}} &= V_{\text{v}}(t_{\text{ES}}) \\ V_{\text{RI}} &= V_{\text{o}}(t_{\text{ES}}) \\ V_{\text{DF}} &= V_{\text{o}}(t_{\text{ED}}) - V_o(t_{\text{ES}}) \\ V_{\text{DEF}} &= V_{\text{v}}(t_{\text{ED}}) - V_v(t_{\text{ES}}) \end{aligned} \tag{12.3}$$

12.2.5 Particle Tracking

The particle tracking algorithm was introduced and initially investigated in a student thesis [137]. To examine the behavior of the flow field and to enable a tracking of single particles, the Lagrangian coherent structures are identified on the basis of the finite-time Lyapunov exponent. Thus, in a post-processing step the velocity field U resulting from the simulation is processed such that $n \approx 15000$ particles are analyzed.

The position x of a particle at time t is determined by its initial position x_0:

$$x = x(x_0, t) \tag{12.4}$$

To deduce the position based on the velocity field U from the simulation, the following ordinary differential equation has to be solved.

$$\dot{x}(x_0, t) = U(x(x_0, t), t) \tag{12.5}$$

The solution is obtained by a time integration in the interval $[t_0, t_e]$, with t_e being the time that

$$x(x_0, t) = \int_{t_0}^{t_e} U(x(x_0, t), t)\, dt \tag{12.6}$$

In figure 12.3, the distribution of the five different blood flow components in the initial state t_0 are visualized. Following their definition, DF and RI components start in the LA. DEF as well as V_{res} are located in the LV.

12.3 Results

Throughout all positions and severity stages, a clear regurgitating jet is observed. As exemplary shown in figure 12.4 for the stage of severe MVR caused by a hole located at the right side of the anterior leaflet (P3), a jet develops with the beginning of the systolic phase ($t = 0.3\,s$) and spreads deeper inside the LA as time progresses. Additionally, a mixing process with LA blood is initiated by the ascending, slightly rotating upper end of the jet, which can be observed at the left boundary of the LA. A slight back flow into the pulmonary veins can also be observed.

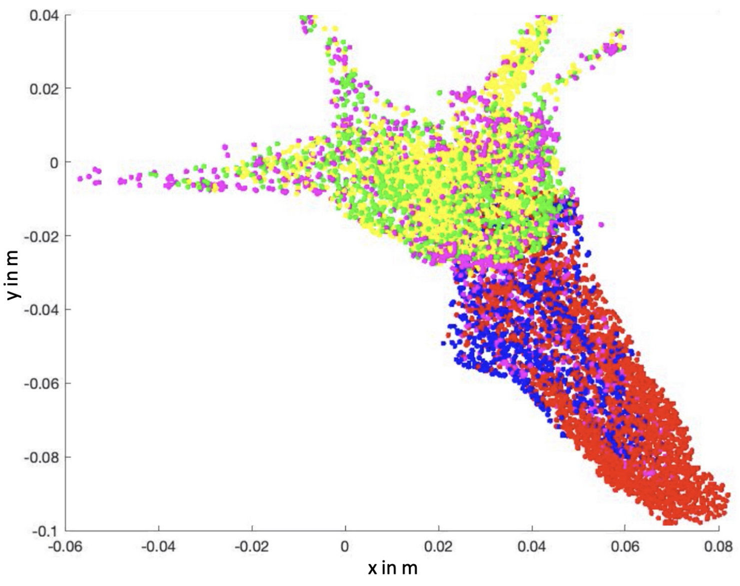


Figure 12.3: Initial distribution of blood flow components in the initial state t_0. Color codes for the flow components: DF (green), RI (yellow), DEF (blue), V_{res} (red) and BF (pink).

While comparing different severity stages at the same spatial position (P3), altered flow patterns (particular in end-diastole) and wash-out fractions are observed in the diseased geometry. Compared to a healthy case (figure 12.5A), the blood spreads deeper towards the apex in the diseased case (figure 12.5B-D) while the LV itself is less washed out. The vorticity in the diseased case was higher throughout all severity grades. With increasing MVR severity, a non-linear behavior of remaining initial blood volume was observed. Comparing four heart cycles, depending on the cardiac phase and MVR severity, sections with de- and increased values were obtained. For example, for moderate mitral regurgitation, the ventricular wash-out was up to 6 % lower than in the healthy control case (see figure 12.6).

The investigation of flow components revealed a significant decrease of the DF component for all MVR severity stages. Quantitative results, exemplary for moderate MVR can be found in table 12.2. For this case the regurgitant blood causes a DF decrease of 18 %. All other blood flow components vary slightly around $\pm 2\,\%$. Already for a moderate MVR, the regurgitant volume was around 15 % of LV volume. Evaluating the results of the particle tracking algorithm confirmed the previous results. The direct flow was reduced by more than 10 %, accounting for the slower washout progress. The visualization of all five blood flow components confirms the high concentration of residual volume blood in the apex as well as a low mixture of V_{res} blood with blood entering the LV in the current cycle in the lower part of the LV (figure 12.7).

Finally, the impact of the disease location is evaluated. Table 12.3 presents an overview of the regurgitation volume and the fraction of regurgitating flow to the severity stage and the position of MVR. The flow quantities are averaged over the last three simulated heart cycles to average variability. It clearly can be observed, that the major differences in BF

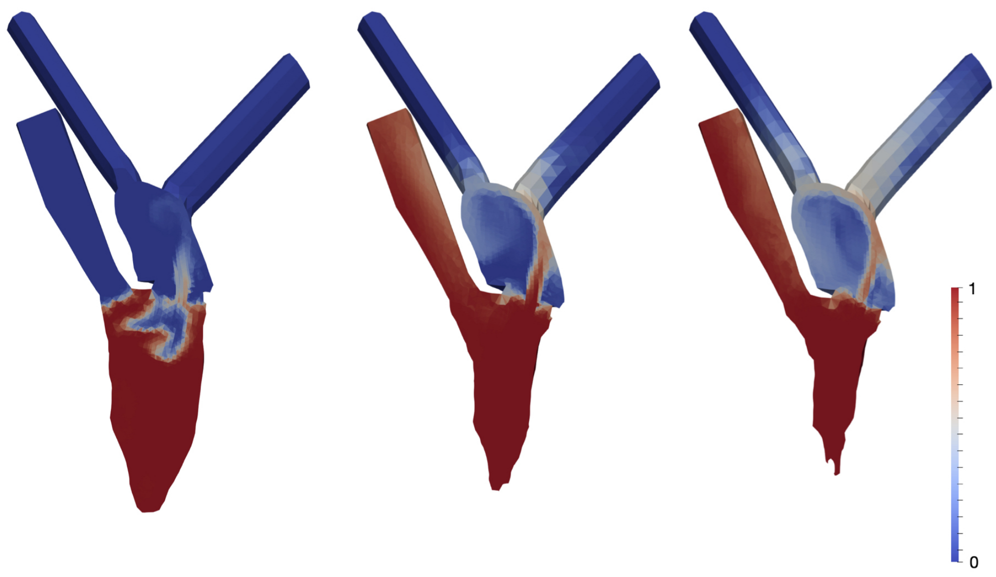


Figure 12.4: Long axis cut through the left heart with the scalar Ψ color coded. Different points in time during the ventricular systole: The begin of the ventricular contraction at $t = 0.3\,s$ (left), as well as $t = 0.5\,s$ during the contraction (central). On the right hand side the end of contraction and thus the begin of relaxation phase ($t = 0.7\,s$) is shown. The clipping plane is kept constant, the reduced size of the LV is not only caused by the contraction, but also due to a twisting movement out of the clipping plane. Clearly visible is the regurgitating flow from LV to LA with increasing intensity over time.

condition	direct flow DF	retained inflow RI	delayed ejection DEF	residual volume V_{res}	back flow BF
healthy	35.3 %	16.0 %	21.9 %	26.8 %	-
moderate MVR	17.3 %	17.5 %	24.3 %	25.5 %	15.4 %

Table 12.2: Ratio of blood flow components, exemplary for moderate MVR (P3), in comparison to the values for the healthy case. Table adapted from [39], published under CC BY 4.0 license.

are caused by the different stages of MVR. However, also for the same severity of MVR, slight differences between the three positions exist: Comparing P1 and P2, the BF varies up to 3.2 % for severe MVR. While these differences can be explained by the difference in resulting EROA surface of up to $0.05\,mm^2$, a systematic decrease of V_{res} and BF can be observed for position P3. While all EROA values are higher than the ones for P2, BF is always decreased. This effect cannot be observed with a hole located at P1. Thus, with a more lateral hole in the MV, located at the P3 posterior leaflet, regurgitation volume is reduced.

Summing up, modeling the MV with a simple planar valve model had shown to deliver accurate results in healthy heart models [38]. Extending this approach to patients suffering from MVR clearly reveals an alteration of blood flow patterns as well as altered ventricular wash-out fractions and significantly reduced direct flow. Additionally, the implementation of

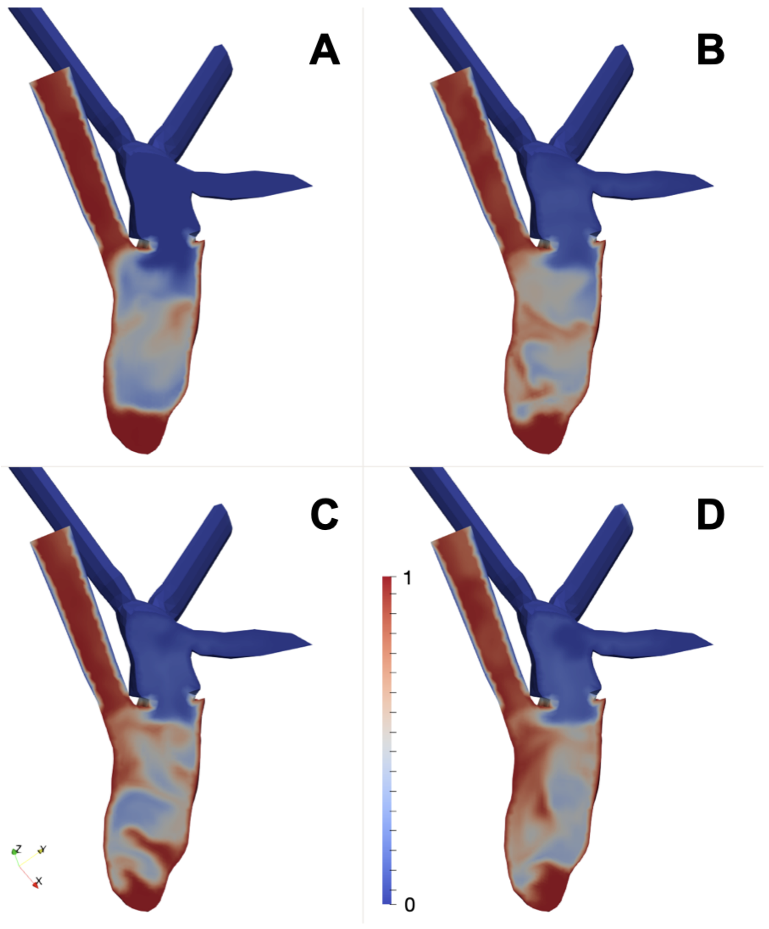


Figure 12.5: Long axis slices of the healthy heart (A) and mild, moderate and severe MVR (B-D). Time: Atrial contraction of the second cycle $t = 1.39s$. Figure adapted from [39], published under CC BY 4.0 license.

a particle tracking algorithm for post-processing the computational fluid dynamics (CFD) simulation data enables to visualize the different flow components and to analyze start and end positions of the concerning trajectories. However, the full quantitative analysis and comparison of different severity stages remains challenging and further analysis tools have to be implemented to further measure LV vorticity. All in all, this work and the implemented methods build a good basis to characterize vortex structures in mitral valve disease in future studies.

12.4 Discussion

Comparing the three severity stages presented in the long axis view in figure 12.5, a clear distinction between the severity stages is hard to define. Additionally, the visualized long axis views only show a single plane of the whole LV volume. Taking into account the

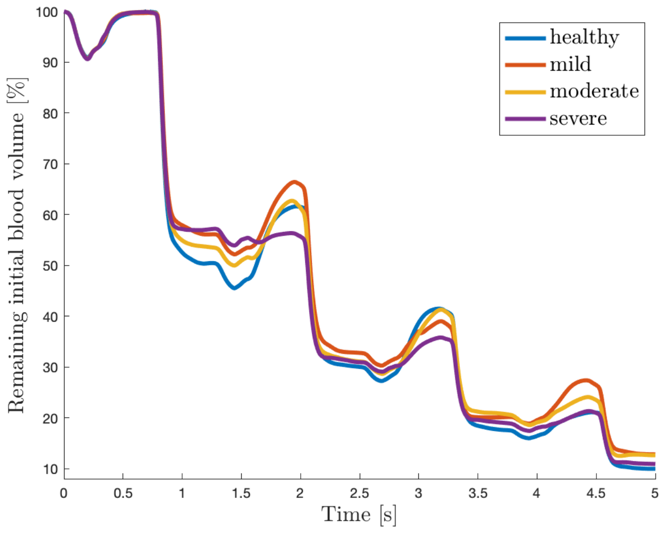


Figure 12.6: Scalar Ψ as a measure for local concentration of initial blood volume that remains in the LV over a time course of four heart cycles. Different degrees of mitral regurgitation. Figure adapted from [39], published under CC BY 4.0 license.

MVR severity	**hole location**	**EROA** in mm^2	**averaged** V_{reg} in ml	**averaged BF** in %
	P1	0.099	13.6	12.3
mild	P2	0.109	16.2	14.6
	P3	0.126	15.4	13.9
	P1	0.282	24.3	21.9
moderate	P2	0.224	24.1	21.8
	P3	0.259	20.9	18.9
	P1	0.342	29.3	26.5
severe	P2	0.396	32.9	29.7
	P3	0.419	28.9	26.0

Table 12.3: Simulated regurgitation volumes V_{reg} as well as regurgitating flow (BF) in depending on effective regurgitation orifice area (EROA) and MVR severity. Results adapted from [136].

non-linear behavior of the V_{res} time courses in figure 12.5 additionally intends that only analyzing a mean value across the whole LV does not account for local variations in altered flow conditions. Therefore, the presented measures may not be sufficient to fully quantify the resulting flow alterations. In addition to the presented methods, 2D short axis slices of the LV could be analyzed regarding their vorticity behavior. An analysis and quantification of vortex structures in different heights could be carried out to account for vorticity location maps. This method could also be used as a measure for the immersion depth of the ventricular jet. This may link to the work of Collia et al. [142], that clearly associated a prolapsed MV with a higher amount of residual volume close to the apex as expected from a lower jet energy. However, a literature comparison will remain challenging, as the classification

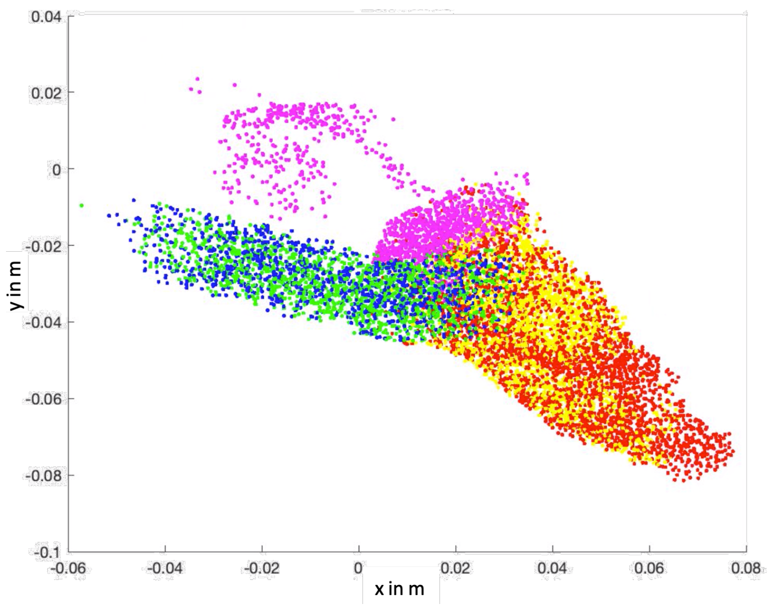


Figure 12.7: Visualization of blood flow components (moderate MVR) in end-diastolic state. Color codes for the flow components: DF (green), RI (yellow), DEF (blue), V_{res} (red) and BF (pink). Figure adapted from [39], published under CC BY 4.0 license.

of MVR severities is only based on upper and lower limits instead of exact values, as shown in table 12.1. Another point to further investigate is the capability of the MV model. As different severity stages are only distinguished by a de- or increased hole diameter, the circular symmetric hole in the medial direction of the posterior leaflet may not fully cover all MVR effects. Implementing a 3D parametric valve model parameterized on the basis of measurement data will serve as a comparison in further studies. Additionally, a closed-loop model of the whole circulatory system, combined with the investigation of both heart sides could deliver further insights in right ventricle (RV) function and the risk of RV failure, as Bonini et al. showed [143]. The method of using a scalar transport to account for blood concentrations however delivered accurate results. A comparison to literature e.g., to Eriksson et al. [144] 4D MRI measurement data showed good accordance with our simulation results of the healthy heart geometry with deviations around $\pm 2\,\%$ for the single flow components.

Chapter 13

Hypertrophic cardiomyopathy

13.1 Introduction

In hypertrophic cardiomyopathy (HCM), the myocardial wall of the left ventricle (LV) is pathologically thickened. In this disease of clinical relevance, the risk for sudden cardiac death is significantly increased. Therefore, an early-stage diagnosis is valuable in terms of decreasing the mortality rate. While in addition to the thickened wall, a myocardial fiber disarray is characteristic for the disease, the movement of the endocardial wall may be altered. Thus, the simulation of fluid dynamics in a HCM heart geometry is of interest to assess to which extent fluid dynamic quantities are altered. Another relevant impact of the disease is the possible outflow tract obstruction. With partial occlusions of different size, an impact on the ejected volume and the flow patterns may occur. However, as the occlusion is not modeled in the data set of the elastomechanics simulations, it is not possible to investigate this scenario in the scope of this thesis. Thus, with a focus on the wall thickening and the fiber disarray, the influence of the pathological conditions on the fluid dynamics in the LV is assessed to address the question, if flow patterns are altered.

13.2 Methods

Fluid dynamic simulations are carried out on a structured cubic grid, based on the immersed boundary method (IBM) approach. All cells close to the endocardial surface as well as the valves are refined. The endocardial surfaces to prescribe the moving wall velocity are obtained from the elastomechanical simulation framework. Inlet and outlet velocity, as well as the outlet pressure are prescribed with the same reasoning as in chapter 10 and are extracted from the circulatory system simulation results. For the 3D representation of the aortic and the mitral valve (MV), the planar (aortic) as well as the Linköping valve model are considered. To assess flow patterns as well as to evaluate the fraction of the initial blood volume that remains in the LV, a passive scalar transport equation is implemented as will be described in section 14.2.7.2. The scalar ψ is considered as a measure of local concentration of blood. It is initialized with a value of $\psi(x,t_0) = 1$ for all cells located in the LV.

The increased wall thickness is visualized in the right part of figure 13.1. It is compared to a model of the healthy heart to visualize differences. In addition, the vectors depicting the fiber direction are included for the healthy (left) and the diseased case.

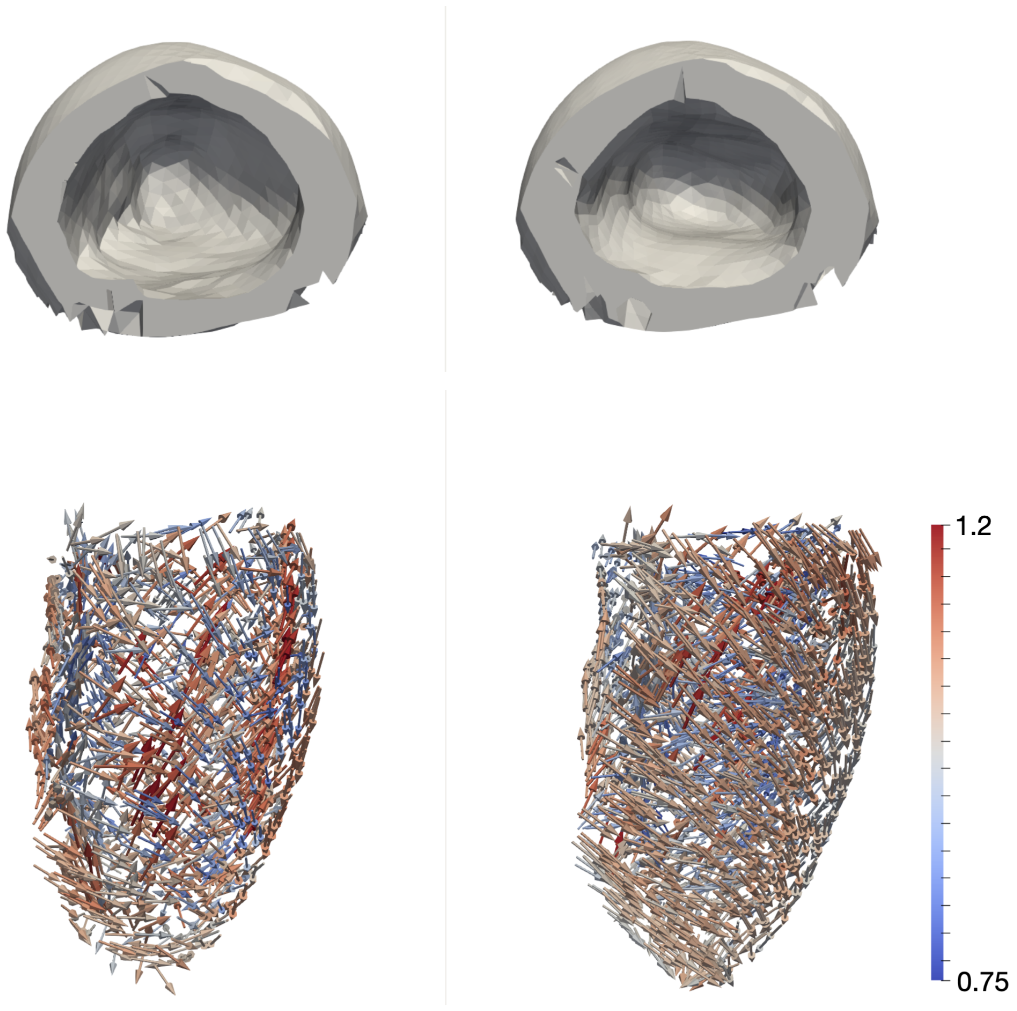


Figure 13.1: Right column: visualization of the pathological HCM geometry. The thickened wall of the LV is visualized from the top perspective. In addition, the vectors indicating the fiber direction are included. For a comparison, on the left hand side, a healthy heart geometry is included.

13.3 Results

The cubic grid is of size 0.225 m $\times$ 0.19 m $\times$ 0.245 m with a basic edge length of approximately 6 mm. During a refinement step all cells close to the moving wall are divided in 64 cells, while all cells in the valve regions are divided in 512 cells. The resulting edge length in the valve zone is approximately 0.75 mm. All in all, the model comprises 697,027 volume elements.

To compute the position of the moving wall prescribed by the external surface geometry files, again a distance field is computed to deduce the scalar indicator field. The resulting distance field Λ is visualized in figure 13.2. Based on equation 4.3 the indicator field Φ is processed and the velocities resulting from the displacement in the endocardial wall geometries are prescribed.

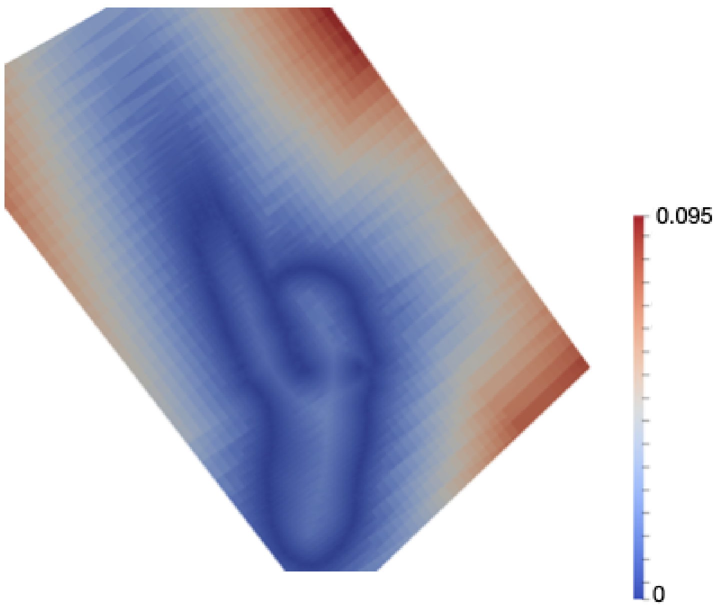


Figure 13.2: Distance field Λ in m. Euclidean distance (ED) of each cell center to the constraining endocardial surface.

In figure 13.3, the mapping field κ visualizing the effective enclosed volume is shown for the initial diastolic state (left), as well as the systolic state with a minimum left ventricular volume. In addition, the cells with the velocity prescribed by the wall movement are visualized in gray.

The prescribed 3D geometry of both applied valve models is immersed in the structured grid. The position of the valves is shown in figure 13.4: The aortic valve (left) as well as the MV are visualized in the open state.

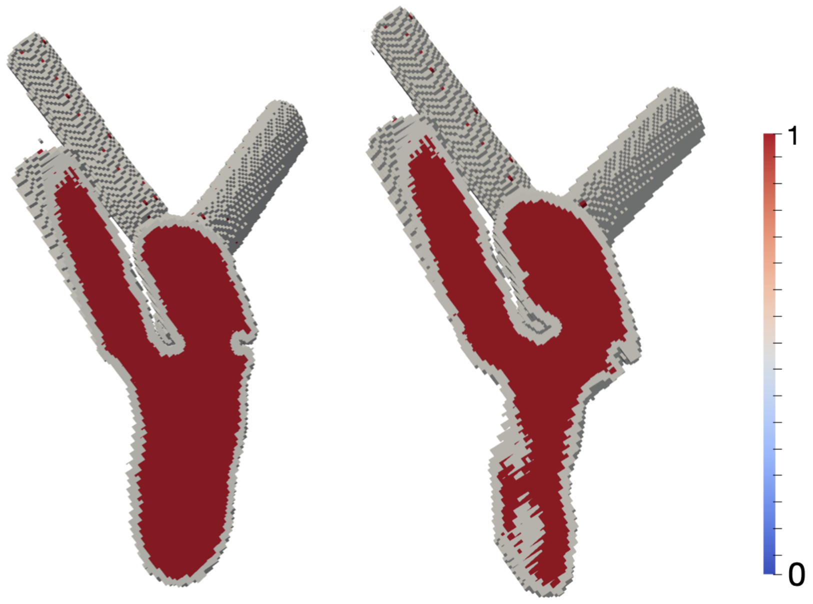


Figure 13.3: Mapping field κ, denoting the volume enclosed by the moving wall. Left: diastolic state. Right: systolic state.

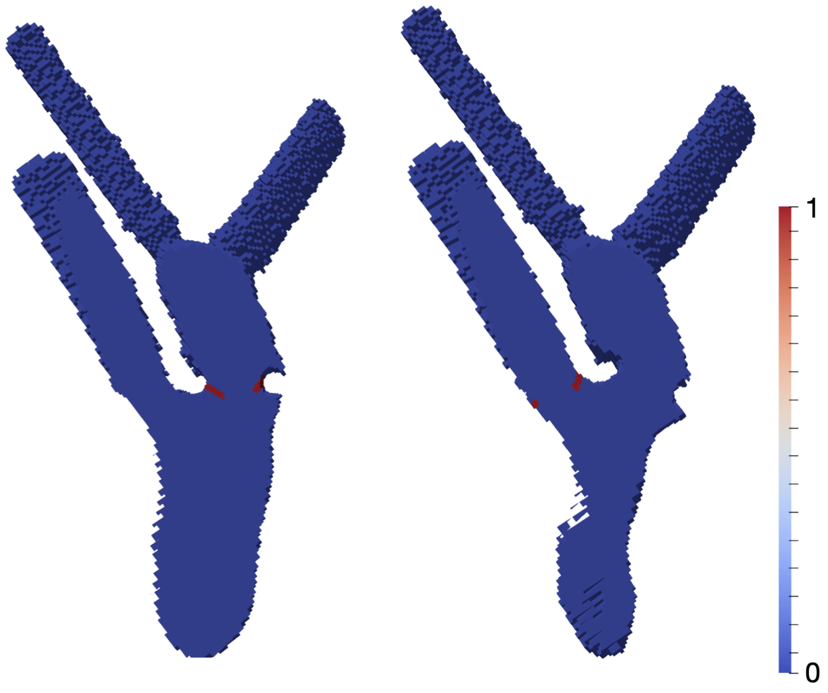


Figure 13.4: Scalar field indicating the location of the aortic (right) and the mitral valve (left) immersed in the cubic grid. Both valves are shown in then opened state.

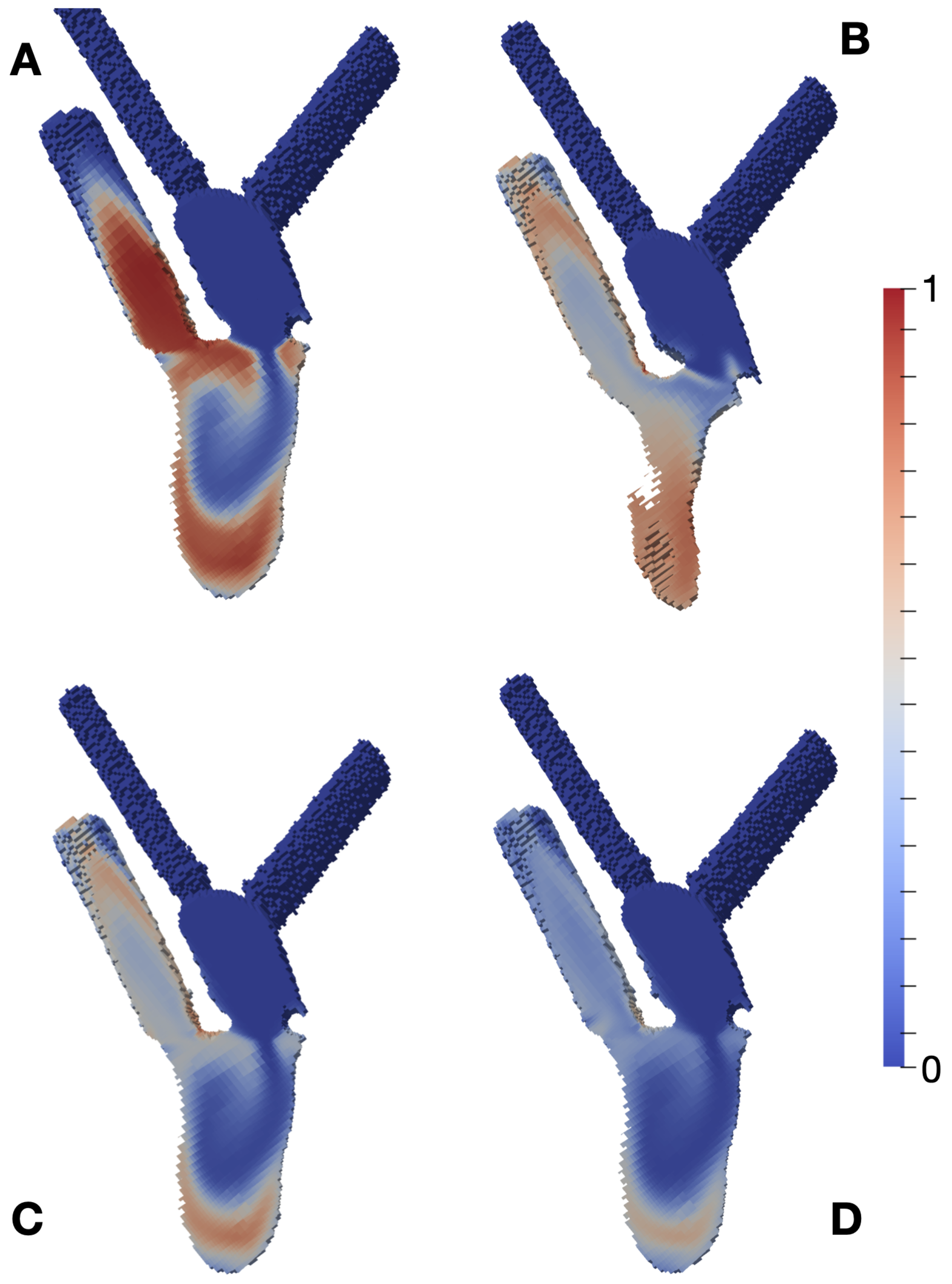


Figure 13.5: Flow patterns in the left part of the HCM heart computed based on the indicator field ψ for several heart cycles: The diastolic state after the first heart cycle (A), as well as the following systole (B) is presented in the top row. In the lower row, the diastolic state after the second (C) and third (D) heart cycle are included.

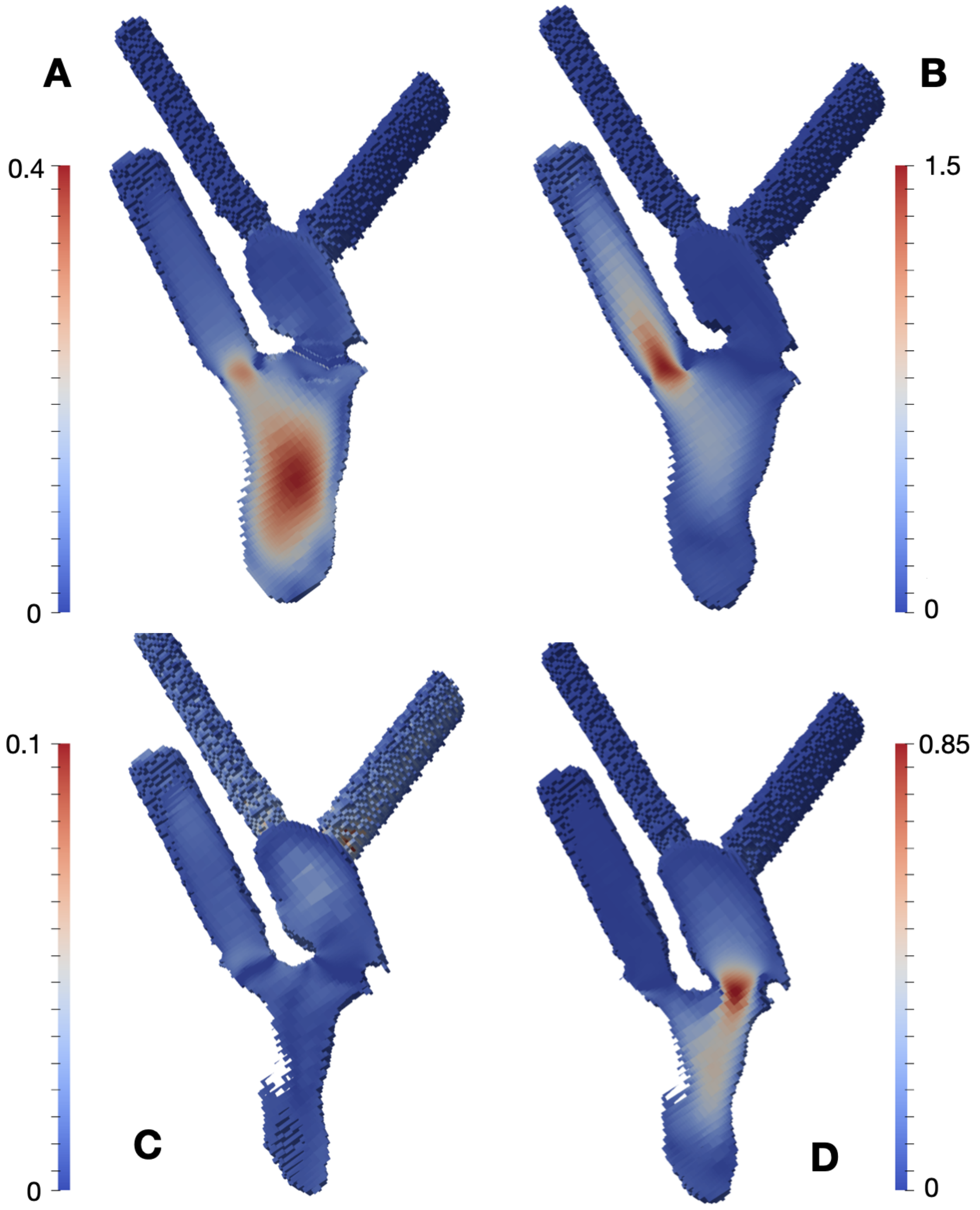


Figure 13.6: Magnitude of the velocity field u in m s^{-1} at different times throughout the cardiac cycle: At the beginning of the ventricular contraction (A), as well as immediately afterwards, when blood is ejected through the aortic trunk (B). The systolic state (C) as well as a time step during the relaxation phase with an inflow across the MV (D) are included in the bottom row.

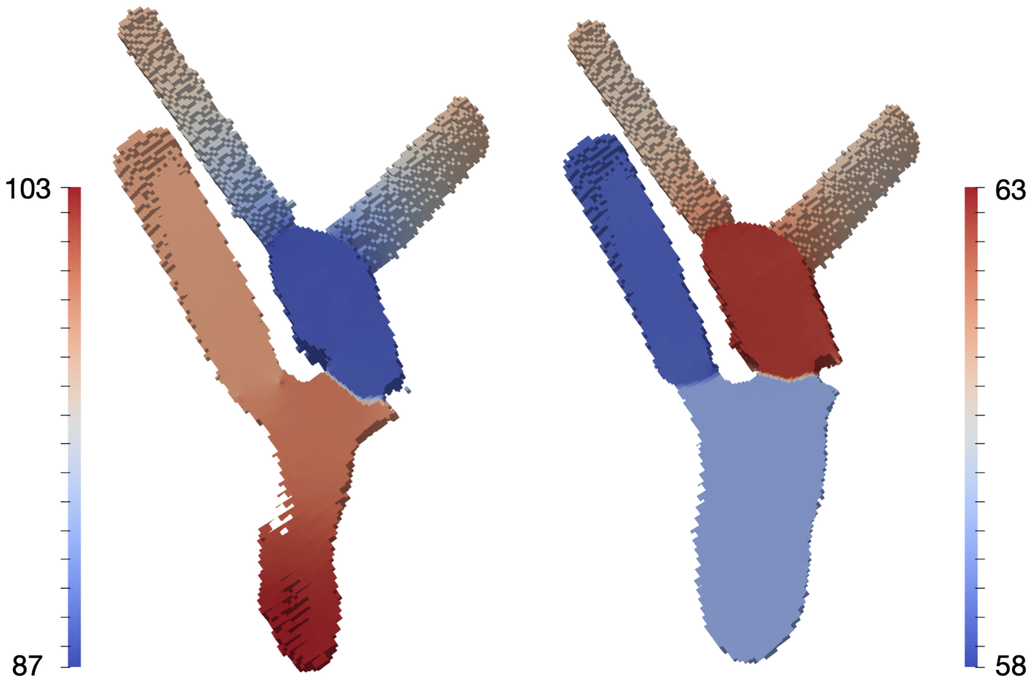


Figure 13.7: Pressure field p in mmHg mapped across the long axis view of the clipped HCM heart geometry. Left: Systolic state. Right: Diastolic state.

In figure 13.5, the flow patterns resulting in the LV are shown for different points in time throughout the series of the simulated four consecutive heart beats. While the flow patterns in the diastolic state after the first heart cycle (A) show a clear jet penetrating towards the apex, a circular washout structure can be observed in the center of the LV. While it seems as if the ejected blood of the first cycle does not reach the end of the aortic trunk, this is just a visualization problem: As the aortic trunks bends out of the cut plane, the corresponding volume is not clipped any longer. In the systolic state of the second heart beat (B), it can be observed, that a considerable amount of blood that entered the LV in this cycle, directly leaves the LV. The following relaxation phase (C) reveals a similar behavior than the first one: The vortex structure is present, however, it seems to prevent blood to fully penetrate towards the apex. In the diastolic phase after the third heart cycle (D), beneath a proceeding washout, the same patterns are observed. This results in a retained fraction of initial LV blood close to the apex region. The observed retained volume is confirmed by the time course of residual volume V_{res} computed based on the scalar field ψ: After five full heart cycles, a residual volume of 15 % remains.

The resulting velocity fields are plotted for different points in time during the cardiac cycle. At the exact start of the ventricular contraction, an increased velocity in the whole center part of the LV can be observed (A). In the subsequent time step (B), the velocity in the aortic valve rises rapidly, up to a maximum value of 1.5ms^{-1} in the center of the valve. In accordance with the previous simulations, this indicates that the narrow valve region may be modeled to tight. When the lowest chamber volume occurs just before the inflation of the LV begins (C), the flow nearly comes to a still stand. As soon as the LV inflates, blood is sucked from the left atrium (LA) through the mitral valve, again resulting in a local maximum of the velocity.

Finally, the pressure across the left heart side is included in figure 7.5. During the systolic state (left), the pressure level in the LV and the aortic tube is in the same range, as the aortic valve is opened. While the mitral valve is closed, a pressure drop towards the LA is modeled as expected. In comparison, just before the start of the ventricular contraction, both valves are closed during the isovolumetric contraction. Due to the IBM representation of the valves, the LA is completely decoupled from the outlet pressure in the aortic trunk and thus a higher pressure above 60 mmHg occurs.

13.4 Discussion

Applying the IBM fluid dynamics solver, the flow quantities in a HCM heart geometry were evaluated successfully. Simulating these pathological conditions, the most prominent finding was a vortex forming in the upper part of the LV that prevents the jet from fully penetrating towards the apex. This also results in a comparably high residual volume of 15 % after four heart cycles. However, now it has to be argued whether these finding are really based on the flow conditions implied by the pathological geometry or if an influence of the valve model supports the formation of this vortex structure. In the previous chapters it was already found that the influence of the valve geometry and timing may have a strong impact on the blood flow dynamics. To validate this procedure, measurement data e.g. 2D flow across the valve would increase the reliability of the model.

For future studies, it would be of high relevance to simulate geometries that also include the obstruction of the outflow trunk. Assessing fluid simulations, the impact on flow patterns and blood flow velocities, but also global quantities as ejection fraction and residual volume may be significant.

PART VI

FLUID MECHANICS AND ELASTOMECHANICS COUPLING

Chapter 14

Sequential coupling algorithm

At the interface between the fluid and the solid mechanics domain, this sub project aims to closer investigate the mutual influences and dependencies of the two central areas of physics in the context of cardiac hemodynamics. In addition to the analysis of the influence in an in silico model of a healthy heart, the implementation and test of a novel coupling approach is in the focus of this chapter.

The content of this chapter is adapted from the publication in the journal Frontiers in Cardiovascular Medicine by Brenneisen at al. under the Creative Commons Attribution 4.0 (CC BY 4.0) international license [40]. First results concerning the pressure factor (PF) were additionally presented at a conference [145].

14.1 Introduction

To accomplish its central task, the heart tissue contracts to eject blood from the heart chamber and beat by beat drive the blood flow. On the first glance, this is a simple cause and effect chain of an elastic tissue. But through its multi-physics nature on different temporal and spatial scales, modeling this process in silico as accurate and computationally beneficial as possible remains a tremendous challenge [146, 147]. Especially striking a balance between high numerical accuracy and low computational effort remains a bottleneck for the transition of numerical simulation tools into clinical applications [148, 149]. In this context fluid-structure interaction (FSI) between the heart muscle and the blood flow plays a key role: As the endocardial walls form the physiological boundary layer for the blood, a close interdependence of the concerning quantities is essential for a correct model. Common concepts and basic strategies of implicit and explicit coupling strategies have been included in the state of the art section 1.2.3. The exchange of information between the single solvers for each different area of physics is one of the key aspects in the cardiac modeling process. However, the degree to which the spatially resolved blood flow has a retrograde effect on the cardiac mechanics in this multi-physics problem remains unresolved up to now [53].

Thus, this in silico study gives an insight to which extent the flow influences the local variations of cardiac wall deformation. This is of interest as widely used 3D-0D coupled circulation systems only consider mean spatial chamber pressure values (compare section 1.2.2). There, the spatial deviations of the flow velocity field are averaged over the whole chamber. Thus, local increases or decreases are neglected and a unique pressure is considered across the whole chamber. Especially in specific phases of the cardiac cycle like e.g., the ejection phase during the ventricular systole, it can be hypothesized that this procedure neglects relevant spatial information. This phenomenon is evaluated quantitatively under physiological conditions in this study.

Second, the retrograde effect of local spatial deviations in the pressure field caused by the blood flow in the heart cavities has an influence on the structural mechanics. Thus, changes in the local deformation of the heart caused by forces resulting from the blood flowing against the myocardial tissue are quantified. Following Bernoulli's principle, areas with an increased flow rate are linked to a lower static pressure acting orthogonal to the myocardial wall. Therefore, it can be expected to find local deviations in the displacement of endocardial wall nodes.

Based on the results of [150] wall shear stress is assumed to be neglectable compared to the internal stresses of the myocardium. Therefore, the endocardial wall pressure field due to flow dynamics is analyzed. Subsequently, a pressure factor is introduced to the solid mechanical simulation. This factor incorporates the information of the spatially resolved pressure deviations from the mean chamber pressure from the fluid dynamics simulation. It is considered in the mechanics simulation to quantify deviations in wall deformation due to the effect of the flow.

Summing up, in this chapter two central basic questions concerning the fluid and solid mechanics coupling procedure are investigated: First, whether decreases or increases of the local pressure resulting on the endocardial surface elements are quantified and localized. These deviations are neglected in the unidirectionally coupled algorithm so far. Second, whether the deviation of the mechanical displacement, that results if the pressure deviations are considered in the solid mechanical simulation, is quantified. Thereby, a cycle-to-cycle coupling algorithm that accounts for the influence of FSI in a computationally and structurally beneficial way is introduced. The previously utilized unidirectional coupling is replaced by a simple and low effort bidirectional approach.

14.2 Methods

In this section, first the organizational scheme of the coupling algorithm is introduced. After some details on the mechanics domain and fluid solver, the coupling procedure itself based on the extraction of the pressure gradient field is introduced.

14.2.1 Algorithm structure

The basic structure of the sequential coupling algorithm is a partitioned approach. This takes advantage of the fact that two stand-alone single area of physics solvers for fluid

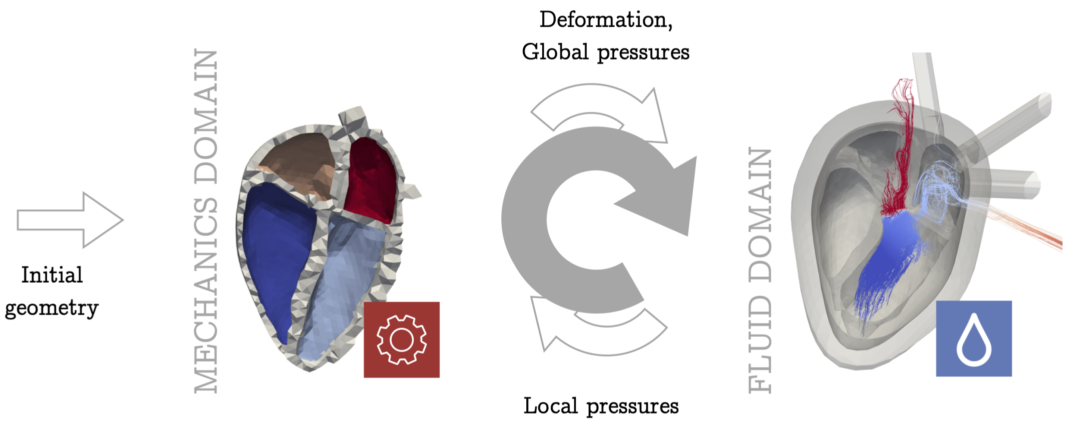


Figure 14.1: Overview of coupling procedure. The arrows denote the flow of information between the two domains. The upper arrow shows the unidirectional information flow from the mechanical domain to the fluid solver as it was implemented before (e.g., [38]). Now the lower arrow takes into account the retrograde effect from fluid dynamics in the mechanical simulation. This sums up to an iterative sequential procedure (gray arrow). Both - in the mechanics and in the fluid domain - several full heart beats are simulated before handing over the data to the other domain. Figure adapted from [40], published under CC BY 4.0 license.

dynamics [107] and solid mechanical deformations [110] are already available from previous projects. In the previously existing unidirectional coupling, information about deformation and pressure conditions are passed from the solid mechanics simulation tool including the circulatory system model to the stand-alone fluid dynamics simulation. By closing the loop of a classical partitioned FSI approach, the two independent solvers are linked by the feedback of fluid dynamic information towards the solid mechanical solver after each heart cycle. Therefore, the local pressures are accessed, allowing for a quantification of the fluid dynamic influence in a first evaluation step. Taking into account these retrograde effect in the solid mechanics simulation also closes the loop of a bidirectional coupling algorithm on the second stage. It fulfills the purpose of a coupling algorithm, which is categorized between a loosely coupled and a strongly coupled approach, as the iterative exchange of data is not realized each time step, but iteratively with each full heart cycle. This sequential cycle-to-cycle coupling accounts for the influence of FSI in a computationally and structurally beneficial way. The basic structure of the sequential coupling algorithm is visualized in figure 14.1. Details of the coupling procedure are explained in the following section 14.2.5.

14.2.2 Mechanics domain

Simulations are performed on an individualized whole-heart geometry created from medical imaging data of a healthy volunteer. To estimate the stress distribution present during image acquisition, the pressure free geometry is estimated by applying the unloading algorithm introduced in section 3.4.4 assuming a pressure of $p_{0,\mathrm{LV}} = p_{0,\mathrm{LA}} = 7.5\,\mathrm{mmHg}$ and

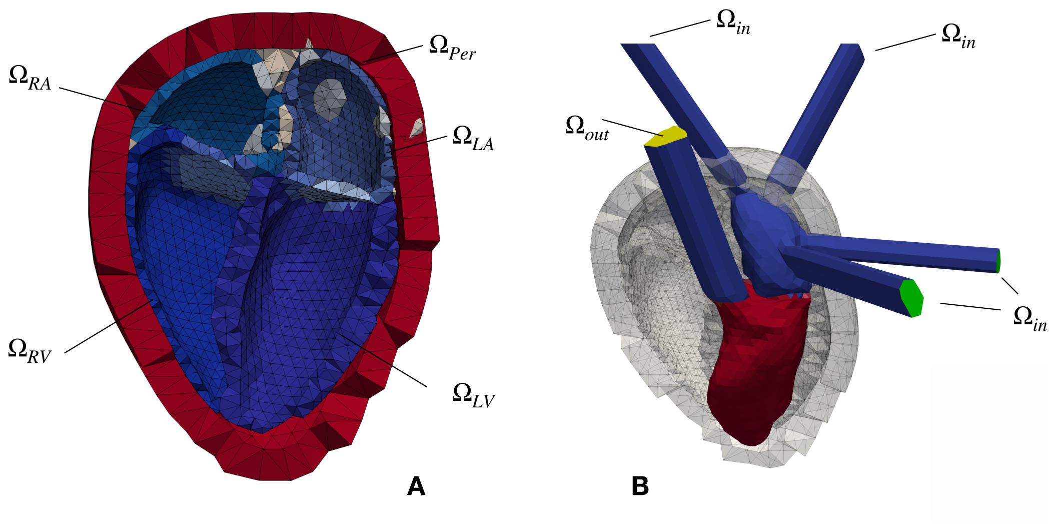


Figure 14.2: **(A)** Clipped heart geometry with the myocardial tissue of the four chambers colored in shades of blue. The endocardial surfaces of the left atrium (Ω_{LA}), left ventricle (Ω_{LV}), right atrium (Ω_{RA}) and right ventricle (Ω_{RV}) form the boundary layer for the fluid simulation. Additionally labeled is the pericardial layer (red, Ω_{Per}) as well as in gray color the area, in which the pulmonary veins (left side of the heart) and the superior vena cava (right heart) open into the corresponding atrium. The blue-gray volume between the atrium and the ventricle denotes the position of the mitral valve (mitral valve (MV), left heart side) and the tricuspid valve (right heart side). **(B)** Fluid geometry of the left side of the heart with elongated vessel trunks. The pressure inlet surfaces (pulmonary veins) are highlighted in green color, whereas the pressure outlet (aorta) is colored yellow. Also depicted is the initial spatial distribution of the scalar Ψ of the scalar transport equation, color coded in red ($\Psi = 1$) and blue ($\Psi = 0$). The gray colored background structures represent the pericardial layer as well as the myocardial tissue in the initial, diastatic state. Figure adapted from [40], published under CC BY 4.0 license

$p_{0,RV} = p_{0,RA} = 4\,\mathrm{mmHg}$ [151]. In addition, the influence of the pericardial layer is considered to represent the influence of surrounding tissue [120]. The mechanical finite element mesh comprises 8,700 nodes and 49,900 linear tetrahedral elements. A clipped model of the resulting heart geometry is shown in figure 14.2A.

As introduced in section 3.4, the myocardial tissue (depicted in different shades of blue in figure 14.2A) is modeled as a hyper-elastic material. The parameters to model the passive material properties are listed in table 14.1.

For the rule-based determination of fiber orientation [114], fiber angle of +60° (endocardial wall) and -60° (epicardial wall) are applied. The periodicity in the active tension development of the double Hill model [115] is set to the duration of one heart cycle ($t = 1.247\,\mathrm{s}$) in accordance with the magnetic resonance imaging (MRI) measurement data.

14.2.3 Fluid domain

Fluid dynamics simulations are executed in the open-source software framework openFOAM under the version v1912 [2]. The fluid solver is based on a finite volume representation

Tissue	**Domain**	**Model**	**C** in kPa	$\mathbf{b_f}$	$\mathbf{b_t}$	$\mathbf{b_{ft}}$	**K** in kPa	ρ_0 in kg m^{-2}
Ventricle	LV, RV	Guccione	0.278	12.0	4.8	8.4	200	1,082
Atrium	LA, RA	Neo-Hooke	7.45	-	-	-	200	1,082
Surrounding	pericardium	Neo-Hooke	10.0	-	-	-	1,000	1,082
Surrounding	fat	Neo-Hooke	3.725	-	-	-	1,000	1,082
Vein	pulmonary vena cava	Neo-Hooke	14.9	-	-	-	200	1,082
Artery	pulmonary, aorta	Neo-Hooke	14.9	-	-	-	200	1,082
Valve plane	mitral, tricuspid, aortic, pulmonary	Neo-Hooke	200	-	-	-	200	1,082

Table 14.1: Passive material parameters used for the different tissue areas in the mechanical simulation. Table adapted from [40], published under CC BY 4.0 license.

of the mesh accounting for the blood volume. The system of equations is solved in an Arbitrary Lagrangian Eulerian (ALE) framework as introduced in section 3.2.2 to represent cardiac contraction. The wall movement (displacement of endocardial boundary) resulting from the mechanical solver is provided as a time series for one full heart cycle. A Laplace equation with quadratic diffusivity is solved to determine the mesh motion velocity of the inner volume cells. Hence, a no-slip condition is applied at the ventricular wall for both velocities: the blood velocity, u_i, and the mesh motion velocity, c_i. The spatially resolved pressure is obtained for each element using a PIMPLE loop (PISO algorithm [152] with iterative marching).

For the fluid simulations, all inlet and outlet trunks are prolonged as described in section 7. The flux boundary conditions at the inlets Ω_{in} and outlets Ω_{out} are Neumann boundary conditions, such that the global volume flux, $\dot{V}$, is in accordance with the volume change prescribed by the boundary conditions from the mechanics simulation. The inlet pressure is prescribed at the pulmonary veins. The outlet pressure is prescribed at the aorta. These Dirichlet-type boundary conditions are provided by the circulatory system model. The left heart as well as its embedding in the surrounding layers (gray color) is shown in figure 14.2B.

All four valves are modeled by means of a porous material law that generates a pressure drop across the respective valve plane as shown in [38]. Since the whole heart and therefore also the valve planes are moving with time, the porosity model developed by [153] is incorporated. It considers an increased kinetic energy term f_i in the NSE (equation 3.8) based on the mesh displacement. The valves are controlled by the ventricular volume changes resulting from the boundary conditions such that all valve planes are blocked if $\left|\dot{V}\right| < 20\,\text{ml}\,s^{-1}$. Up to a flux of $\left|\dot{V}\right| < 160\,\text{ml}\,s^{-1}$, the permeability rises linearly with the flow and the valve plane is fully permeable if $\left|\dot{V}\right| > 160\,\text{ml}\,\text{s}^{-1}$. The same limits in reverse order apply for the closure of the valves. The porosity $\phi_p = 1$ is kept constant. Exemplary, the valve plane permeability k_p is shown in figure 14.3 for one heart cycle.

Based on the mitral valve diameter d_{MV} = 15 mm, the Reynolds number reached at maximum velocity $u_{\max}$ in the valve plane is $re_{max} = U_{\max}\rho d_{\text{MV}}/\mu = 2760$. The Womersley number lies in the physiological range [154], as $wo = d_{\text{MV}}\sqrt{\omega\rho/\mu}$ = 19.3, where ω repre-

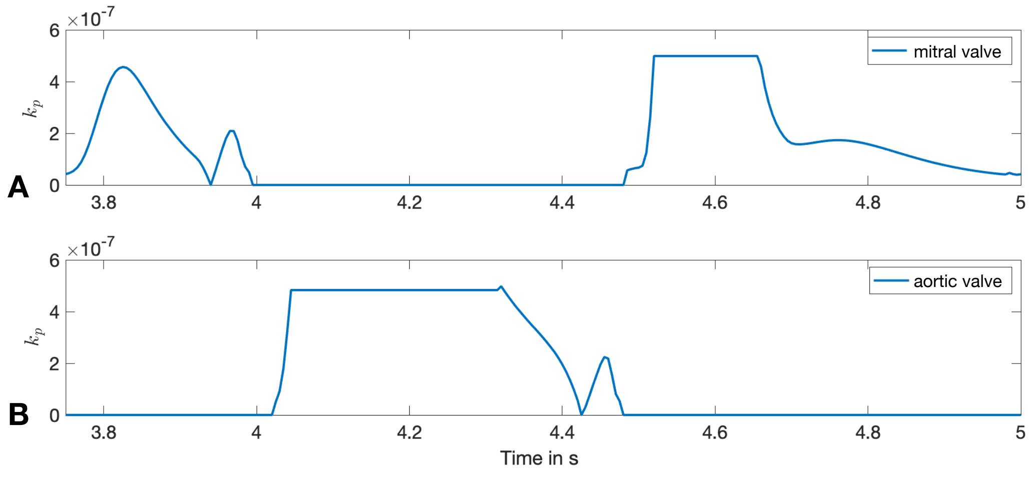


Figure 14.3: Porosity of the valves during the time course of one heart cycle. Both sub-figures show the valve plane permeability k_p for the left side of the heart. Horizontal lines depict a fully opened or closed valve state. (A): Mitral valve. (B): Aortic valve. Figure adapted from [40], published under CC BY 4.0 license.

sents the angular frequency. The tetrahedral fluid volume mesh consist of 143,000 cells in the left heart, which corresponds to the number of cells found to be sufficient by [155]. The simulations start with an initial time step of $\Delta t_0 = 0.001$ s. Afterwards, the time stepping is adjusted automatically to satisfy a Courant-Friedrichs-Lewy number of $CFL \leq 0.7$. At the initialization of the solver, the fluid is at rest.

14.2.4 Pressure gradient

Up to now, only unidirectional coupling influences from the solid mechanics simulation are considered by means of a moving boundary condition in the fluid solver. To account for the retrograde influences the blood has on the myocardial tissue walls, the forces fluid dynamics impose to the local blood pressure are investigated. In this context, the force resulting parallel to the myocardial wall is calculated by the wall shear stress τ_{wss} with the following correlation:

$$\tau_{\mathrm{wss}} = \frac{F_{||}}{A} \,. \tag{14.1}$$

In an ideal fluid, the molecules react instantaneously to an external deflection parallel to the wall by establishing a new rest position. Thus, in an ideal fluid, wall shear stress is assumed to be $\tau_{\mathrm{wss}} = 0$. Even though blood is far from being an ideal fluid, it can be argued that the influence of wall shear stress is small compared to the force resulting transversal to the wall. This can be justified by the fact that the anatomy of the heart chambers as a large cavity does not allow to form a classic pipe flow profile. The blood close to the wall moves and exchanges much less than the blood volume in the central area. This supports the observation that the influence of shear stress is small compared to static pressure and can be neglected.

Thus, based on the results of [150] we assume wall shear stress to be neglectable compared to the internal stresses of the myocardium.

In the current state, the circulatory system only considers a mean pressure value p_{LA}, p_{RA}, p_{LV} and p_{RV} throughout each heart chamber. Local pressure deviations are not considered and get neglected by the averaging. To adequately model the local influence, the pressure field has on the mechanical motion a 3D spatially resolved relative pressure factor r has to be introduced as proposed in [156]. This factor captures the local deviation of the single face compared to the chamber mean value. In each face i, the factor r_i is calculated based on the mean, maximum and minimum value of the chamber pressure p_{c} for each time step j [145]:

$$r_{i,j} = y_{\mathrm{s},j} \cdot \frac{p_{i,j} - p_{\mathrm{mean},j}}{p_{\mathrm{max},j} - p_{\mathrm{min},j}} + 1 \tag{14.2}$$

for each element i at the time $t = j \cdot \Delta t$. The statistical values minimum (p_{min}), maximum (p_{max}) and mean (p_{mean}) pressure are evaluated spatially for the concerning heart chamber at the current time t [145]. An additional scaling factor y_{s} weighs the influence of the relative pressure factor based on the absolute pressure values p in the fluid mechanics simulation.

The subtraction of the mean pressure p_{mean} and the choice of the denominator in equation 14.2 ensures that the mean value of the pressure factor is equal to one. Therefore, the equation

$$\frac{1}{N_c} \sum_{i=1}^{N_c} r_{i,j} = 1 \tag{14.3}$$

is fulfilled for each of the four heart chambers, where N_c is the number of surface elements in the corresponding heart chamber c with $c \in \{\mathrm{LA}, \mathrm{LV}, \mathrm{RA}, \mathrm{RV}\}$. This definition of a mean-free pressure factor is consistent with the circulatory system model, with one average pressure $p_{\mathrm{mech},c}$ for each heart chamber only. Thus, the pressure conditions in the circulatory system model do not get out of balance due to the influence of the fluid solver. For a scaling factor $y_{\mathrm{s}} = 1$, the dimensionless pressure factor r ranges between $r \in (0,2)$, where $r \in (1,2)$ for a fluid pressure higher than the mean pressure provided by the circulatory system.

As a static scaling factor $y_{\mathrm{s}} = \mathrm{const}$, that maps all element pressures p_i on the scaling factor interval $y_{\mathrm{s}} \in (0\ ,\ 2)$ is not capable to fully reproduce the absolute pressure ranges, a time-resolved scaling factor $y_{\mathrm{s}}(t)$ is introduced [145]:

$$y_{\mathrm{s},t} = 2 \cdot \frac{p_{\mathrm{max},t} - p_{\mathrm{mean},t}}{p_{\mathrm{max},t}} \tag{14.4}$$

also based on the statistic quantities of the fluid simulation. This choice ensures that a multiplication of the scaling factor y with the mean pressure p_{mean} reproduces the fluid pressure p_i without changing the mean pressure factor $y_{\mathrm{mean}} = 1$. In order to ensure the robustness of the procedure, the scaling factor is limited to the range $y_{\mathrm{s}} \in (-3,3)$.

Finally, this computed pressure factor r serves as an input to the subsequent mechanical simulation, thus closing the loop of the iterative coupling procedure. The adapted mechanical pressure p^*_{mech} is calculated by a multiplication of the pressure estimated by the circulatory system with the pressure factor r following equation

$$p^*_{\mathrm{mech},i,t} = p_{\mathrm{mech},c,t} \cdot r_{i,t}. \tag{14.5}$$

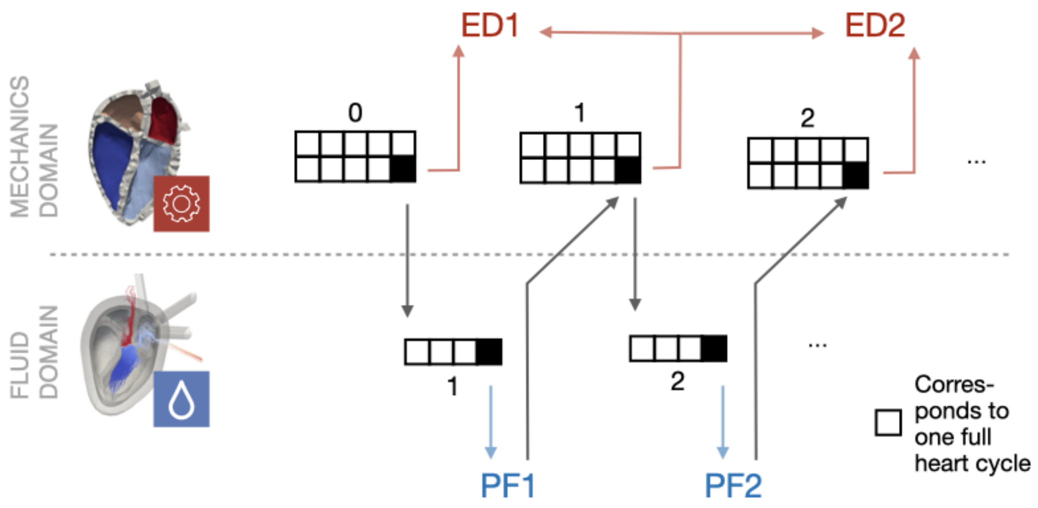


Figure 14.4: Schematic overview of the coupling procedure. Starting with a mechanical simulation, the last heart cycle deformation is used as a boundary condition for the subsequent fluid simulation. The processed PF is used as an input in the next mechanical simulation, when it has reached a steady state. This procedure is repeated. The Euclidean distance (ED) is evaluated between the last cycle of two subsequent mechanical simulations. Figure adapted from [40], published under CC BY 4.0 license.

14.2.5 Fluid-structure coupling

Figure 14.4 shows a schematic overview of the coupling procedure. The initial mechanical simulation is run for ten heart cycles to reach a steady state. The deformation, i.e., the coordinates of the endocardial surface of the last heart cycle is extracted. Also the pressure at the pulmonary veins and the aorta is extracted from the circulatory system for the last heart cycle. These quantities are respected as a boundary condition in the fluid simulation. After four cycles of fluid simulation, a limit cycle is reached. The spatially resolved pressure p_i with $i = 1, \dots, N$ for all N endocardial wall elements from the three-dimensional pressure field p is extracted. For all elements on the endocardial wall, the pressure factor $r(i,t)$ is calculated following equation 14.2. This pressure factor is then respected as a boundary condition in the last cycle of the subsequent solid mechanical simulation. The number of ten mechanics simulations is also applied in this stage to ensure that all transients have decayed. This procedure is applied iteratively. For this study, $N_\mathrm{L} \approx 2{,}200$ endocardial surface elements were evaluated in the left heart ($N_\mathrm{LV} \approx 1{,}500$ and $N_\mathrm{LA} \approx 700$).

14.2.6 Circulatory system model

The model of the circulatory system comprising all four heart chambers and the dynamic model for the valves is implemented as described in section 3.5. To consider the PF r in the computations of the circulatory system plugin, the equations are adapted. A change in the overall structure is not necessary, as $r(i,t)$ was chosen to maintain the mean pressure value in each heart chamber as implied by equation 14.3. In the function, in which previously the chamber pressure was applied to each element during the calculation routine, now a multiplication with the PF $r(i,t)$ based on equation 14.5 is added. To account for the correct PF, the current time step needs to be mapped to the corresponding PF index continuously. For a flexible control, the pressure gradient can be considered for computation by setting a bool variable in the setup configuration.

The parameters are initialized similar as in Gerach et al [25]. However, no static valve model is used. Instead, the valve is modeled by a dynamic approach to adequately capture opening and closing effects like introduced in section 3.5. For the detailed values, see appendix B.

14.2.7 Measures for evaluation

In this section, the three criteria used for quantification and evaluation later on, are introduced. First, the mechanical deformation difference between the different iterations is analyzed. Additionally, a scalar transport equation to evaluate fluid simulations is introduced. Finally, simulation results are evaluated based on pressure volume (pV)-loops. The latter have already been described in section 2.2.

14.2.7.1 Deformation

To evaluate the effect on the mechanical simulation, the displacement of the endocardial wall throughout a heart cycle is analyzed. Therefore the Euclidean distance (ED), d, is calculated:

$$d_{i,j} = \sqrt{(x_{i,j+1} - x_{i,j})^2 + (y_{i,j+1} - y_{i,j})^2 + (z_{i,j+1} - z_{i,j})^2} \tag{14.6}$$

for each node i in time and space, based on the node location (x,y,z) in the two subsequent iterations j and $j+1$. A graphical overview over the iterations $j=0$, $j=1$ and $j=2$ of the mechanical solver computations is given in figure 14.4.

14.2.7.2 Hemodynamics

As a measure for the fluid simulation results with diagnostic value, the mixing procedure of blood in the ventricles is analyzed by solving a scalar transport equation. It is a convection-diffusion equation suggested by [104] and it accounts for the passive transport of the scalar Ψ by the velocity field u. Ψ can therefore be considered as the local concentration of initial blood and enables a tracking of residual blood volume. The passive transport equation

$$\frac{\partial \Psi(x,t)}{\partial t} + U_i(x,t)\frac{\partial \Psi(x,t)}{\partial x_i} - \frac{\partial}{\partial x_i}\left(D\frac{\partial \Psi(x,t)}{\partial x_i}\right) = 0 \tag{14.7}$$

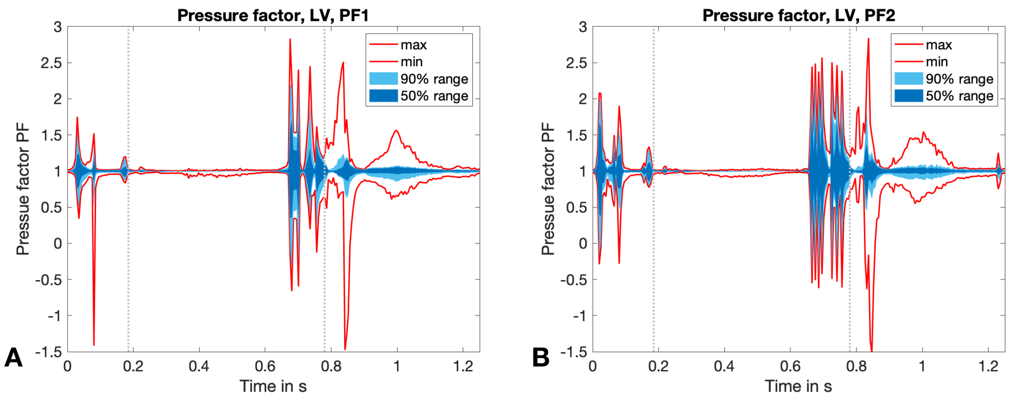


Figure 14.5: Time course of the PF over one heart cycle. Calculated by equation 14.2, it visualizes the influence of the fluid dynamics simulation. Values $PF > 1$ denote a fluid pressure higher than the one calculated by the circulatory system. The statistical distribution of the minimum and the maximum value, as well as the area which covers 50 % and 90 % of all values are highlighted in red as well as dark and light blue color. For temporal orientation, the vertical gray lines mark the start and the end of the systolic phase. Left: First iteration of the fluid solver. Right: Second iteration of the fluid solver. Figure adapted from [40], published under CC BY 4.0 license.

with $i \in \{1,2,3\}$ also comprises an artificial diffusion coefficient $D = 10^{-10}\,\mathrm{m^2 s^{-1}}$ for numerical stability. For $t_0 = 0\,\mathrm{s}$, the scalar Ψ is initialized with

$$\Psi(x,t_0) = \begin{cases} 1 & forallelementsintheLV \\ 0 & \text{elsewhere} \end{cases} \tag{14.8}$$

The distribution of Ψ in the initial state is shown in figure 14.2B.

14.3 Results

Following this sequential coupling approach, three iterations of mechanics simulations and the associated two fluid simulations are simulated like it is visualized in figure 14.4. In this section, the results of the introduced pressure factor as well as the ED as performance measures are presented. Finally, the influences onto the fluid domain as well as the pV-loops are shown.

14.3.1 Pressure factor

To investigate the pressure deviations in the fluid simulation, we evaluated the time course of the normalized pressure factor in figure 14.5. The first fluid simulation is shown in figure 14.5A, the second iteration in figure 14.5B.

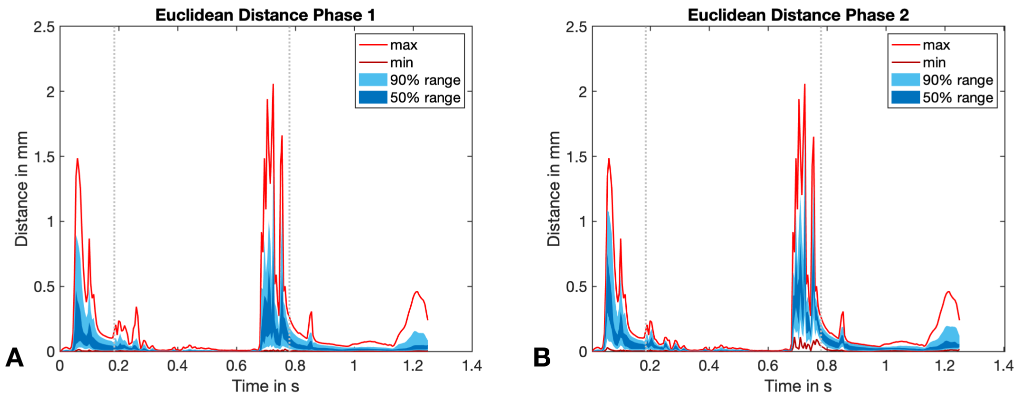


Figure 14.6: Time course of ED for the last heart cycle in every iteration of the mechanical solver. For an overview see again figure 14.4. The statistical distribution of the minimum and the maximum value, as well as the area which covers 50 % and 90 % of all values are highlighted in red as well as dark and light blue color. A: ED0 - ED between initial mechanical simulation (j=0) and subsequent iteration (mechanical simulation 1). B: ED1 - ED between mechanical simulation 1 (j=1) and subsequent mechanical iteration. See figure 14.4 for an overview. Figure adapted from [40], published under CC BY 4.0 license.

The PF is in the range of $-1.5 < PF < 3$ for both simulations. The mean value is $PF_{\text{mean}} = 1$ per definition, but also the 50 % quantile is pretty close to $PF_{\text{mean}} = 1$ throughout most of the time. At two characteristic points of the heart cycle, higher deviations can clearly be observed: On the one hand, at the beginning of the heart cycle ($t = 0.1$ s) and on the other hand during the ventricular systole ($t = 0.7$ s). The highest deviations occur during systole close to the aortic valve.

If we investigate a spatially resolved map of the pressure factor, it is clearly visible that the large pressure deviations occur close to the aortic valve, where blood is ejected during systole. Thus, a pressure higher than the mean pressure of the chamber (which is the one corresponding to the boundary conditions from the mechanical simulation) is present. This is also accompanied by the spatial distribution of the Euclidean distance in figure 14.7.

While comparing the two fluid iterations shown in the overview figure 14.4, the basic shape of the PF time course stays equal throughout the plots of figure 14.5A and figure 14.5B. On the one hand, this confirms a consistent behavior of the coupling algorithm: The timing of the coupling framework matches the two single physics solvers. On the other hand the reduced amplitude of the pressure factor by around 10 % shows the tendency to converge towards a limit cycle.

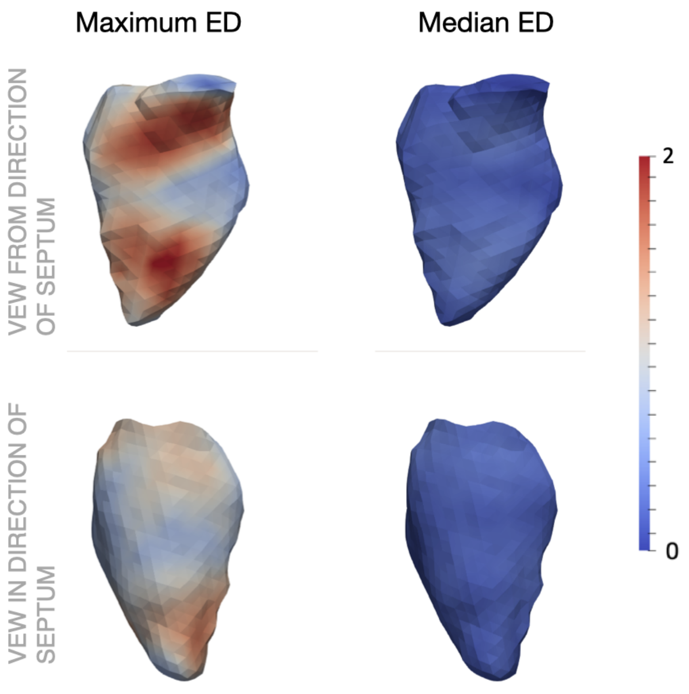


Figure 14.7: Spatial distribution of ED across the left ventricle. The first column shows the maximum values, the second column the median values. In the upper row, the ventricle is visualized from a septal position, the bottom row shoes the opposite view. ED between the initial mechanical simulation (j =0) and the subsequential one. Figure adapted from [40], published under CC BY 4.0 license.

14.3.2 Deformation

In the second step the ED between the corresponding nodes of the mechanics simulation in subsequent iterations is evaluated. The time course of the overall ED for the last heart cycle is shown in figure 14.6.

For the left ventricle (left ventricle (LV)), the maximal ED of all endocardial wall nodes was smaller than 2 mm between the first and second iteration. The maximal distance between the second and third iteration was 70 µm, thus the limit of necessary cycles was already reached after two iterations.

The spatial distribution of the ED throughout the LV is shown in figure 14.7.

In the left part, the maximum ED is shown from the direction of the septum (upper part), as well as from the opposite side (lower part). The maximum deviation clearly occurs in the area of the aortic valve and close to the apex. These two locations correspond to the two characteristic points in the cardiac cycle: While blood flows into the ventricle during atrial contraction, the jet towards the apex increases the pressure. In turn, during ventricular systole, blood moves towards the aorta and increases the pressure in this area.

Comparing to the right part of figure 14.7 reveals that the median ED is distributed uniformly across the LV and is in the range below 1 mm. Minimum, maximum and mean

Euclidean distance ED	left ventricle (LV)			left atrium (LA)			overall		
in mm	min	max	mean	min	max	mean	min	max	mean
iteration 0/1 (j=0 / ED0)	0.00	2.07	0.06	0.00	2.05	0.03	0.00	2.07	0.04
iteration 1/2 (j=1 / ED1)	0.00	2.06	0.06	0.00	1.66	0.03	0.00	2.06	0.04

Table 14.2: Euclidean distance in mm. It is calculated as the deviation of a node comparing two subsequent iterations of the mechanical solver. In this case, the ED between iteration 0 and 1 (j=0) as well as the ED between iteration 1 and 2 (j=1) are given and quantified for the three regions LV, right ventricle (RV) and the overall geometry. Minimum, mean and maximum values are listed. Table adapted from [40], published under CC BY 4.0 license.

absolute numbers for the ED are listed for two iterations in Table 14.2. These results show that the changes in deformation as a reaction to the pressure gradient distribution are small but still exist. Comparing the right part of the time course in figure 14.6 reveals an extremely small difference between the ED of the first (iteration 0/1) and the following iteration. With a deviation that is two orders of magnitude smaller than the ED itself, it can be stated that a limit cycle is already reached after one iteration of the fluid solver.

14.3.3 Hemodynamics

To evaluate the effects of the coupling on the fluid dynamic quantities, the residual volume V_{res} in the left ventricle is investigated. The time course of four heart cycles is shown in figure 14.8. Hereby, the residual volume is calculated by a multiplication of the scalar Ψ computed by the passive scalar transport (equation 14.7). Therefore, the blood volume that initially (at $t = 0\,\mathrm{s}$) started in the LV and still remains there is depicted in milliliters. If the influence of the pressure factor is not yet considered (fluid simulation phase 1), a residual volume $V_{\mathrm{res},1} \approx 20\,\mathrm{ml}$ is achieved. Considering the influence of the pressure factor, a residual volume of $V_{\mathrm{res},2} \approx 16\,\mathrm{ml}$ results. So to sum up, a difference of $\Delta V_{\mathrm{res}} = 4\,\mathrm{ml}$, compared with a $V_{\mathrm{ED}} \approx 180\,\mathrm{ml}$ describes a deviation of approximately 2 %. Therefore, the slight changes in deformation of the boundary surface reveal a small but not neglectable accumulated residual volume. Consideration of the coupling algorithm thus increases the influence of the washout process in the ventricle.

Figure 14.9 additionally presents the spatial distribution of the scalar field Ψ for three different consecutive iterations of the fluid dynamics solver. As visualized in the schematic overview (figure 14.4), fluid iteration 1 which is shown on the left hand side of the figure does not yet consider influences of the local pressure field. The first pressure factor $PF1$ is considered in the consecutive iteration 2. In the long axis cuts of the geometry, flow patterns inside the left ventricle can be compared. The point in time $t = 1.1$ s during the relaxation phase (compared to the whole cycle length of $t = 1.247\,\mathrm{s}$) is chosen closely around the diastatic state with a fully relaxed mechanical state. On the other hand, in comparison with figure 14.8, the statistical value of the overall residual volume V_{res} indicates a completely matching value in both iterations based on the evaluations of the scalar field Ψ. However, the spatial distribution of Ψ indicates, that even with an exactly matching residual volume

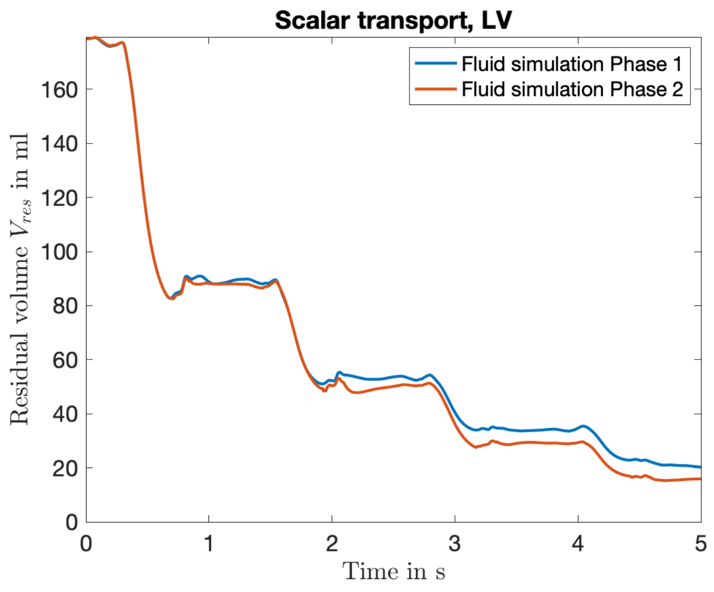


Figure 14.8: Time course of the residual volume introduced to quantify the washout during the fluid simulation. It is computed based on the transport of the scalar Ψ, which can be seen as a local concentration of initial blood volume in the LV. Shown is the curve for the four full heart cycles, color coded are the two iterations of the fluid solver. Figure adapted from [40], published under CC BY 4.0 license.

V_{res}, the flow patterns as well as the ventricular washout differ. Compared to the second fluid dynamics iteration, the jet towards the apex is qualitatively lower in the first iteration. This reveals an influence of the sequential coupling algorithm and shows an underestimation of the washout if the coupling is not considered. Comparing the second to the third iteration again confirms the analyses from the mechanical results: The deviations in the flow patterns are not significant any more. This observation can also be confirmed throughout the following time steps and highlights that the influence of the coupling algorithm is already completely incorporated after one iteration and the limit cycle is reached.

14.3.4 Pressure-volume-loop

PV-loops are often used to represent the work and efficiency of a mechanical system. The same holds true for the human heart: The pV-loops in figure 14.10 show the mean ventricular chamber pressure plotted against the ventricular chamber volume and reveal the influence of our coupling framework. As the mean pressure of each chamber $p_{\text{mean},c}$ is per definition kept constant (equation 14.3), deviations in the pV-loop are only caused by an altered chamber volume. Therefore, the changes in the diagram are not that great. However, some deviations exist and a low alteration of the chamber volume can be found. While the blue line shows the resulting pV-loop after the nine heart cycles of mechanical simulations (limit cycle, compare overview in figure 14.4), the red and green lines show the result of considering the pressure

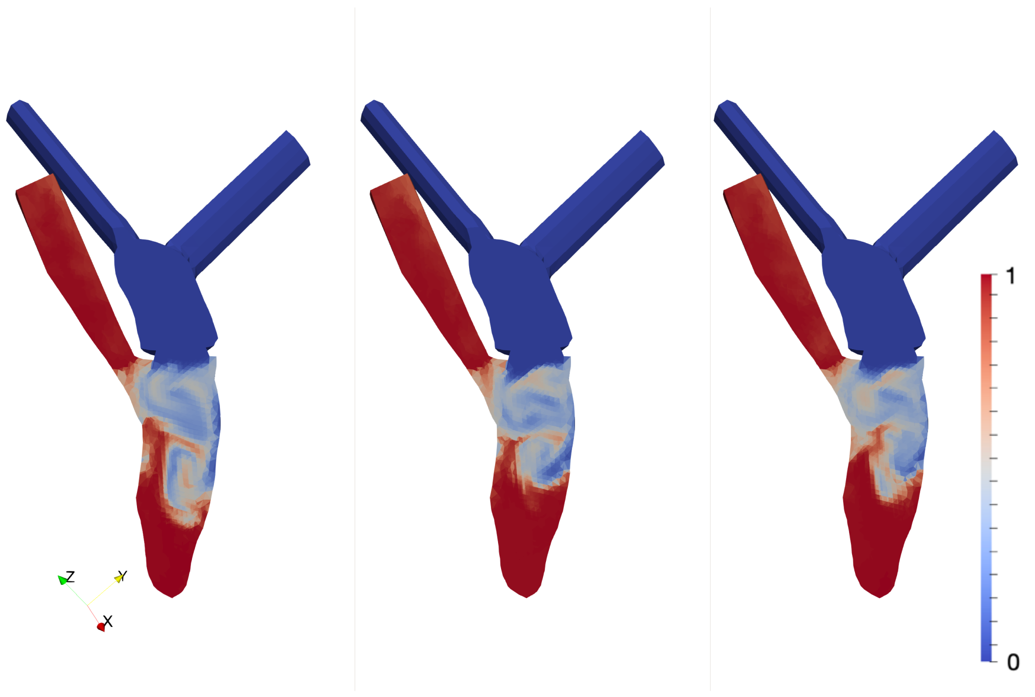


Figure 14.9: Long axis cut through the fluid simulation mesh. Color coded the spatial distribution of the scalar field Ψ during the ventricular relaxation period at the time $t = 1.1\,\mathrm{s}$. From left to right, three iterations of fluid simulation results are included (fluid iteration 1 at the left hand side, for indexing also compare figure 14.4). While between the first and the second iteration, a slightly altered flow pattern and a reduced jet to the apex can be observed, the comparison between iteration 2 and 3 reveals less differences.

factor. Deviations can clearly be observed in the lower left part of the curve. At this point in time, the isovolumetric relaxation (left vertical line) and the diastolic filling (lower horizontal line) take place.

Comparing the pV-loops of the two iterations again confirms the convergence of the system towards a limit cycle: Deviations in the chamber volume become smaller. Furthermore, it can be recognized that the deviations in the mean chamber volume are below $\Delta V_c < 1\,\mathrm{ml}$ at every point in time.

However, while comparing the initial iteration (blue) to the first (red) and second (green) also reveals differences: On the one hand, at the end of the systole, a pressure peak can be observed in the top left area of the loop. On the other hand, especially in the lower and left part of the loop, oscillations are introduced by the iterative coupling procedure. However, we observe that the amplitudes of both artifacts decrease in the second iteration.

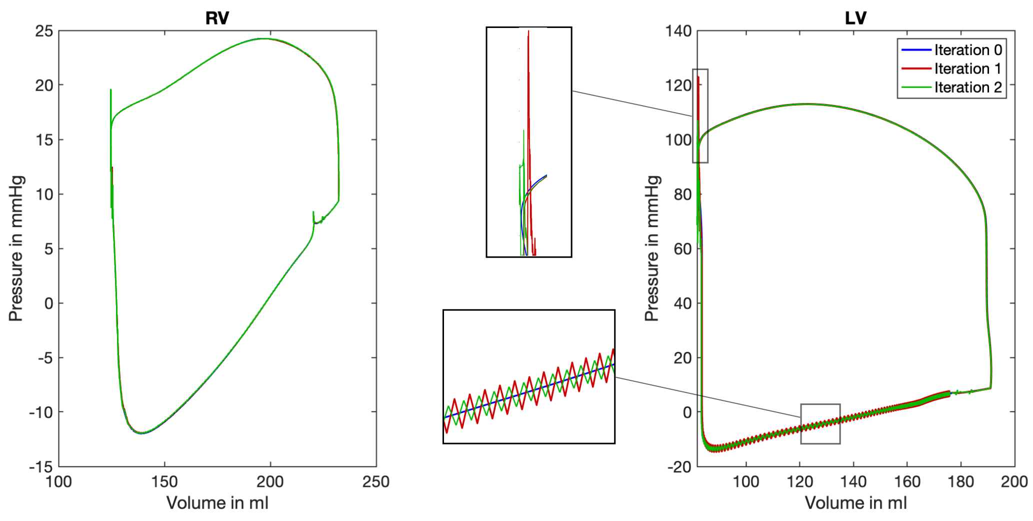


Figure 14.10: PV-loop for the right (right ventricle (RV)) and left LV ventricle. Color-coded are the three iterations of the mechanical solver while the first iteration depicts the limit cycle after nine iterations of the mechanical solver (mechanical simulation 0, blue color). For the LV, two characteristic areas are shown enlarged in the center of the figure to highlight the differences between the three iterations. Figure adapted from [40], published under CC BY 4.0 license.

14.4 Discussion

The novel coupling approach combines realistic feedback with comparably low computational effort and independence of the two single area of physics solvers. The coupling interface can be used independently from the exact solver architecture and therefore offers a high degree of flexibility and applicability. It unites simulation software packages focusing on only one area of physics to exchange relevant information in a bidirectional manner. In this way, we enable to consider fluid dynamical influences in structural mechanics simulations without a need of direct interaction during the simulation itself.

The main task and also the biggest challenge that partitioned FSI solvers face is the adequate exchange of coupling data. In most cases, different underlying mesh geometries have to be compared and resulting quantities have to be mapped and interpolated onto the concerning meshes. This process is highly dependent on the accuracy of all models, as well as a correct mapping. Therefore, coupling frameworks are prone to introduce errors [46]. Since in our approach, the solvers do not exchange information in each time step Δt, the correct temporal alignment of the simulation results is of high importance as well. As we experienced during testing, already a small shift in the alignment of the timing can cause simulations not to converge.

When comparing the novel approach with existing approaches in literature, we can find that partitioned approaches continue to gain in importance. The group around [147] implemented a monolithic solver to tackle the multi-physics problem. To account for the influences of the fluid simulation in the mechanics domain, a fluid Cauchy stress tensor is considered. Thus, in each time step, fluid and mechanics equations are solved within one single solver. Like in this work, the Navier Stokes equation (NSE) are formulated in ALE description. However, when using measurement data as a boundary condition for fluid simulation in the application part, the motion equations are no longer solved. Based on the measured time courses, the deformation of the myocardial walls is known a priori and is not computed based on the monolithic equations any more. This approach still delivered accurate results. Lately, [24] additionally implemented an implicit-explicit scheme based on an inter-grid transfer operator to enable the computation using different mesh resolutions and discretizations in space and time. Their coupling scheme is therefore segregated and staggered. In the fully coupled model suggested by [28], a partitioned FSI simulation was used. They tackled the multi-physics problem of cardiac simulations by a division in two domains with a Dirichlet-Neumann decomposition each. Based on a strongly coupled staggered approach, the electro-mechanical domain is bidirectionally coupled to the fluid domain. A fluid-structure interface is implemented to compute the discrete position of the nodes based on [51]. Also [53] presented a multi-way coupled computational model based on an immersed boundary grid for the left part of the heart. In their fluid-structure-electrophysiology coupling approach, a central structure evolution equation is solved to link between the three named physical domains. So in each time step, the new mechanical geometry is used as an input for the three single area of physics solvers. The approach was found to deliver accurate results on an idealized left heart geometry. Another approach suggested by [157] is also based on a partitioned solver. A block Gauss–Seidel implicit scheme is used, but there the three domains fluid, mechanics and mesh motion are considered. Overall, there are many FSI approaches and applications, with most of them based on a partitioned approach. The review of [46] concisely presents a large number in their corresponding application fields.

Summing up, all approaches listed in this section are based on a staggered scheme. Thus, our approach is based on the same basic decomposition, that is used in established processes. However, as we suggest a cycle to cycle coupling, the effort of implementing an elaborated coupling scheme, that closely couples both areas of physics, is circumvented.

An abstract way to model the influence of the circulatory system is to use a 3D-0D coupled approach [25, 31, 158, 159]. To adequately define the interface between the circulatory system pressures and the fluid simulation boundary pressures, we introduced the prolonged trunks in the left heart to model the pulmonary veins and the aorta. These rigid tubes also allow for steady state flow conditions.

To replicate realistic blood flow patterns, [38] showed that a comparably simple valve model can be sufficient. Inserting a tripping ring along the valve annulus enhances vortex formation such that the typical vortex ring develops in the left ventricular chamber. In this way, previous limitations of the comparably simple valve model can be overcome [160, 161]. In the present study, the valve planes between atria and ventricles incorporate a notable narrowing, which already represents the effective blockage of the valve leaflets considered sufficient to demonstrate the basic performance of the coupling algorithm. The same holds true for the limitation already discussed by [38]: This highly efficient numerical representation of the mitral valve (MV) may not be sufficient to study MV flow characteristics or MV pathology in detail.

In hemorheology studies, the characteristics of the blood are investigated. It is a matter of debate whether blood can be modeled as a Newtonian fluid, such that the Navier stokes equations as presented above hold true. As for example [162] state, length and time scales are sufficiently large in macroscopic views of the heart chambers, such that the continuum hypothesis holds true. Therefore, we model blood as a Newtonian fluid in this study. However, non-Newtonian effects could potentially play a role in the prolonged trunks [163].

Considering the scalar transport equation introduced in section 14.2.7.2, the consideration of an artificial diffusion factor D may have an impact on the resulting flow quantities. However, as the factor is on a really small scale ($\propto 1 \cdot 10^{-9}\,\mathrm{m^2s^{-1}}$), the impact on the resulting flow and pressure field is not significant.

The results presented in this study show that the sequential coupling approach can be successfully applied for a healthy heart model. In particular, the Euclidean distance converged for successive mechanical iterations. A limit cycle was reached after already one iteration and oscillations decreased significantly.

The numerical stability of the novel approach was high for the investigated heart model. However, as mentioned in section 14.2.4, the pressure factor has an impact on the absolute pressure values and the absolute deviations. Thus, a non-limited scaling factor could lead to instabilities of the mechanical simulation. Especially if the fluid and the mechanical simulation differ strongly, e.g., caused by a time offset, numerical stability can be impaired. The oscillations appearing in the pV-loop are probably introduced by the application of the pressure factor: A steady state system that is exposed to a pressure step, in our case the multiplication with a pressure factor, is brought out of balance and starts to oscillate. To cope with this scenario, normally several iterations are computed, until a new steady state is reached. As in our case the circulatory system does not compute several iterations, the discontinuity of the multiplied pressure factor (pressure step) has to be handled. We observed that oscillations decreased in a subsequent coupling iteration. Pressure discontinuities in the fluid simulation that arise locally, i.e., in neighboring elements could lead to instabilities in the subsequent mechanical simulation. These local pressure discontinuities could for example be caused by a wrong labeling of cells: As the pressure factor is computed based on the mean pressure in each chamber, the correct classification of cells is of high importance, especially at the edges of neighboring regions.

With a mean ED of 0.06 mm in the LV when comparing the initial (iteration 0) and the first iteration, the tissue deformation resulting from the fluid pressure field is considerably low. Especially compared to the uncertainty that is introduced during the imaging process, during image segmentation and during mesh creation, the mean deviation is neglectable. With the same reasoning, the resulting volume alteration of below 1 ml can be neglected compared to the overall chamber volumes. Thus, the additional effort of a fully coupled simulation is likely not justified. Nevertheless, it can be argued that the maximum geometrical alteration of up to 2 mm is in the same order of magnitude as the imaging uncertainty. However, the maximum deviation is limited to a very small number of nodes in time and space. In the regions around the aorta and the apex, fluid dynamics had a higher influence on the model displacement at characteristic points in time (figure 14.6).

We contextualize these deviations with an increased blood flow velocity. As increased blood flow results in a lower static wall pressure, it deflects the wall to a lower extent than areas in a mean flow range. This phenomenon is expected to occur in the area around the aorta during ventricular systole, when the blood is ejected out of the LV. As we observe high blood velocities (up to 200 ml/s) in the aortic region, we would expect the fluid to have an influence on the mechanical deformation through FSI. During the ventricular filling phase, the area around the apex is affected by this phenomenon. As the scalar transport confirms, there is a jet towards the apex. This jet also accounts for a local velocity increase, resulting in a higher ED comparing iterations 0 and 1.

Thus, if the focus of interest is exactly in the detailed resolution of these regions, the impact of fluid dynamics has to be considered especially at the mentioned characteristic points in time. However, even in this case, it is sufficient to compute one iteration of the sequential pressure-deviation coupling algorithm.

Considering the washout of the LV in figure 14.8, the altered mechanical deformation leads to a noticeable increase. This is due to the fact that residual volume is summed up cycle by cycle, which finally leads to an increased washout of 4 %. Thus, in the case of simulating subsequent heart cycles, considering one iteration of the sequential coupling procedure would lead to more precise results.

In conclusion, the retrograde influence of fluid dynamics on mechanical deformation appears as not particularly relevant for most of the time of the heart cycle in a healthy configuration. In other cases, one iteration of the fluid-mechanic coupling would be sufficient to account for the minor deviations.

Regarding other coupling approaches, the main advantage of the sequential coupling approach is the independence of the mechanics and fluid solvers. The presented approach enables to use different, completely independent software packages for the simulation of the different areas of physics. The solvers are only linked by exchanging the pressure factor and the only requirement for the mechanical solver is the ability to incorporate the external pressure factor during simulation.

In future work, this iterative coupling approach will have to prove its ability to deliver accurate results also for diseased heart models. To improve stability of the coupling framework and to increase smoothness of the results, a time window for the determination of the scaling factor y_s can be introduced. Additionally, the degree of automation can be increased towards a fully automatic pipeline.

Summing up, we observed that the influence of fluid dynamic pressure resulted in a structural deviation of up to 2 mm. These deviations could already be resolved after one coupling iteration. In conclusion, the sequential coupling approach with its comparably low computational effort delivered promising results for modeling fluid-structure interaction in cardiac simulations.

PART VII

FINAL REMARKS

Chapter 15

Outlook and conclusions

15.1 Outlook

To accurately reproduce quantitative values obtained from simulations based on body-fitted mesh approaches with the immersed boundary method (IBM) solver, the accordance between the solvers for the different areas of physics has to be increased. More concretely, the representation of the valve dynamics revealed to have a high influence on blood flow dynamics and introduced deviations. Thus, either the valve geometry can further be improved by modeling diameter and opening times based on the circulatory system parameters. On the other hand, the circulatory system parameters can be adjusted in accordance with the valve geometry.

While the acquired 4D flow measurements only allowed for a basic qualitative validation of the IBM solver, an additional study has to be set up to quantitatively validate the solver results. However, the acquisition of a complete data set comprising electrophysiology, elastomechanical and fluid dynamics data of the same heart geometry would form a thorough base for the validation of the elastomechanics and the fluid dynamics solver. This would also be a necessary requirement for a potential use in the clinically relevant context.

Investigating further application fields of the solver, of course the direct incorporation of measured data as the constraining endocardial surface would broaden potential application areas. Under consideration of the necessary match between the solvers for the different areas of physics, also further relevant diseases can assessed. Closing the loop of the multi-physics modeling, also a coupling to the cardiac electrophysiology is conceivable to directly model the influence of alterations in atrial electrophysiology on the resulting flow patterns.

Evaluating the influences of the fluid dynamics simulations on the elastomechanical deformations with the sequential coupling algorithm revealed a deviation in a low range. In addition, already after one iteration, all influences had been balanced and no further deviations in the displacement occurred. Therefore, it was concluded that this implies that already one iteration of the sequential coupling process is sufficient to model all retrograde effects. However, to really prove that unidirectional coupling is sufficient, a fully coupled solver would be needed as a reliable, ground truth reference solution.

15.2 Conclusions

In the scope of this thesis, two novel approaches in the context of light coupling strategies to simulate the fluid dynamics in the human heart were proposed and assessed concerning adequate functionality. While verifying and evaluating both of them, possibilities and limitations of the light coupling methods were carved out.

At first, a unidirectionally coupled Finite Volume (FV) solver based on an IBM approach was introduced. The movement of the endocardial surfaces was prescribed by applying external surface files, boundary conditions (BC)s as pressure values and flow rates were extracted from the circulatory system framework. Simulations showed to basically be capable to deliver results with a similar accuracy than the already implemented body-fitted approach. However, simulations also revealed a clear challenge in unidirectionally coupled solver frameworks: While both solver for the different areas of physics have to be matched correctly to not introduce deviations due to differences in the models. This was found to be especially a challenge in the context of the correct representation of the geometric valve shape and time control.

Applying the fluid dynamics solver to diseased heart geometries, flow quantities were computed under pathological conditions. Fluid simulations revealed alterations in the flow conditions in both considered pathologies, in patients suffering from hypertrophic cardiomyopathy as well as in patients with mitral valve regurgitation. Applying a particle tracking algorithm to post process the simulated data set, flow components were distinguished.

Focusing on the interdependence between the elastomechanical and the fluid dynamics solver, the retrograde effect on the elastomechanical simulation were assessed. A sequential coupling algorithm was introduced to hand over fluid simulation quantities to the elastomechanics solver in a bidirectionally coupled manner. It was found, that the introduced deviation is in the millimeter range and thus considerably low. The bidirectional exchange of information did not show a significant impact on the resulting flow patterns. Therefore, it was not mandatory to consider the bidirectional exchange of information in the considered healthy heart geometry. However, when sequential coupling is considered, the effects are already balanced after only one iteration.

Summarized, light coupling strategies have to be used with caution and only for cases, in which an appropriate accordance between the involved solvers for the different areas of physics is ensured. If this condition is fulfilled, a significant reduce in computational effort and in the exchange of data between the solvers can be achieved. Based on the results of the studies in the scope of this thesis, the following conclusions can be drawn:

The IBM implementation proved to be a reliable method to model a moving boundary. Thus, it successfully can be applied to simulate blood flow dynamics in the heart. While the method was verified on various simplified scenarios, it revealed a good accordance with the compared analytical solution: The boundary layer, as well as flow quantities in 2D patch simulations were adequately reproduced.

The missing feedback loop between the fluid dynamics and the elastomechanics solver in unidirectionally coupled frameworks showed to increase the influence of the BC on the fluid simulation results. Exemplary, the representation of the valves had a high impact on flow quantities: While the volume change is prescribed by the results of the circulatory system, an alteration in the valve diameter leads to an increased pressure drop across the valve diameter. Taking this into account, light coupling strategies can only be applied, if the different area of physics solvers are correctly matched.

Fluid dynamic solvers based on a unidirectional coupling approach can deliver valuable insights into the flow conditions in pathological heart geometries. Flow patterns can be visualized, flow components can be distinguished and quantified and also measures relevant in the clinical context can be assessed.

A sequential coupling algorithm can be applied to consider the influences of the fluid dynamics simulations in the elastomechanical simulation framework. This bidirectional coupling approach leads to a balanced state in only one iteration. Considering the influences of the spatially resolved pressure gradient field resulting from the fluid dynamics simulation showed a limited impact on the elastomechanical simulation results. Differences vanished after one iteration of the cycle-to-cycle based approach.

PART VIII

APPENDIX

Appendix A

Trunk prolongation script

The following MATLAB scripts are used to automatically compute the heart geometry with prolonged inlet and outlet trunks as introduced in chapter 5.
extractFluidSurfaces.m:

```
% extractFluidSurfaces.m
% Read mechanic simuation result, extract surfaces for fluid ...
    simulation and save stl files.
% Works for T4 and T10 input files - T10 geometries are converted ...
    to T4.
% Input: [req].vtu file of CardioMechancis Simulation result
% Output: .stl file of LA, LV, RA, RV, LAV, RAV, shortTrunks
% jb072, 14.01.2022

%% clear
clear
close all
clc

%% Input data
mfile='/Users/.../Baseline.vtu'; %mechanics input file
fpath='/Users/.../fluidGeo/'; %fluid output folder

%% Material indexes
names=[{'LA'}, {'LV'}, {'RA'}, {'RV'}, {'LAV'}, {'RAV'}, ...
    {'tinyTrunks'}, {'shortTrunks'}];
mats=[{133}, {131}, {132}, {130}, {37}, {36}, {38},{[38, 39]}]; ...
                % earlier (in Ekis simulations)
mats=[{133}, {131}, {132}, {130}, {37}, {36}, {[38, 35]},{[38, 39, ...
    35]}];    % now     (in Tobis simulations)
mats=[{133}, {131}, {132}, {130}, {37}, {36}, {[38, 233]},{[38, ...
    39, 233]}]; % now     (in fluid simulations)

%% Read mechanical geometry
m=readVTK(mfile);

```

```
%% Extract cells by material and export .stl surfaces
for i=1:1:size(mats,2)
    %extract cells by material
    [cells, points, surf_e]=extract_cellByMaterial(m, mats{i});

    % Export stl surfaces
    stlwrite(surf_e,[fpath names{i} '.stl'])
end

disp(['Exported stl file of different surfaces relevant for fluid ...
    simulation to ' fpath])

%% End of pipeline



%% FUNCTIONS

function [ST_points_new, ST_cells_new] = ...
    delete_RemainingPoints(ST_points, ST_cells_new)
% Delete points that are not longer needed and shift cell IDs
    % identify outdated points
    todelete=[];
    for i=size(ST_points,1):-1:1
        if isempty(find(ST_cells_new(:,:)==i))
            todelete=[todelete;i];
        end
    end

    % shift pointIDs and delete outdated points
    ST_points_new=ST_points;
    for i=1:1:size(todelete,1)
       ST_cells_new = ST_cells_new - double(ST_cells_new>todelete(i));
       ST_points_new(todelete(i),:)=[]; %delete points inside the ...
           (deleted) surface
    end

end

function [cells, points, surf_e]=extract_cellByMaterial(m, mat)
% extract all cells with corresponding material and return ...
    triangulation
cells=[];
for i=1:1:size(mat,2)
    idx_c=(m.cellData.Material==mat(i));
    cells=[cells; double(m.cells(idx_c,:))];
end

%check if surface or volume elements
if sum(cells(:,4)≠0) >0
    terminal=zeros(1,size(cells,2)-3); %fill up with zeros (1x for ...
        T4, 7x for T10)
    for i=1:1:size(cells,1)
        if cells(i,4)≠0
            cells(end+1,:)=[cells(i,2) cells(i,3) cells(i,4) ...
                terminal];
            cells(end+1,:)=[cells(i,1) cells(i,2) cells(i,4) ...
                terminal];
            cells(end+1,:)=[cells(i,1) cells(i,3) cells(i,4) ...
                terminal];
        end
    end
end

cells(:,4:end)=[];

%extract surface - remove internal faces (caused by volume cells ...
    or 2 materials)
for i=size(cells,1):-1:1
    [b,¬]=find(cells(i,1)==cells);
    [c,¬]=find(cells(i,2)==cells(b,:));
    [d,¬]=find(cells(i,3)==cells(b(c),:));
    if size(d,1)>1
        cells(b(c(d(1))),:)=[NaN NaN NaN];
        cells(b(c(d(2))),:)=[NaN NaN NaN];
    end
end

idx_k=¬isnan(cells(:,1));
cells=cells(idx_k,:);

% Check if T10. If yes, convert to T4
if size(m.cells,2)==10
    allcells=double(m.cells);
    idxT10=min(allcells(allcells(:,5)>0,5)); %Minimum value ...
        greater than 0
    centrals=cells(sum(cells≥idxT10,2)==3,:); %list central T10 ...
        surface element
    if ¬isempty(centrals)
        cells_init=cells;
        cells=[];

        for i=1:1:size(centrals,1)
            [b,¬]=find(centrals(i,1)==cells_init);
            [c,¬]=find(centrals(i,2)==cells_init(b,:));
            [d,¬]=find(centrals(i,3)==cells_init(b,:));
            [e,¬]=find(centrals(i,2)==cells_init);
            [f,¬]=find(centrals(i,3)==cells_init(e,:));
            allPoints=[cells_init(b(c),:) ; cells_init(b(d),:) ; ...
                cells_init(e(f),:)];
            cells=[cells;(allPoints(allPoints<idxT10))'];
        end
    end
else
```

```
    idxT10=[];
end

% triangulate
if ¬isempty(idxT10)
    m.points(idxT10:end,:)=[];
end

[points, cells] = delete_RemainingPoints(double(m.points), cells);
surf_e=triangulation(cells, points);
%trisurf(surf_e)
end
```

prolongTrunks.m:

```
% prolongTrunks.m
% Read heart geometry, prolong trunks and save stl files.
% Input: [req] .stl file of short trunks from mechanical mesh ...
    (material 38+39+(35/34))
% Input: [req] .stl file of LV (material 131)
% Input: [req] .stl file of LA (material 133)
% Input: [req] .stl file of RV (material 130)
% Input: [req] .stl file of RA (material 132)
% Input: [req] .stl file of LAV or RAV plane (material 37 or 36)
% Input: [req] left or right heart (default isleft=true))
% Input: [opt] length of trunks (default length=10 [times avgLength])
% Input: [opt] .stl file of tiny trunks from mechanical mesh ...
    (material 38)
% Input: [opt] delPoints: List of nodes. For each node, del. ...
    connected cells
% Input: [opt] path to .stl input files
% Output: .stl file of prolonged trunks (whole heart side geometry)
% Output: .stl file of pressure inlet
% Output: .stl file of pressure outlet
% jb072, 04.01.2022

%% clear
clear
close all
clc

%% Input data
inpath='/Users/.../fluidGeo';
isleft=true;
length=12; %multiple of avgEdgeLength, with which the trunks are ...
    prolonged.

delPoints=[];
% delPoints=[0.01,0.01,0.01]; %if there are faulty tets in the ...
    mechanical mesh, list the top point here to remove connected ...
    cells.

%% Output folder
if ¬exist([inpath '/left'], 'dir')
   mkdir([inpath '/left'])
end
if ¬exist([inpath '/right'], 'dir')
   mkdir([inpath '/right'])
end

%% Read geometry
RV=stlread([inpath,'/RV.stl']);
LV=stlread([inpath,'/LV.stl']);
RA=stlread([inpath,'/RA.stl']);
```

```
RAV=stlread([inpath,'/RAV.stl']);
LA=stlread([inpath,'/LA.stl']);
LAV=stlread([inpath,'/LAV.stl']);
ST=stlread([inpath,'/shortTrunks.stl']); %shortTrunks
if isfile([inpath,'/tinyTrunks.stl'])
    TT=stlread([inpath,'/tinyTrunks.stl']); %tinyTrunks
end

ST_i=triangulation(ST.faces, ST.vertices);


%% If left heart, mark right heart trunk for deletion and other ...
    way around
if isleft
    V=LV; %Ventricle = left ventricle
    delV=RV; %RV to delete
    A=LA;
    delA=RA;
    AV=LAV;
else
    V=RV; %Ventricle = right ventricle
    delV=LV;
    A=RA;
    delA=LA;
    AV=RAV;
end

%% visualization
% V_t=triangulation(V.faces, V.vertices);
%
% figure
% trisurf(V_t)
% hold on
% trisurf(ST_i)

%% Preprocess: Delete points that appear multiple times
% First: Reduce points that appear multiple times
[V_points,V_cells] = reduce_MultiplePoints(V);
[A_points,A_cells] = reduce_MultiplePoints(A);
[ST_points,ST_cells] = reduce_MultiplePoints(ST);
[delV_points,delV_cells] = reduce_MultiplePoints(delV);
[delA_points,delA_cells] = reduce_MultiplePoints(delA);
[AV_points,AV_cells] = reduce_MultiplePoints(AV);

%% Delete unneeded trunks
%Ventricle
[ST_points, ST_cells, ST_u] = delete_AllConnectedCells(ST_points, ...
    ST_cells, delV_points);
%trisurf(ST_u)

%Atrium
```

```
[ST_points, ST_cells, ST_u] = delete_AllConnectedCells(ST_points, ...
    ST_cells, delA_points);
%trisurf(ST_u)

%% Process ventricular trunks
% Delete shared surfaces ventricle
[ST_points_new, ST_cells_new, V_points, V_cells, ...
    edge_Points_v,deletedV]=delete_sharedSurfaces(ST_points, ...
    ST_cells, V_points, V_cells);

% Prolong ventricular trunks
[v_trunks_c, v_trunks_p,LT_VT, v_lids_c, ...
    v_lids_p]=prolong_chamberTrunks(edge_Points_v, ST_cells_new, ...
    ST_points_new, length);

%triangulate and export lid
ST_VL=triangulation(v_lids_c{1}, v_lids_p{1});

if isleft
    stlwrite(ST_VL,[inpath,'/left/outlet.stl'])
    disp(['Exported stl file of outlet geometry to ' inpath ...
        '/left/outlet.stl'])
else
    stlwrite(ST_VL,[inpath,'/right/outlet.stl'])
    disp(['Exported stl file of outlet geometry to ' inpath ...
        '/right/outlet.stl'])
end

%% If available: Process tiny trunks and export (for volumeMesh ...
    creation)
% Extract semilunar valve (SLV): aortic/pulmonary valve
if exist('TT','var')

    % First: Reduce points that appear multiple times
    [TT_points,TT_cells] = reduce_MultiplePoints(TT);

    % Find common points with ventricular trunk
    [match] = find_CommonPoints(TT_points, v_trunks_p{1});

    %Extract all connected cells
    [connected_faces,rest,found_edge_Points] = ...
        identify_ConnectedCells(TT_cells, match(:,1), false);
    [TT_points_new, TT_cells_new] = ...
        delete_RemainingPoints(TT_points, ...
        TT_cells(connected_faces,:));

    %Triangulate
    TT_s=triangulation(TT_cells_new, TT_points_new);
    %trisurf(TT_s);

    %Export
    if isleft
```

```
        stlwrite(TT_s,[inpath,'/left/SLV.stl'])
        disp(['Exported stl file of semiliunar valve geometry to ' ...
            inpath '/left/SLV.stl'])
    else
        stlwrite(TT_s,[inpath,'/right/SLV.stl'])
        disp(['Exported stl file of semiliunar valve geometry to ' ...
            inpath '/right/SLV.stl'])
    end

    % Also export a version without the elements at the side for ...
        the gmsh volume mesh

    % Delete side cells
    [dSLV_points, dSLV_cells, ¬, ¬...
        ,¬,¬]=delete_sharedSurfaces(TT_points_new, TT_cells_new, ...
        v_trunks_p{1}, v_trunks_c{1});

    %Triangulate
    TT_s=triangulation(dSLV_cells, dSLV_points);
    %trisurf(TT_s);

    %Export
    if isleft
        stlwrite(TT_s,[inpath,'/left/SLV_gmsh.stl'])
        disp(['Exported stl file of semiliunar valve geometry for ...
            gmsh to ' inpath '/left/SLV_gmsh.stl'])
    else
        stlwrite(TT_s,[inpath,'/right/SLV_gmsh.stl'])
        disp(['Exported stl file of semiliunar valve geometry for ...
            gmsh to ' inpath '/right/SLV_gmsh.stl'])
    end
end

%% Process atrial trunks
% Delete shared surfaces atrium
[ST_points_new, ST_cells_new, A_points, A_cells, edge_Points_a, ¬...
    ]=delete_sharedSurfaces(ST_points_new, ST_cells_new, A_points, ...
    A_cells);

%     %% Visualization
%     ST_t=triangulation(ST_cells_new, ST_points_new);
%
%     figure
%     trisurf(ST_t)
%     hold on
%
%     stlwrite(ST_t,[path,'/sTrunks.stl'])
%     disp(['Exported stl file of short trunks geometry to ' path ...
    '/sTrunks.stl'])
%
%     A_t=triangulation(A_cells, A_points);
%
```

```
%     figure
%     trisurf(A_t)
%     hold on
%
%     stlwrite(A_t,[path,'/LA_wholes.stl'])
%     disp(['Exported stl file of LA geometry to ' path ...
    '/LA_wholes.stl'])

% Prolong atrial trunks
[a_trunks_c, a_trunks_p,LT_AT, a_lids_c, ...
    a_lids_p]=prolong_chamberTrunks(edge_Points_a, ST_cells_new, ...
    ST_points_new, length);

%triangulate and export lids
inlet_p=[];
inlet_c=[];

for i=1:1:size(a_trunks_p,2)
    %ST_AL{i}=triangulation(a_lids_c{i}, a_lids_p{i}); ...
        %triangulate each lid single
   inlet_c=[inlet_c;a_lids_c{i}+size(inlet_p,1)];
   inlet_p=[inlet_p;a_lids_p{i}];
end

ST_in=triangulation(inlet_c, inlet_p); %triangulate all lids

if isleft
    stlwrite(ST_in,[inpath,'/left/inlet.stl'])
    disp(['Exported stl file of inlet geometry to ' inpath ...
        '/left/inlet.stl'])
else
    stlwrite(ST_in,[inpath,'/right/inlet.stl'])
    disp(['Exported stl file of inlet geometry to ' inpath ...
        '/right/inlet.stl'])
end

%% Process AV geometry: Delete cells shared with atrium and ventricle
% delete cells AV/ventricle
[AV_points_new, AV_cells_new, V_points, V_cells, ¬...
    ,deletedCells1]=delete_sharedSurfaces(AV_points, AV_cells, ...
    V_points, V_cells);
ST_V=triangulation(V_cells, V_points);
ST_AV=triangulation(AV_cells, AV_points);

%Visualization
%trisurf(ST_V)
%trisurf(ST_AV)

% save deleted cells AV/ventricle
delCellsAV1=AV_cells(deletedCells1,:);
[del1_AV_p, del1_AV_c] = delete_RemainingPoints(AV_points, ...
    delCellsAV1);
```

```

% delete cells AV/atrium
[AV_points, AV_cells, A_points, A_cells, ¬...
    ,deletedCells2]=delete_sharedSurfaces(AV_points_new, ...
    AV_cells_new, A_points, A_cells);
ST_A=triangulation(A_cells, A_points);
ST_AV=triangulation(AV_cells, AV_points);

% Visualization
%trisurf(ST_A)
%trisurf(ST_AV)

% save deleted cells AV/atrium
delCellsAV2=AV_cells_new(deletedCells2,:);
[del2_AV_p, del2_AV_c] = delete_RemainingPoints(AV_points_new, ...
    delCellsAV2);

% combine and export deleted surfaces (for volume mesh creation)
dAV_cells=[del1_AV_c; del2_AV_c+size(del1_AV_p,1)];
dAV_points=[del1_AV_p; del2_AV_p];

dAV=triangulation(dAV_cells,dAV_points);
%trisurf(dAV)

if isleft
    stlwrite(dAV,[inpath,'/left/AV_gmsh.stl'])
    disp(['Exported stl file of LAV geometry for gmsh to ' inpath ...
        '/left/AV_gmsh.stl'])
else
    stlwrite(dAV,[inpath,'/right/AV_gmsh.stl'])
    disp(['Exported stl file of AV geometry for gmsh to ' inpath ...
        '/right/AV_gmsh.stl'])
end

clearvars deletedCells1 deletedCells2;

%% Combine all part surfaces into one complete surface
all_p=[];
all_c=[];

%combine all parts in one surface (excluding lids) - for GMSH
allpoints=[{A_points,V_points,AV_points} ,a_trunks_p,v_trunks_p];
allcells=[{A_cells,V_cells,AV_cells} ,a_trunks_c,v_trunks_c];
allnames={'A_points','V_points','AV_points'};
for i=1:1:size(a_trunks_p,2)
    allnames(end+1)={['atrialTrunk ' num2str(i)]};
end
for i=1:1:size(v_trunks_c,2)
    allnames(end+1)={['ventricularTrunk ' num2str(i)]};
end

disp(newline)
```

```
for i=1:1:size(allpoints,2)
   all_c=[all_c;allcells{i}+size(all_p,1)];
   all_p=[all_p;allpoints{i}];
   disp(['Center point of ' allnames{i} ' is (' ...
       num2str(mean(allpoints{i})) ') >>use in set batch'])
end
disp(newline)

%% Optimization: If mechanical mesh is faulty correct surface
% Appears especially, when single points (and 3-4 connected cells) is
% shifted inside/outside the true surface
del_cells=[];
del_points=[];

for i=1:1:size(delPoints,1)
dist=(all_p(:,1)-delPoints(i,1)).^2 + ...
    (all_p(:,2)-delPoints(i,2)).^2 + (all_p(:,3)-delPoints(i,3)).^2;
[¬, id_ex]=min(dist);

[connected_faces,¬,found_edge_Points] = ...
    identify_ConnectedCells(all_c, id_ex, true);

%store deleted cells and points for visualization
[del_p, del_c] = delete_RemainingPoints(all_p, ...
    all_c(connected_faces,:));
del_cells=[del_cells;del_c+size(del_points,1)];
del_points=[del_points;del_p];

all_c(connected_faces,:)=[]; %delete concerning cells

if size(found_edge_Points,1)==3
    all_c=[all_c; found_edge_Points'];
    elseif size(found_edge_Points,1)==4 %find point that is most ...
        far away
        dist=rms((all_p(found_edge_Points(2:4),:)-all_p(found_edge_Points(1),:))');
        [¬,p]=max(dist);
        vec1=[2;3;4];
        vec2=vec1;
        vec2(p)=1;
        all_c=[all_c; found_edge_Points(vec1)'; ...
            found_edge_Points(vec2)'];
    end
end

if ¬isempty(del_points)
    del_t=triangulation(del_cells,del_points);
    %trisurf(del_t,'Facecolor','red')

    [all_p, all_c] = delete_RemainingPoints(all_p, all_c);
end

%% Triangulate and export without lids
```

```
%triangulate and export all surfaces
ST_all=triangulation(all_c, all_p); %triangulate all lids

if isleft
    stlwrite(ST_all,[inpath,'/left/fluid_gmsh.stl'])
    disp(['Exported stl file of whole fluid geometry for GMSH to ' ...
        inpath '/left/fluid_gmsh.stl'])
else
    stlwrite(ST_all,[inpath,'/right/fluid_gmsh.stl'])
    disp(['Exported stl file of whole fluid geometry for GMSH to ' ...
        inpath '/right/fluid_gmsh.stl'])
end

%% Add lids, triangulate and export with lids
%combine all parts in one surface (including lids) - for MATLAB
allpoints=[{inlet_p},v_lids_p];
allcells=[{inlet_c} ,v_lids_c];

for i=1:1:size(allpoints,2)
   all_c=[all_c;allcells{i}+size(all_p,1)];
   all_p=[all_p;allpoints{i}];
end

%triangulate and export all surfaces
ST_all_l=triangulation(all_c, all_p); %triangulate all lids

if isleft
    stlwrite(ST_all_l,[inpath,'/left/fluid.stl'])
    disp(['Exported stl file of whole fluid geometry to ' inpath ...
        '/left/fluid.stl'])
else
    stlwrite(ST_all_l,[inpath,'/right/fluid.stl'])
    disp(['Exported stl file of whole fluid geometry to ' inpath ...
        '/right/fluid.stl'])
end

%% visualization
% single components
% figure
% hold on
% %trisurf(ST_u) %Updated small trunks
% trisurf(LT_VT{1}) %ventricular trunks = aorta
% trisurf(ST_VL) %ventricular lid = outlet
% for i=1:1:size(LT_AT,2)
%     trisurf(LT_AT{i}) % plot atrial trunk i;
% end
% trisurf(ST_in) % plot atrial lids;
% trisurf(ST_V)
% trisurf(ST_A)
% trisurf(ST_AV)

% all components
```

```
figure
hold on
trisurf(ST_all)

if exist('del_t','var') %corrected any cell
    trisurf(del_t,'Facecolor','red')
end

%% End of pipeline




%% How to create a video
% Set a breakpoint at line 95. Run until there.
%
% if ¬exist([inpath '/video'], 'dir')
%    mkdir([inpath '/video'])
% end
%
% [AV_points, AV_cells, V_points, V_cells, ¬...
    ,¬]=delete_sharedSurfaces(AV_points, AV_cells, V_points, V_cells);
% ST_V=triangulation(V_cells, V_points);
% ST_AV=triangulation(AV_cells, AV_points);
% [AV_points, AV_cells, A_points, A_cells,¬, ¬...
    ]=delete_sharedSurfaces(AV_points, AV_cells, A_points, A_cells);
% ST_A=triangulation(A_cells, A_points);
% ST_AV=triangulation(AV_cells, AV_points);
%
% figure
% hold on
% xlim([-0.1 0.1])
% ylim([-0.1 0.1])
%
% trisurf(ST_V)
% trisurf(ST_A)
% trisurf(ST_AV)
% trisurf(ST_u)
%
% uncomment block starting with line 985: "step-by-step-video"
% continue execution
%
% % if you also want to plot the lids, use this section
% trisurf(ST_VL) %ventricular lid = outlet
% trisurf(ST_in) % plot atrial lids;
% f=1;
% inpath='/Users/.../fluidGeo';
% while isfile([inpath '/video/prolongTrunk_' num2str(f) '.png'])
%     f=f+1;
% end
% saveas(gcf,[inpath '/video/prolongTrunk_' num2str(f) '.png'])

```

```

%% FUNCTIONS

function [V_points,V_cells] = reduce_MultiplePoints(V)
    % sort out double points
    [V_points,¬,V_pointID]=unique(V.vertices,'rows');

    for i=1:1:size(V.faces,1)
        for j=1:1:size(V.faces,2)
            V_cells(i,j)=V_pointID(V.faces(i,j));
        end
    end
end

function [match] = find_CommonPoints(ST_points, V_points)
% find ST_points in V_points and store in matrix match.
% Syntax: ST_point_ID, V_point_ID, distance
    match=[];

    for i=1:1:size(ST_points,1)

        dist=(V_points(:,1)-ST_points(i,1)).^2 + ...
            (V_points(:,2)-ST_points(i,2)).^2 + ...
            (V_points(:,3)-ST_points(i,3)).^2;

        if(find(dist<1E-10))
            match=[match; ...
                i,find(dist<1E-10),dist(find(dist<1E-10))]; %list ...
                of ST_pointID found in V_pointID and corresponding ...
                distance
        end
    end
end

function [ST_cells_double] = find_CommonCells(ST_cells, match)
% Based on common points (correspondency in match), extract cells
    % identify double faces (also included in V) included in ST_cells
    commfaces=zeros(size(ST_cells,1),size(ST_cells,2));

    for i=1:1:size(match,1)
        commfaces=commfaces+(ST_cells==match(i,1));
    end

    ST_cells_double=find((commfaces(:,1)+commfaces(:,2)+commfaces(:,3))==3);
end

function [ST_cells_new] = delete_CellsbyCells(ST_cells, ...
    ST_cells_double)
% Delete cells ST_cells_double from ST_cells
    % delete double cells from ST_cells
    ST_cells_new=ST_cells;
    for i=size(ST_cells_double,1):-1:1
```

```
        ST_cells_new(ST_cells_double(i),:)=[];
    end
end

function [T1_cells] = delete_CellsbyIDs(T1_cells, distant_cellIDs)
% Delete cells by ID from T1_cells

distant_cellIDs=sort(distant_cellIDs);
    for i=size(distant_cellIDs,1):-1:1
        T1_cells(distant_cellIDs(i),:)=[];
    end
end

function [ST_points_new, ST_cells_new, edgePoints] = ...
    delete_RemainingPoints_KeepEdge(ST_points, ST_cells_new, match)
% Delete points that are not longer needed and shift cell IDs
    % identify outdated points
    todelete=[];
    for i=size(ST_points,1):-1:1
        if isempty(find(ST_cells_new(:,:)==i))
            todelete=[todelete;i];
        end
    end

    % shift pointIDs and delete outdated points
    ST_points_new=ST_points;
    edgePoints=match(:,1);

%     %Visualization
%     tt=triangulation(ST_cells_new,ST_points)
%     trisurf(tt)
%     hold on
%     scatter3(ST_points(edgePoints,1), ST_points(edgePoints,2), ...
    ST_points(edgePoints,3),'r*')
%     text(ST_points(edgePoints,1), ST_points(edgePoints,2), ...
    ST_points(edgePoints,3),num2str(edgePoints))
%     scatter3(ST_points(todelete,1), ST_points(todelete,2), ...
    ST_points(todelete,3),'b*')

    for i=1:1:size(todelete,1)
       ST_cells_new = ST_cells_new - double(ST_cells_new>todelete(i));
       edgePoints = edgePoints - double(edgePoints>todelete(i));
       ST_points_new(todelete(i),:)=[]; %delete points inside the ...
           (deleted) surface
       edgePoints(find(edgePoints==todelete(i),1,'first'),:)=[]; ...
           %points on the edge of the surface
    end
end

function [ST_points_new, ST_cells_new] = ...
    delete_RemainingPoints(ST_points, ST_cells_new)
% Delete points that are not longer needed and shift cell IDs
```

```
    % identify outdated points
    todelete=[];
    for i=size(ST_points,1):-1:1
        if isempty(find(ST_cells_new(:,:)==i))
            todelete=[todelete;i];
        end
    end

    % shift pointIDs and delete outdated points
    ST_points_new=ST_points;
    for i=1:1:size(todelete,1)
       ST_cells_new = ST_cells_new - double(ST_cells_new>todelete(i));
       ST_points_new(todelete(i),:)=[]; %delete points inside the ...
           (deleted) surface
    end
end

function [connected_faces,rest,found_edge_Points] = ...
    identify_ConnectedCells(ST_cells_new, all_edge_Points,only_first)
% Use edge points to identify connected cells.
% Then loop until all connected faces are identified.
    %start with 1st edge point.
    edge_Points=all_edge_Points(1);

    %Find all connected cells
    connected_faces=[];
    num_connected_faces=[NaN,NaN];
    found_edge_Points=[];

    while size(connected_faces,1)≠num_connected_faces(2)
    %Repeat stage 1 and 2 until no more cells are added!!

        %stage 1
        for i=1:1:size(edge_Points,1)
            [c,¬]=find(ST_cells_new==edge_Points(i));
            connected_faces=unique([connected_faces;c]);
        end

        %stage 2
        for i=1:1:size(connected_faces,1)
            edge_Points=unique([edge_Points;ST_cells_new(connected_faces(i),:)']);
        end

        %stop criterion
        if only_first %if only first runthrough is performed, ...
            still return edge points
            found_edge_Points=edge_Points;
            found_edge_Points(found_edge_Points==all_edge_Points,:)=[];
            rest=[];
            return;
        end
        num_connected_faces(2)=num_connected_faces(1);
```

```
            num_connected_faces(1)=size(connected_faces,1);
        end

    %Check if all edge points are captured or if there is an ...
        independant part
    for i=size(all_edge_Points,1):-1:1
        if find(edge_Points==all_edge_Points(i,1))
            found_edge_Points=[found_edge_Points; ...
                all_edge_Points(i,1)];
            all_edge_Points(i,:)=[];
        end
    end

    %If there are leftover points, return them
    rest=all_edge_Points;
end

function [n,a] = calc_NormalVector(ST_points_new, T1_points, ...
    edge_Points)
    % plane trough remaining points (that for the edge of the surface
    a=fit([ST_points_new(edge_Points,1),ST_points_new(edge_Points,2)], ...
        ST_points_new(edge_Points,3),'poly11');
    n=[a.p10 a.p01 -1]; %normal vector
    n=n/norm(n);

    % ensure that normal vector points inside the trunk direction
    b=mean(ST_points_new(edge_Points,:))+n*rms(mean(ST_points_new(edge_Points,:))); ..
        %center of edge point shifted in normal direction
    c=mean(ST_points_new(edge_Points,:))-n*rms(mean(ST_points_new(edge_Points,:))); ..
        %center of edge point shifted oppposite to normal direction

    if rms(mean(T1_points)-b)<rms(mean(T1_points)-c)
        n=-n; %change direction of n
    end
end

function [n] = correct_NormalVector(edge_Points_up,edge_Points_low)
% Calculate normal vector as mean of upper and lower edge Point field
% Also ensure direction corresponds to n_old
    % vector trough mid points of upper and lower plane
    n=mean(edge_Points_low)-mean(edge_Points_up);
    n=n/norm(n);

%     %Visualization
%     p_m=mean(edge_Points);
%     moved=p_m-0.05*n_old;
%     moved2=p_m-0.05*n;
%     test=[p_m;moved];
%     test2=[p_m;moved2];
%
%     figure
%     plot(a, [edge_Points(:,1) edge_Points(:,2)], edge_Points(:,3))
%     hold on
%     plot3(p_m(1),p_m(2),p_m(3),'o')
%     plot3(moved(1),moved(2),moved(3),'o')
%     line(test(:,1),test(:,2),test(:,3))
%     plot3(moved2(1),moved2(2),moved2(3),'o')
%     line(test2(:,1),test2(:,2),test2(:,3),'Color','red')
%     axis equal;
end

function [distant_cells, distant_cellIDs] = ...
    extract_perpendicularCells(T1_points, T1_cells, n)
% identify all cells that have the same (similar) normal vector ...
    like n
ang=zeros(size(T1_cells,1),1);

for i=1:1:size(T1_cells,1)
    x=T1_points(T1_cells(i,1),:);
    y=T1_points(T1_cells(i,2),:);
    z=T1_points(T1_cells(i,3),:);

    c=cross((z-x),(y-x));
    c=c/norm(c);

    ang(i) = atan2d(norm(cross(c,n)),dot(c,n));
end

% % plot
% plot3(x(1),x(2),x(3),'*')
% plot3(x(1)-0.05*c(1),x(2)-0.05*c(2),x(3)-0.05*c(3),'*')

distant_cellIDs=find(ang<45|ang>135);
distant_cells=T1_cells(distant_cellIDs,:);

% Up to this point, the cells are only categorized by angle.
% Additionally evaluate connected cells.
rest=1:1:size(T1_points,1);
rest=rest';
all_connected_faces=[];
k=1;

while ¬isempty(rest)
    [connected_faces,rest,¬]=identify_ConnectedCells(distant_cells, ...
        rest, false);
    if ¬isempty(connected_faces)
        all_connected_faces{k}=connected_faces;
        k=k+1;
    end
end

%chose surface with most cells.
maxel=0;
for k=1:1:size(all_connected_faces,2)
```

```
    numelems=size(all_connected_faces{k},1);
    if numelems>maxel
        maxel=numelems;
        id=k;
    end
end

%choose correct distant cells
distant_cells=distant_cells(all_connected_faces{id},:);

%choose correct distant cellIDs
distant_cellIDs=distant_cellIDs(all_connected_faces{id},:);

% Up to this point, the cells are only categorized by angle and ...
    connectivity.
% Additionally evaluate if cells at the edge are missing:
% This is the case, if all three point are already in the ...
    triangulation
% because they are used by neighbouring points
allpoints=unique(distant_cells);
found_cells=[];

    for i=1:1:size(allpoints,1)
        [a,¬]=find(T1_cells==allpoints(i));
        if ¬isempty(a)
            for j=1:1:size(a,1)
                [e,¬]=find(T1_cells(a(j),1)==allpoints);
                [f,¬]=find(T1_cells(a(j),2)==allpoints);
                [g,¬]=find(T1_cells(a(j),3)==allpoints);
                if ¬isempty(e)&&¬isempty(f)&&¬isempty(g)
                    found_cells=[found_cells;a(j)];
                end
            end
        end
    end

distant_cellIDs=unique(found_cells);
distant_cells=T1_cells(unique(found_cells),:);
end

function [avg_edgeLength, neighbours] = ...
    calc_averageEdgeLength(outerEdge_points)
% Find 2 neighbours for each node and calc distance. Calc average.

% find two neighbours for each node
neighbours=zeros(size(outerEdge_points,1),3);

for i=1:1:size(outerEdge_points,1)
    dist=sqrt((outerEdge_points(:,1)-outerEdge_points(i,1)).^2 + ...
        (outerEdge_points(:,2)-outerEdge_points(i,2)).^2 + ...
        (outerEdge_points(:,3)-outerEdge_points(i,3)).^2);
    [¬,d]=min(dist);
    dist(d)=NaN;
    neighbours(i,1)=d; %first minimum ist point itself // ignored ...
        // equals i
    [c,d]=min(dist);
    neighbours(i,2)=d; %second minimum is closest neighbour point
    neighbours(i,4)=c; %distance to closest neighbour point
    dist(d)=NaN;
    [c,d]=min(dist);
    neighbours(i,3)=d; %third minimum is second closest neighbour ...
        point
    neighbours(i,5)=c; %distance to second closest neighbour point
    dist(d)=NaN;
end

% calculate mean value
avg_edgeLength=mean(mean(neighbours(:,4:5)));
end

function [longTrunk_points, longTrunk_cells] = ...
    add_CellRow(longTrunk_points, longTrunk_cells, ...
    outerEdge_points,neighbours, avg_edgeLength,j,n, match)
 %add points
    addedpoints=outerEdge_points-avg_edgeLength*j*n;
    longTrunk_points=[longTrunk_points; addedpoints];
    %scatter3(addedpoints(:,1),addedpoints(:,2),addedpoints(:,3),'*')

    %compute faces
    idx=[1 NaN];
    numNeigh=size(neighbours,1); %number of points in edge ring ...
        (added each cycle)
    offset=size(longTrunk_points,1)-numNeigh;

    while true
        if j==1 %first loop: points at the edge are not at the end ...
            of the list
            if neighbours(idx(1),2)≠idx(2)
                f1=[match(idx(1),2) match(neighbours(idx(1),2),2) ...
                    idx(1)+offset];
                f2=[match(neighbours(idx(1),2),2) ...
                    neighbours(idx(1),2)+offset idx(1)+offset];
                idx(2)=idx(1);
                idx(1)=neighbours(idx(1),2);
            else
                f1=[match(idx(1),2) match(neighbours(idx(1),3),2) ...
                    idx(1)+offset];
                f2=[match(neighbours(idx(1),3),2) ...
                    neighbours(idx(1),3)+offset idx(1)+offset];
                idx(2)=idx(1);
                idx(1)=neighbours(idx(1),3);
            end
        else
            if neighbours(idx(1),2)≠idx(2)
```

```
                f1=[idx(1)+offset-numNeigh ...
                    neighbours(idx(1),2)+offset-numNeigh ...
                    idx(1)+offset];
                f2=[neighbours(idx(1),2)+offset-numNeigh ...
                    neighbours(idx(1),2)+offset idx(1)+offset];
                idx(2)=idx(1);
                idx(1)=neighbours(idx(1),2);
            else
                f1=[idx(1)+offset-numNeigh ...
                    neighbours(idx(1),3)+offset-numNeigh ...
                    idx(1)+offset];
                f2=[neighbours(idx(1),3)+offset-numNeigh ...
                    neighbours(idx(1),3)+offset idx(1)+offset];
                idx(2)=idx(1);
                idx(1)=neighbours(idx(1),3);
            end

        end

        longTrunk_cells=[longTrunk_cells;f1;f2]; % add 2 new cells

        if idx(1)==1
            break
        end
    end
end

function [ST_points, ST_cells, ST_u] = ...
    delete_AllConnectedCells(ST_points, ST_cells, delA_points)
% Find all connected Cells, delete them and corresponding points, ...
    update
% triangulation ST_u

%Find points included in both geometries
[match] = find_CommonPoints(ST_points, delA_points);
%scatter3(ST_points(match(:,1),1),ST_points(match(:,1),2),ST_points(match(:,1),3))

%Find and delete all connected cells
rest=match;
all_connected_faces=[];

while ¬isempty(rest)
    [connected_faces,rest,¬] = identify_ConnectedCells(ST_cells, ...
        rest, false);
    all_connected_faces=[all_connected_faces;connected_faces];
end

[ST_cells] = delete_CellsbyIDs(ST_cells, all_connected_faces);

%Identify outdated points, delete them and shift cell IDs
[ST_points, ST_cells] = delete_RemainingPoints(ST_points, ST_cells);

```

```
%Update triangulation
ST_u=triangulation(ST_cells, ST_points);
end

function [ST_points_new, ST_cells_new, V_points, V_cells, ...
    edge_Points,ST_cells_double]=delete_sharedSurfaces(ST_points, ...
    ST_cells, V_points, V_cells)
% Identify shared surface, delete from both triangulations. Return ...
    edge and deleted cells.

%% Find shared points and cells and delete them from ST (lower lid ...
    of trunks)
%Find points included in both geometries
[match] = find_CommonPoints(ST_points, V_points);

%Same for cells. Also delete them.
[ST_cells_double]=find_CommonCells(ST_cells, match(:,1));
[ST_cells_new] = delete_CellsbyCells(ST_cells, ST_cells_double);

%Identify outdated points, delete them and shift cell IDs
[ST_points_new, ST_cells_new, edge_Points] = ...
    delete_RemainingPoints_KeepEdge(ST_points, ST_cells_new, match);

%% visualization
% ST_t=triangulation(ST_cells_new, ST_points_new);
%
% figure
% trisurf(ST_t)
% hold on

%% Also find shared points and cells and delete them from V
%Same for cells. Also delete them.
[V_cells_double]=find_CommonCells(V_cells, match(:,2));
[V_cells] = delete_CellsbyCells(V_cells, V_cells_double);

%Identify outdated points, delete them and shift cell IDs
[V_points, V_cells] = delete_RemainingPoints(V_points, V_cells);

%% visualization
% V_t=triangulation(V_cells, V_points);
%
% figure
% trisurf(V_t)
end

function [longTrunk_cells, longTrunk_points, LT_t, lid_cells, ...
    lid_points] = calc_prolongedTrunk(T1_cells, T1_points, ...
    ST_points_new, edge_Points, length)
% Calculates points and cells for one prolonged trunk, given by
    %% visualization
    % ST_t=triangulation(T1_cells, T1_points);
    %
    % figure
    % trisurf(ST_t)
    % hold on

    %% Fit a plane through the EdgePoints and calculare normal vector
    [n,a] = calc_NormalVector(ST_points_new, T1_points, edge_Points);

    %% Alternatively (to be independant from plane fit) determine n
    % by calculating the vector between
    % mean(allTrunkPoints)-mean(EdgePoints)
    n=mean(T1_points)-mean(ST_points_new(edge_Points,:));
    n=n/norm(n);

%     %% visualization
%     %calculate mean of all remaining points
%     p_m=mean(T1_points);
%     moved=p_m-0.05*n;
%     test=[p_m;moved];
%
%     ST_t=triangulation(T1_cells, T1_points);
%
%     figure
%     plot(a, [ST_points_new(edge_Points,1) ...
    ST_points_new(edge_Points,2)], ST_points_new(edge_Points,3))
%     hold on
%     plot3(moved(1),moved(2),moved(3),'o')
%     plot3(p_m(1),p_m(2),p_m(3),'o')
%     line(test(:,1),test(:,2),test(:,3))
%     trisurf(ST_t)

    %% determine if triangle normal vector is parallel to plane normal
    [distant_cells, distant_cellIDs] = ...
        extract_perpendicularCells(T1_points, T1_cells, n);
    [distant_points, distant_cells] = ...
        delete_RemainingPoints(T1_points, distant_cells);

    % Delete distant cells and points from T1 mesh.
    [T1_cells] = delete_CellsbyIDs(T1_cells, distant_cellIDs);
    [T1_points, T1_cells] = delete_RemainingPoints(T1_points, ...
        T1_cells);

%      %% visualization
%      ST_t=triangulation(distant_cells, distant_points);
%
%      figure
%      trisurf(ST_t)

    %% identify common points of T1_cells and distant_cells ...
        (=upper plane)

    % find points included in both geometries
    [match] = find_CommonPoints(distant_points, T1_points);

    %% compute average edge length of outer edges
    % Extract Edge
    outerEdge_points=distant_points(match(:,1),:);

    %Calc avg length
    [avg_edgeLength, neighbours] = ...
        calc_averageEdgeLength(outerEdge_points);

    %% Correct normal vector (use upper plane normal vector)
    [n] = ...
        correct_NormalVector(outerEdge_points,ST_points_new(edge_Points,:));

%     %% visualization
%     %calculate mean of all remaining points
%     p_m=mean(T1_points);
%     moved=p_m-0.05*n;
%     test=[p_m;moved];
%
%     ST_t=triangulation(T1_cells, T1_points);
%
%     figure
%     plot(a, [ST_points_new(edge_Points,1) ...
   ST_points_new(edge_Points,2)], ST_points_new(edge_Points,3))
%     hold on
%     plot3(moved(1),moved(2),moved(3),'o')
%     plot3(p_m(1),p_m(2),p_m(3),'o')
%     line(test(:,1),test(:,2),test(:,3))
%     trisurf(ST_t)
%     axis equal

%     %% visualization
%     T1_t=triangulation(T1_cells, T1_points);
%
%     figure
%     hold on
%     trisurf(T1_t)
%
%     for i=1:1:size(outerEdge_points,1)
%        plot3(outerEdge_points(i,1), outerEdge_points(i,2), ...
   outerEdge_points(i,3),'o')
%        text(outerEdge_points(i,1), ...
   outerEdge_points(i,2),outerEdge_points(i,3),string(i),'FontSize',18)
%     end

    %% add points and cells with avg_edgeLength distance in normal ...
        direction
    longTrunk_points=T1_points;
    longTrunk_cells=T1_cells;

    for j=1:1:length
```

```
        [longTrunk_points, longTrunk_cells] = ...
            add_CellRow(longTrunk_points, longTrunk_cells, ...
            outerEdge_points,neighbours, avg_edgeLength,j,n, match);

%         % If step-by-step-video should be recorded
%         LT_t=triangulation(longTrunk_cells, longTrunk_points);
%         trisurf(LT_t)
%         f=1;
%         inpath='/Users/.../fluidGeo';
%         while isfile([inpath '/video/prolongTrunk_' num2str(f) ...
    '.png'])
%             f=f+1;
%         end
%         saveas(gcf,[inpath '/video/prolongTrunk_' num2str(f) ...
    '.png'])
    end

    LT_t=triangulation(longTrunk_cells, longTrunk_points);

    %% add lid to the top of the trunk (move plane in normal ...
        direction)
    lid_points=distant_points-n*avg_edgeLength*length;
    lid_cells=distant_cells;
end

function [trunks_c, trunks_p, triang, lids_c, ...
    lids_p]=prolong_chamberTrunks(edge_Points_v, ST_cells_new, ...
    ST_points_new, length)
% prolong all trunks belonging to the concerning chamber

    %% separate the concerning single trunk(s) and prolong it
    rest=edge_Points_v;
    all_connected_faces=[];
    k=1;

    % list all cells that are connected to the edge ring.
    % write one cell element for each closed surface.
    while ¬isempty(rest)
        [connected_faces,rest,edge_Points{k}] = ...
            identify_ConnectedCells(ST_cells_new, rest, false);
        all_connected_faces{k}=connected_faces;
        k=k+1;
    end

    %% for each trunk (=1 cell element) prolong the trunk.
    for k=1:1:size(all_connected_faces,2)
        %% create seperate triangulation
        % extract connected cells
        T1_cells=ST_cells_new(all_connected_faces{k},:);

        %Identify outdated points, delete them and shift cell IDs
        [T1_points, T1_cells] = ...
            delete_RemainingPoints(ST_points_new, T1_cells);

        ST_t=triangulation(T1_cells, T1_points);

        %figure
        %trisurf(ST_t)

        %prolong one single trunk
        [longTrunk_cells, longTrunk_points, LT_t, lid_cells, ...
            lid_points] = calc_prolongedTrunk(T1_cells, T1_points, ...
            ST_points_new, edge_Points{k}, length);

        %% save trunk in cell array
        trunks_c{k}=longTrunk_cells;
        trunks_p{k}=longTrunk_points;

        %% save lids in cell array
        lids_c{k}=lid_cells;
        lids_p{k}=lid_points;

        %% save triangulation in cell array
        triang{k}=LT_t;

    end
end
```

extractTunksTimeSeries.m:

```
% extractTrunksTimeSeries.m
% Read mechanic simuation results, adapt trunks and save stl files.
% Input: [req].vtu file of CardioMechancis Simulation results ...
    (time series)
%  Assumes the first vtk file to represent the inital geometry! ...
    And also
%  to be the base for the fluid surface (created with prolongTrunks.m)
% Input: [req].stl file of fluid surface
% Input: [req].stl file of inlet
% Input: [req].stl file of outlet
% Output: .stl file of fluid geometry (whole heart side time series)
% jb072, 12.01.2022

%% clear
clear
close all
clc

%% Input data
fpath='/Users/.../fluidGeo/left'; %fluid input data
mpath='/Users/.../Baseline_vtu'; %mechanics input path

%% Output folder
if ¬exist([fpath '/fluidTimeSeries'], 'dir')
   mkdir([fpath '/fluidTimeSeries'])
end

%% List mechanical simulation results
% get number of simulation steps and filename
l = dir(mpath);
l(1:2)=[]; %delete '.' and '..' directory from list
for i=size(l,1):-1:1
    if l(i).isdir==1 % ignore folders
        l(i)=[];
    elseif l(i).name(1)=='.' % ignore mac files (starting with .)
        l(i)=[];
    end
end
if size(l,1)==0
    disp(['No files in mechanical simulation results path ' mpath]);
    return
end

%% sort in natural order (not ascii based)
natnum=zeros(size(l,1),1);
for i=1:1:size(l,1)
    idx=strfind(l(i).name,'.');
    natnum(i,1)=str2double(l(i).name(idx(1)+1:idx(end)-1));
end
```

```
natnum=sort(natnum);

idx=strfind(l(1).name,'.');
pre=l(1).name(1:idx(1)); %file name prefix
post=l(1).name(idx(2):end); %file name ending

%% Read initial mechanical geometry
m0=readVTK([mpath '/' pre num2str(natnum(1)) post]);

%% Read fluid geometry
FL=stlread([fpath,'/fluid.stl']);
inl=stlread([fpath,'/inlet.stl']);
outl=stlread([fpath,'/outlet.stl']);

FL_i=triangulation(FL.faces, FL.vertices);
%trisurf(FL_i)

%% Preprocess: Delete points that appear multiple times
% First: Reduce points that appear multiple times
[F_points,F_cells] = reduce_MultiplePoints(FL);
[inl_points,inp_cells] = reduce_MultiplePoints(inl);
[outl_points,outp_cells] = reduce_MultiplePoints(outl);

%% Find correspondances
% moving fluid cells point IDs
[match] = find_CommonPoints(F_points, m0.points);

mov_pointsF_id=match(:,1);
mov_pointsM_id=match(:,2);

[mov_cells]=find_CellsByPointID(mov_pointsF_id, F_cells);
[mov_points, mov_cells] = delete_RemainingPoints(F_points, mov_cells);

% static fluid cells point IDs
stat_pointsF_id=(1:1:size(F_points,1))';

for i=size(stat_pointsF_id,1):-1:1
    [c,¬]=find(stat_pointsF_id(i)==(mov_pointsF_id));
    if ¬isempty(c)
        stat_pointsF_id(i)=[];
    end
end

[stat_cells]=find_CellsByPointID(stat_pointsF_id, F_cells);
[stat_points, stat_cells] = delete_RemainingPoints(F_points, ...
    stat_cells);

% inlet and outlet point IDs
[match] = find_CommonPoints(F_points, inl_points);
inl_pointsF_id=match(:,1);

[inl_cells]=find_CellsByPointID(inl_pointsF_id, F_cells);
```

```
[inl_points, inl_cells] = delete_RemainingPoints(F_points, inl_cells);

[match] = find_CommonPoints(F_points, outl_points);
outl_pointsF_id=match(:,1);

[outl_cells]=find_CellsByPointID(outl_pointsF_id, F_cells);
[outl_points, outl_cells] = delete_RemainingPoints(F_points, ...
    outl_cells);

% remove inlet and outlet from static points
for i=size(stat_pointsF_id,1):-1:1
    [c,¬]=find(stat_pointsF_id(i)==(inl_pointsF_id));
    [d,¬]=find(stat_pointsF_id(i)==(outl_pointsF_id));
    if ¬isempty(c)||¬isempty(d)
        stat_pointsF_id(i)=[];
    end
end

[stat_cells]=find_CellsByPointID(stat_pointsF_id, F_cells);
[stat_points, stat_cells] = delete_RemainingPoints(F_points, ...
    stat_cells);

%% Visualization
F_m=triangulation(mov_cells, mov_points);
F_s=triangulation(stat_cells, stat_points);
F_in=triangulation(inl_cells, inl_points);
F_out=triangulation(outl_cells, outl_points);

% figure;
% hold on;
% ...
    trisurf(F_m,'Facecolor','black','FaceAlpha',.2,'EdgeColor','black','EdgeAlpha',.4)
% ...
    trisurf(F_in,'Facecolor','red','FaceAlpha',.2,'EdgeColor','red','EdgeAlpha',.4)
% ...
    trisurf(F_out,'Facecolor','red','FaceAlpha',.2,'EdgeColor','red','EdgeAlpha',.4)
% trisurf(F_s)

%% Detect edge loops (edge only used by 1 triangle)
[outl_edgepointIDs]=determine_edgePoints(outl_cells);
%scatter3(outl_points(outl_edgepointIDs,1),outl_points(outl_edgepointIDs,2),outl_points

[inl_edgepointIDs]=determine_edgePoints(inl_cells);
%scatter3(inl_points(inl_edgepointIDs,1),inl_points(inl_edgepointIDs,2),inl_points(inl_

[mov_edgepointIDs]=determine_edgePoints(mov_cells);
%scatter3(mov_points(mov_edgepointIDs,1),mov_points(mov_edgepointIDs,2),mov_points(mov_

%% Split up trunks and lids in single triangulations (only ...
    connected cells)
rest=(1:1:size(stat_points,1))';
trunk=[];
```

```
k=1;

while ¬isempty(rest)
    [connected_faces,rest,found_edgePoints]=identify_ConnectedCells(stat_cells, ...
        rest);
    if ¬isempty(connected_faces)
        trunk{k}=stat_pointsF_id(found_edgePoints);
        k=k+1;
    end
end

% visualization
% for i=1:1:5
%     figure
%     a=stat_points(trunk{i},:)
%     scatter3(a(:,1),a(:,2),a(:,3),'o');
% end

%same for mov_edgepointIDs
[¬, neighbours] = ...
    calc_averageEdgeLength(mov_points(mov_edgepointIDs,:));
neighbours(:,1)=mov_edgepointIDs(neighbours(:,1),1);
[mov_edgeLoops]=find_edgeLoops(neighbours);

%same for inl_edgepointIDs
[¬, neighbours] = ...
    calc_averageEdgeLength(inl_points(inl_edgepointIDs,:));
neighbours(:,1)=inl_edgepointIDs(neighbours(:,1),1);
[inl_edgeLoops]=find_edgeLoops(neighbours);

clearvars mov_edgepointIDs inl_edgepointIDs;

%% Match edge loop and corresponding trunk
% outlet
m=mean(outl_points(outl_edgepointIDs,:));

% figure
% hold on
% scatter3(m(:,1),m(:,2),m(:,3),'*')

for k=1:1:size(trunk,2)
   p=F_points(trunk{k},:);
%    scatter3(p(:,1),p(:,2),p(:,3))
   ed(k)=min(sqrt((p(:,1)-m(1)).^2+(p(:,2)-m(2)).^2+(p(:,3)-m(3)).^2));
end

[¬,outl_trunkID]=min(ed);

% inlet
for j=1:1:size(inl_edgeLoops,2)
    m=mean(inl_points(inl_edgeLoops{j},:));

```

```
%     figure
%     hold on
%     scatter3(m(:,1),m(:,2),m(:,3),'*')
%
    for k=1:1:size(trunk,2)
        p=F_points(trunk{k},:);
%         scatter3(p(:,1),p(:,2),p(:,3))
        ed(k)=min(sqrt((p(:,1)-m(1)).^2+(p(:,2)-m(2)).^2+(p(:,3)-m(3)).^2));
    end
    [¬,inl_trunkID(j)]=min(ed);
end


% moving // edge Points of lower end
for j=1:1:size(mov_edgeLoops,2)
    m=mean(mov_points(mov_edgeLoops{j},:));

%     figure
%     hold on
%     scatter3(m(:,1),m(:,2),m(:,3),'*')

    for k=1:1:size(trunk,2)
        p=F_points(trunk{k},:);
%         scatter3(p(:,1),p(:,2),p(:,3))
        ed(k)=min(sqrt((p(:,1)-m(1)).^2+(p(:,2)-m(2)).^2+(p(:,3)-m(3)).^2));
    end
    [¬,moving_trunkID(j)]=min(ed);
end

% store in lo_pointID matrix
for k=1:1:size(moving_trunkID,2)
    lo_pointID{moving_trunkID(k)}=mov_pointsF_id(mov_edgeLoops{k});
end

% store in up_pointID matrix
for k=1:1:size(inl_trunkID,2)
    up_pointID{inl_trunkID(k)}=inl_pointsF_id(inl_edgeLoops{k});
end

up_pointID{outl_trunkID}=outl_pointsF_id(outl_edgepointIDs);

clearvars moving_trunkID mov_edgeLoops inl_trunkID inl_edgeLoops;

%% Visualization
figure
hold on
for i=1:1:size(trunk,2)
    a=F_points(up_pointID{i},:);
    scatter3(a(:,1),a(:,2),a(:,3),'b*')
    b=F_points(lo_pointID{i},:);
    scatter3(b(:,1),b(:,2),b(:,3),'r*')
    c=F_points(trunk{i},:);
    scatter3(c(:,1),c(:,2),c(:,3),'g*')
end

%% Find vertical lines
% all needed cellIDs (referred to Fpoints) in lo_pointID, trunk, ...
    up_pointID
for i=1:1:size(trunk,2)
    %calc normal vector form lo to up
    n=mean(F_points(up_pointID{i},:))-mean(F_points(lo_pointID{i},:));

    %points per horizontal surfaces
    h_pts=size(lo_pointID{i},1);

    %points per vertical surfaces
    v_pts=size(trunk{i},1)/h_pts;

    trunk_trace{i}=zeros(v_pts+2,h_pts);

    for j=1:1:h_pts
        trunk_trace{i}(1,j)=lo_pointID{i}(j); %point in lo surface
        trunk_trace{i}(v_pts+2,j)=up_pointID{i}(j); %point in lo ...
            surface

        for k=1:1:v_pts
           a=F_points(trunk_trace{i}(k,j),:);
           a_pre=a+n*1/(v_pts+1); %predicted point position in n ...
               direction
           all=F_points(trunk{i},:);
           [¬,d]=min(sqrt((all(:,1)-a_pre(1)).^2+(all(:,2)-a_pre(2)).^2 ...
               +(all(:,3)-a_pre(3)).^2));
           trunk_trace{i}(k+1,j)=trunk{i}(d); %add point in trace
        end
    end
end

%% Visualization
% figure
% hold on
% for i=1:1:size(trunk_trace,2)
%     for k=1:1:size(trunk_trace{i},2)
%         a=F_points(trunk_trace{i}(:,k),:);
%         scatter3(a(:,1),a(:,2),a(:,3),'*')
%     end
% end

%% Time loop: For each available mechanic time step: extract trunk ...
    displacement
for k=1:1:size(natnum,1) %each time step
    disp(['Processing Time Step ' num2str(k)])

    % read vtu
    m=readVTK([mpath '/' pre num2str(natnum(k)) post]);

    %process trunks
    for i=1:1:size(trunk_trace,2) %each trunk

        for j=1:1:size(trunk_trace{i},2) %each trace line
            % extract points
            idx_F=trunk_trace{i}(1,j);
            pos_F=find(mov_pointsF_id==idx_F);
            idx_M=mov_pointsM_id(pos_F);

            % extract displacement
            n=m.points(idx_M,:)-F_points(idx_F,:); %vector with ...
                displacement of lo node

            % displace points for each trace line
            v_pts=size(trunk_trace{i},1); %vertical number of ...
                trunk points (including 2 lids)
            for y=1:1:v_pts %each point in trace line
                F_points(trunk_trace{i}(y,j),:)=F_points(trunk_trace{i}(y,j),:)+n*((v_p
                % a=n*((v_pts-y)/(v_pts-1))/sqrt(3);
                % sqrt(a(1)^2+a(2)^2+a(3)^2) %to test geometric length
            end
        end
    end

    %copy LA/LV movement
    F_points(mov_pointsF_id,:)=double(m.points(mov_pointsM_id,:));

    %triangulate and export
    F=triangulation(F_cells, F_points);
    %trisurf(F)

    %export with index k-1
    stlwrite(F,[fpath '/fluidTimeSeries/LV_' num2str(k-1) '.stl'])
end

disp(['Exported stl file of whole fluid geometry to ' fpath ...
    '/fluidTimeSeries/ - filename starting with LV_0.stl'])
%% End of pipeline


%% FUNCTIONS
function [V_points,V_cells] = reduce_MultiplePoints(V)
    % sort out double points
    [V_points,¬,V_pointID]=unique(V.vertices,'rows');

    for i=1:1:size(V.faces,1)
        for j=1:1:size(V.faces,2)
            V_cells(i,j)=V_pointID(V.faces(i,j));
        end
    end
end
```

```

function [match] = find_CommonPoints(ST_points, V_points)
% find ST_points in V_points and store in matrix match.
% Syntax: ST_point_ID, V_point_ID, distance
    match=[];
    for i=1:1:size(ST_points,1)

        dist=(V_points(:,1)-ST_points(i,1)).^2 + ...
            (V_points(:,2)-ST_points(i,2)).^2 + ...
            (V_points(:,3)-ST_points(i,3)).^2;

        if(find(dist<1E-10))
            match=[match; ...
                i,find(dist<1E-10),dist(find(dist<1E-10))]; %list ...
                of ST_pointID found in V_pointID and corresponding ...
                distance
        end
    end
end

function [ST_points_new, ST_cells_new] = ...
    delete_RemainingPoints(ST_points, ST_cells_new)
% Delete points that are not longer needed and shift cell IDs
    % identify outdated points
    todelete=[];
    for i=size(ST_points,1):-1:1
        if isempty(find(ST_cells_new(:,:)==i))
            todelete=[todelete;i];
        end
    end

    % shift pointIDs and delete outdated points
    ST_points_new=ST_points;
    for i=1:1:size(todelete,1)
       ST_cells_new = ST_cells_new - double(ST_cells_new>todelete(i));
       ST_points_new(todelete(i),:)=[]; %delete points inside the ...
           (deleted) surface
    end
end

function [found_cells]=find_CellsByPointID(PointIDs, all_cells)
% Given a list of pointIDs, this function extracts all cells that are
% spanned by only these points
allpoints=unique(PointIDs);
found_cellIDs=[];

for i=1:1:size(all_cells,1)
    [a,¬]=find(all_cells(i,1)==allpoints);
    [b,¬]=find(all_cells(i,2)==allpoints);
    [c,¬]=find(all_cells(i,3)==allpoints);

    if ¬isempty(a)&&¬isempty(b)&&¬isempty(c)
```

```
        found_cellIDs=[found_cellIDs;i];
    end
end

found_cells=all_cells(found_cellIDs,:);
end

function [edgepointIDs]=determine_edgePoints(cells)
edges=zeros(3*size(cells,1),2);
for i=1:1:size(cells,1)
    edges(3*i-2,:)=[cells(i,1), cells(i,2)];
    edges(3*i-1,:)=[cells(i,2), cells(i,3)];
    edges(3*i,:)=[cells(i,1), cells(i,3)];
end

for i=1:1:size(edges,1)
    [c,d]=find(edges(i,1)==edges);
    [e,f]=find(edges(i,2)==edges(c,:));
    if size(e,1)>1
        edges(c(e),:)=NaN;
    end
end

edgepointIDs=unique(edges(¬isnan(edges)));
end

function [connected_faces,rest,found_edge_Points] = ...
    identify_ConnectedCells(ST_cells_new, all_edge_Points)
% Use edge points to identify connected cells.
% Then loop until all connected faces are identified.
    %start with 1st edge point.
    edge_Points=all_edge_Points(1);

    %Find all connected cells
    connected_faces=[];
    num_connected_faces=[NaN,NaN];
    found_edge_Points=[];

    while size(connected_faces,1)≠num_connected_faces(2)
    %Repeat stage 1 and 2 until no more cells are added!!

        %stage 1
        for i=1:1:size(edge_Points,1)
            [c,¬]=find(ST_cells_new==edge_Points(i));
            connected_faces=unique([connected_faces;c]);
        end

        %stage 2
        for i=1:1:size(connected_faces,1)
            edge_Points=unique([edge_Points;ST_cells_new(connected_faces(i),:)']);
        end

```

```
            %stop criterion
            num_connected_faces(2)=num_connected_faces(1);
            num_connected_faces(1)=size(connected_faces,1);
        end

    %Check if all edge points are captured or if there is an ...
        independant part
    for i=size(all_edge_Points,1):-1:1
        if find(edge_Points==all_edge_Points(i,1))
            found_edge_Points=[found_edge_Points; ...
                all_edge_Points(i,1)];
            all_edge_Points(i,:)=[];
        end
    end

    %If there are leftover points, return them
    rest=all_edge_Points;

end

function [avg_edgeLength, neighbours] = ...
    calc_averageEdgeLength(outerEdge_points)
% Find 2 neighbours for each node and calc distance. Calc average.
% find two neighbours for each node
neighbours=zeros(size(outerEdge_points,1),3);

for i=1:1:size(outerEdge_points,1)
    dist=sqrt((outerEdge_points(:,1)-outerEdge_points(i,1)).^2 + ...
        (outerEdge_points(:,2)-outerEdge_points(i,2)).^2 + ...
        (outerEdge_points(:,3)-outerEdge_points(i,3)).^2);
    [¬,d]=min(dist);
    dist(d)=NaN;
    neighbours(i,1)=d; %first minimum ist point itself // ignored ...
        // equals i
    [c,d]=min(dist);
    neighbours(i,2)=d; %second minimum is closest neighbour point
    neighbours(i,4)=c; %distance to closest neighbour point
    dist(d)=NaN;
    [c,d]=min(dist);
    neighbours(i,3)=d; %third minimum is second closest neighbour ...
        point
    neighbours(i,5)=c; %distance to second closest neighbour point
    dist(d)=NaN;
end

% calculate mean value
avg_edgeLength=mean(mean(neighbours(:,4:5)));
end

function [edgeLoop]=find_edgeLoops(neighbours)
k=1;
toCheck=(1:1:size(neighbours,1))';
```

```

while ¬isempty(toCheck)
    el=[];
    idx=[toCheck(1) NaN];
    idx_e=idx(1);

    while true
        if neighbours(idx(1),2)≠idx(2)
            idx(2)=idx(1);
            idx(1)=neighbours(idx(1),2);
        else
            idx(2)=idx(1);
            idx(1)=neighbours(idx(1),3);
        end

        el=[el;neighbours(idx(2),1)];
        toCheck(toCheck==idx(2))=[];

        if idx(1)==idx_e
            break
        end
    end

    edgeLoop{k}=el;
    k=k+1;
end

end
```

Appendix **B**

Elastomechanics and circulatory system parameters

The included parameter file contains an overview of the underlying models and the material parameters assumed for the elastomechanical simulations. Additionally, the parameters of the circulatory system are listed.

```
<settings>Baseline Simulation for geo01</settings>
<General>
    <LogFile>./Baseline_T4_CFD_Pressure/Baseline.log</LogFile>
</General>

<Mesh>
    <Type>T4</Type>
    <Format>Tetgen</Format>
    <Sorting>None</Sorting>
    <Tetgen>
        <Unit>1e-3</Unit>
        <Nodes>../Geometry/Fluid/heartT4.node</Nodes>
        <Elements>../Geometry/Fluid/heartT4.ele</Elements>
        <Bases>../Geometry/Fluid/heartT4.bases</Bases>
        <Surfaces>../Geometry/Fluid/heartT4.sur</Surfaces>
        <DetermineNeighbors>true</DetermineNeighbors>
        <EnforceOrthonormalBases>false</EnforceOrthonormalBases>
    </Tetgen>
    <Surfaces>
        <!-- LV -->
        <Surface_131> <Type>CAVITY</Type> </Surface_131>
        <!-- RV -->
        <Surface_130> <Type>CAVITY</Type> </Surface_130>
        <!-- LA -->
        <Surface_133> <Type>CAVITY</Type> </Surface_133>
        <!-- RA -->
        <Surface_132> <Type>CAVITY</Type> </Surface_132>
        <!-- epicardium -->
        <Surface_159> <Type>CAVITY</Type> </Surface_159>
        <!-- pericardium -->
        <Surface_160> <Type>CONTACT_MASTER</Type> </Surface_160>
        <Surface_161> <Type>CONTACT_SLAVE</Type> </Surface_161>
        <!-- CFD surfaces -->
        <Surface_230> <Type>T3</Type> </Surface_230>
        <Surface_231> <Type>T3</Type> </Surface_231>
    </Surfaces>
    <Transform>
        <T4toT10>false</T4toT10>
        <T3toT6>false</T3toT6>
    </Transform>
</Mesh>

<Materials>
    <Global>
        <Damping>
            <Rayleigh>
                <Alpha>500</Alpha>
                <Beta>0.01</Beta>
            </Rayleigh>
        </Damping>
    </Global>

    <Mat_Default>
        <IgnoreCorruptElements>true</IgnoreCorruptElements>
    </Mat_Default>

<Mat_30>
        <!-- RV -->
        <Type>Guccione</Type>
        <Density>1082</Density>
        <Guccione>
            <C>278.0</C>
            <b1>12.0</b1>
            <b2>4.8</b2>
            <b3>8.4</b3>
            <K>2e5</K>
        </Guccione>
        <TensionMax>120000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.335</ContrTimeOffset>
                <RelaxTimeOffset> 0.563</RelaxTimeOffset>
                <ContrRateConst>  1.32 </ContrRateConst>
                <RelaxRateConst> 21.9  </RelaxRateConst>
                <OnsetTime>       0.184 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_30>

    <Mat_31>
        <!-- LV -->
        <Type>Guccione</Type>
        <Density>1082</Density>
        <Guccione>
            <C>278.0</C>
            <b1>12.0</b1>
            <b2>4.8</b2>
            <b3>8.4</b3>
            <K>2e5</K>
        </Guccione>
        <TensionMax>120000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.335</ContrTimeOffset>
                <RelaxTimeOffset> 0.563</RelaxTimeOffset>
                <ContrRateConst>  1.32 </ContrRateConst>
                <RelaxRateConst> 21.9  </RelaxRateConst>
                <OnsetTime>       0.184 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_31>

    <Mat_32>
        <!-- RA -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>7450</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
        <TensionMax>40000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.066</ContrTimeOffset>
                <RelaxTimeOffset> 0.251 </RelaxTimeOffset>
                <ContrRateConst>  1.99 </ContrRateConst>
                <RelaxRateConst> 11.2  </RelaxRateConst>
                <OnsetTime>       0.0 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_32>

    <Mat_33>
        <!-- LA -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>7450</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
        <TensionMax>40000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.066</ContrTimeOffset>
                <RelaxTimeOffset> 0.251 </RelaxTimeOffset>
                <ContrRateConst>  1.99 </ContrRateConst>
                <RelaxRateConst> 11.2  </RelaxRateConst>
                <OnsetTime>       0.0 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_33>

    <Mat_34>
        <!-- RAV ring -->
        <Type>Guccione</Type>
        <Density>1082</Density>
        <Guccione>
            <C>5000</C>
            <b1>14.4</b1>
            <b2>5.76</b2>
            <b3>10.08</b3>
            <K>2e5</K>
        </Guccione>
        <TensionMax>120000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.335</ContrTimeOffset>
                <RelaxTimeOffset> 0.563</RelaxTimeOffset>
                <ContrRateConst>  1.32 </ContrRateConst>
                <RelaxRateConst> 21.9  </RelaxRateConst>
                <OnsetTime>       0.184 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_34>

    <Mat_35>
        <!-- LAV ring -->
        <Type>Guccione</Type>
        <Density>1082</Density>
        <Guccione>
            <C>5000</C>
            <b1>14.4</b1>
            <b2>5.76</b2>
            <b3>10.08</b3>
            <K>2e5</K>
        </Guccione>
        <TensionMax>120000</TensionMax>
        <TensionModel>FromFunction</TensionModel>
        <FromFunction>
            <Type>DoubleHill</Type>
            <StartTime>0.0</StartTime>
            <StopTime>30.0</StopTime>
            <DoubleHill>
                <Period>          1.247  </Period>
                <ContrTimeOffset> 0.335</ContrTimeOffset>
                <RelaxTimeOffset> 0.563</RelaxTimeOffset>
                <ContrRateConst>  1.32 </ContrRateConst>
                <RelaxRateConst> 21.9  </RelaxRateConst>
                <OnsetTime>       0.184 </OnsetTime>
            </DoubleHill>
        </FromFunction>
    </Mat_35>

    <Mat_36>
        <!-- RAV plane -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>2e5</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_36>

    <Mat_37>
        <!-- LAV plane -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
```

```
            <C1>2e5</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_37>

    <Mat_38>
        <!-- inner trunks -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>14900</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_38>

    <Mat_232>
        <!-- Pulmonary Valve -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>14900</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_232>

    <Mat_233>
        <!-- Aortic Valve -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>14900</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_233>

    <Mat_39>
        <!-- outer trunks -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>14900</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_39>

    <Mat_40>
        <!-- fat -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>3725</C1>
            <C2>0</C2>
            <B>2e5</B>
        </MooneyRivlin>
    </Mat_40>

    <Mat_60>
        <!-- surrounding tissue -->
        <Type>MooneyRivlin</Type>
        <Density>1082</Density>
        <MooneyRivlin>
            <C1>1000</C1>
            <C2>0</C2>
            <B>1e4</B>
        </MooneyRivlin>
    </Mat_60>
</Materials>

<Export>
    <Format>VTK</Format>
    <Prefix>./Baseline_T4_CFD_Pressure/CM</Prefix>
    <TimeStep>5e-3</TimeStep>
    <Options>
        <Material>true</Material>
        <Fixation>true</Fixation>
        <FiberAllQuadraturePoints>false</FiberAllQuadraturePoints>
        <SurfaceElementID>false</SurfaceElementID>
        <SurfaceID>false</SurfaceID>
        <SurfaceNormal>false</SurfaceNormal>
        <ActiveStress>true</ActiveStress>
        <TetgenBases>false</TetgenBases>
        <Deformation>false</Deformation>
        <Lambda>true</Lambda>
        <Cauchy>false</Cauchy>
        <LocalActivationTime>false</LocalActivationTime>
        <Fiber>true</Fiber>
        <Sheet>true</Sheet>
        <SheetNormal>true</SheetNormal>
        <AbsDisplacement>true</AbsDisplacement>
        <Displacement>false</Displacement>
        <Velocity>false</Velocity>
        <Acceleration>false</Acceleration>
    </Options>
</Export>

<Solver>
    <NonZeros>2000</NonZeros>
    <DomainDecomposition>false</DomainDecomposition>
    <Precision>1e-12</Precision>
    <Verbose>true</Verbose>
    <LU>true</LU>
    <Type>NewmarkBeta</Type>
    <NewmarkBeta>
        <!-- Dissipation of higher frequencies for Gamma ≥ 0.5 -->
        <!-- Unconditional stability for 2⋆Beta ≥ Gamma ≥ 0.5 -->
        <Beta>0.3</Beta>
        <Gamma>0.6</Gamma>
        <ConsistentMassMatrix>true</ConsistentMassMatrix>
    </NewmarkBeta>
    <Formulation>TotalLagrangian</Formulation>
    <StartTime>0</StartTime>
    <StopTime>18.705</StopTime>
    <TimeStep>1e-3</TimeStep>

    <SNES>
        <Export>
            <Filename>Baseline_data/SNES.dat</Filename>
        </Export>
    </SNES>

    <Plugins>
        <acCELLerate>false</acCELLerate>
        <Circulation>true</Circulation>
        <LoadUnloadedState>true</LoadUnloadedState>
        <ContactHandling>true</ContactHandling>
    </Plugins>
</Solver>

<Plugins>
    <acCELLerate>
        <ProjectFile>./AccFiles/settings.aclt</ProjectFile>
        <acCELLerateMesh>./AccFiles/Geometry/Bar_LVL2.vtu</acCELLerateMesh>
        <MaterialFile>./AccFiles/materialIntra.def</MaterialFile>
        <ForceOffset>0.0</ForceOffset>
        <ResultPreFix>Acc</ResultPreFix>
        <ResultFolder>./Baseline_T4_CFD_Pressure/</ResultFolder>
        <PvdFileName>Acc</PvdFileName>
        <NumberOfQuadP>1</NumberOfQuadP>
        <Material>31</Material>
        <UpdateSysA>true</UpdateSysA>
        <minStretchVal>0.0</minStretchVal>
        <maxStretchVal>2.0</maxStretchVal>
        <constStretchRate>false</constStretchRate>
        <valStretchRate>0.0</valStretchRate>
        <Permute>false</Permute>
        <Export>true</Export>
    </acCELLerate>

    <LoadUnloadedState>
        <Nodes>/Users/jb072/00_Projects/CardioMechanics/04_CFD/Unloading_T4/UnloadedState_Cy
    </LoadUnloadedState>
    <ContactHandling>
        <Export>./Baseline_T4_CFD_Pressure/ContactHandling.dat</Export>
        <MaxDistance>0.05</MaxDistance>
        <TransitionDistance>1e-4</TransitionDistance>
        <MaxAngle>90</MaxAngle>
        <Alpha>1e7</Alpha>
        <Beta>1</Beta>
        <InitType>Linear</InitType>
        <Steps>5</Steps>
    </ContactHandling>

    <Circulation>
        <UsePressureGradient>false</UsePressureGradient>
        <PressureGradientFilePath>/Users/jb072/00_Projects/CardioMechanics/04_CFD/Phase

        <Tolerance>1e-7</Tolerance>
        <MaxIterations>200</MaxIterations>
        <Perturbation>false</Perturbation>
        <SecantIterations>0</SecantIterations>

        <PreloadingTime1>0.0</PreloadingTime1>
        <PreloadingTime2>0.0</PreloadingTime2>

        <PressureUnit>mmHg</PressureUnit>
        <VolumeUnit>ml</VolumeUnit>

        <CouplingDamping>
            <InitialFactor> 0.5 </InitialFactor>
            <DeclineFactor> 0.9 </DeclineFactor>
        </CouplingDamping>

        <Circs>

            <Circ_1>
                <Active>true</Active>
                <Type>CircWholeHeartValvesDynamic</Type>
                <ExportFile>./Baseline_T4_CFD_Pressure/Circ_1.dat</ExportFile>
                <MaxIntegrationTimeStep>1e-4</MaxIntegrationTimeStep>

                <CircParameters>
                    <BloodDensity>7.95e-4</BloodDensity>

                    <SysArtValveMax> 0.95 </SysArtValveMax>
                    <SysArtValveMin> 0.001 </SysArtValveMin>
                    <SysArtValveAreaRef> 7.0 </SysArtValveAreaRef>
                    <SysArtValveRateOpening> 20.0 ...
                        </SysArtValveRateOpening>
                    <SysArtValveRateClosing> 10.0 ...
                        </SysArtValveRateClosing>

                    <SysArtResist>0.07</SysArtResist>
                    <SysArtCompli>2.0</SysArtCompli>
                    <SysArtVolumeUnstr>800.0</SysArtVolumeUnstr>

                    <SysPerResist>0.9</SysPerResist>

                    <SysVenResist>0.03</SysVenResist>
                    <SysVenCompli>100.0</SysVenCompli>
                    <SysVenVolumeUnstr>2850.0</SysVenVolumeUnstr>

                    <RavValveMax> 0.7 </RavValveMax>
                    <RavValveMin> 0.001 </RavValveMin>
                    <RavValveAreaRef> 15.0 </RavValveAreaRef>
                    <RavValveRateOpening> 20.0 </RavValveRateOpening>
                    <RavValveRateClosing> 6.0 </RavValveRateClosing>

                    <PulArtValveMax> 0.98 </PulArtValveMax>
                    <PulArtValveMin> 0.001 </PulArtValveMin>
                    <PulArtValveAreaRef> 7.0 </PulArtValveAreaRef>
                    <PulArtValveRateOpening> 20.0 ...
                        </PulArtValveRateOpening>
                    <PulArtValveRateClosing> 10.0 ...
                        </PulArtValveRateClosing>

                    <PulArtResist>0.02</PulArtResist>
                    <PulArtCompli>10.0</PulArtCompli>
                    <PulArtVolumeUnstr>150.0</PulArtVolumeUnstr>

                    <PulPerResist>0.07</PulPerResist>

                    <PulVenResist>0.03</PulVenResist>
                    <PulVenCompli>15.0</PulVenCompli>
                    <PulVenVolumeUnstr>200.0</PulVenVolumeUnstr>

                    <LavValveMax> 0.7 </LavValveMax>
                    <LavValveMin> 0.001 </LavValveMin>
                    <LavValveAreaRef> 15.0 </LavValveAreaRef>
                    <LavValveRateOpening> 20.0 </LavValveRateOpening>
                    <LavValveRateClosing> 6.0 </LavValveRateClosing>
                </CircParameters>

                <InitialConditions>
                    <TotalVolume>5500.0</TotalVolume>
                    <SysArtVolume>959.733</SysArtVolume>
                    <PulArtVolume>279.737</PulArtVolume>
                    <PulVenVolume>311.339</PulVenVolume>
                    <LvPressure>7.5</LvPressure>
                    <RaPressure>4.0</RaPressure>
                    <RvPressure>4.0</RvPressure>
                    <LaPressure>7.5</LaPressure>
                </InitialConditions>

                <Cavities>
```

```
                    <LvSurface>131</LvSurface>
                    <RaSurface>132</RaSurface>
                    <RvSurface>130</RvSurface>
                    <LaSurface>133</LaSurface>

                    <LvPreloading>0.0</LvPreloading>
                    <RaPreloading>0.0</RaPreloading>
                    <RvPreloading>0.0</RvPreloading>
                    <LaPreloading>0.0</LaPreloading>
                </Cavities>
            </Circ_1>
        </Circs>
    </Circulation>

</Plugins>
```

Appendix C

Measurement time series

The mechanical movement of the endocardial surfaces for some characteristic points in the cardiac cycle are included in this section. While each heart beat is resolved with 30 frames, each fifth frame is included in the figure.

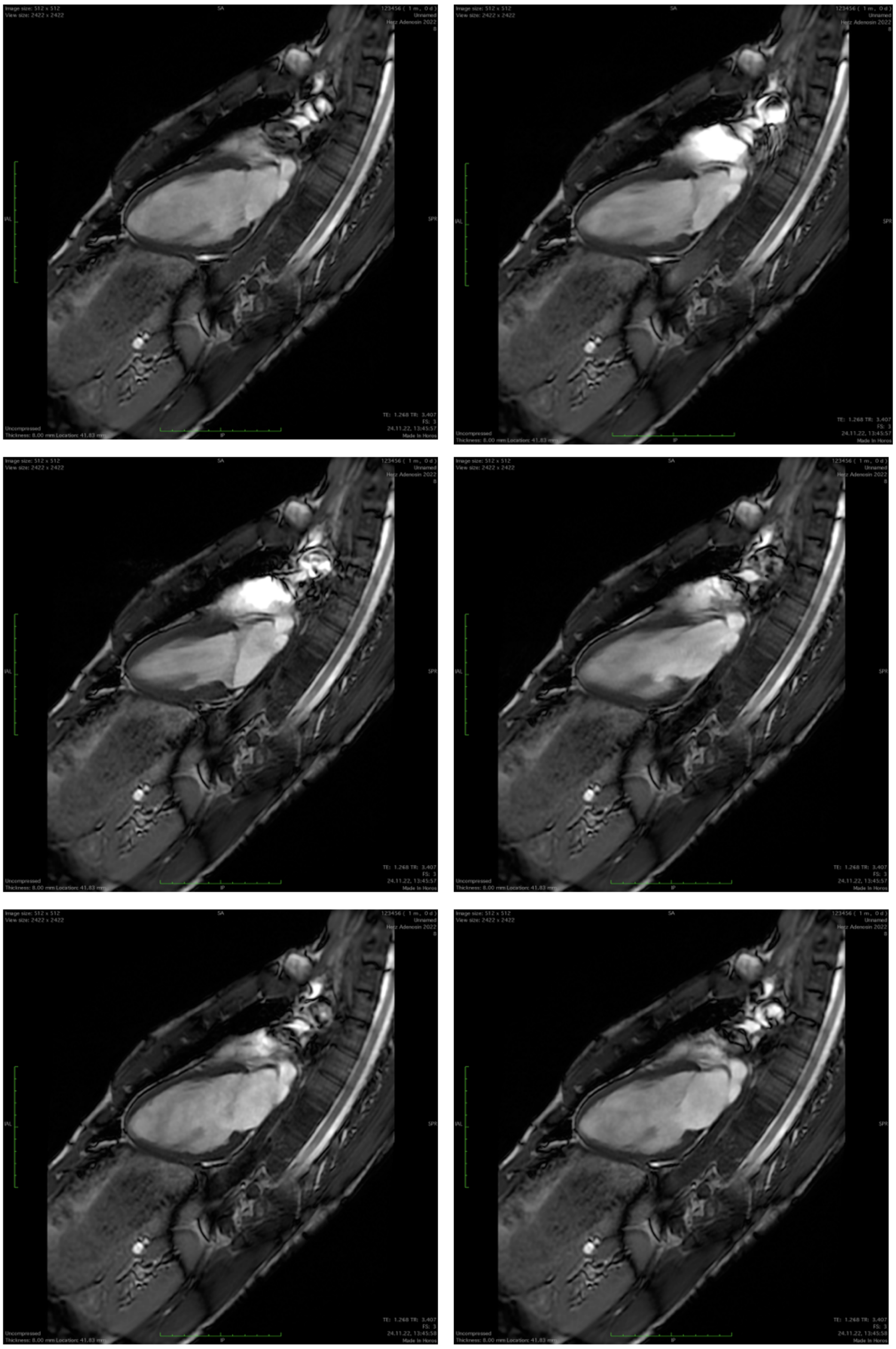


Figure C.1: Two chamber long axis view.

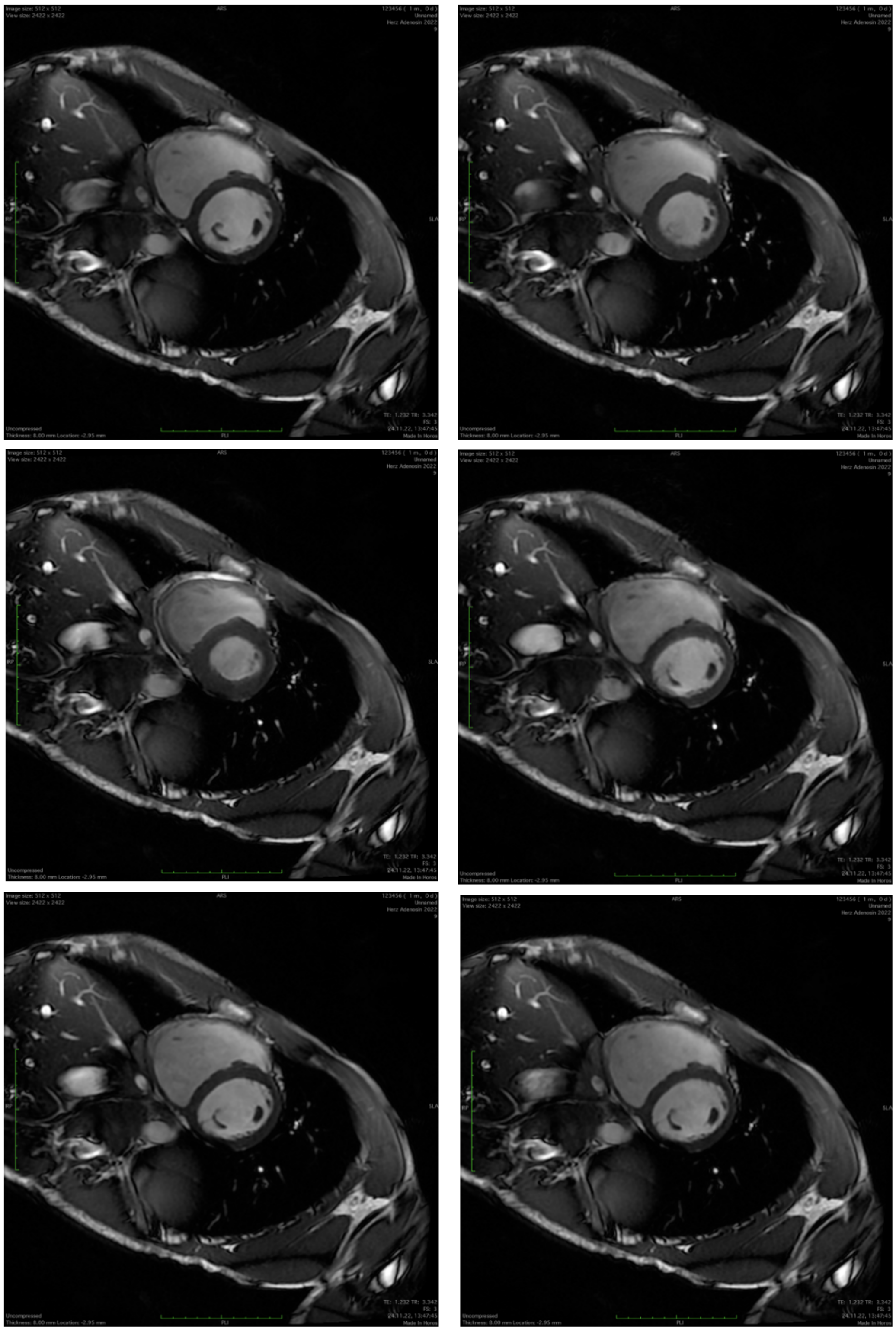


Figure C.2: Short axis view.

References

[1] H. Jasak, “OpenFOAM: Open source CFD in research and industry,” *International Journal of Naval Architecture and Ocean Engineering*, vol. 1, pp. 89–94, 2009. doi:10.2478/ijnaoe-2013-0011

[2] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” *Computers in Physics*, vol. 12, pp. 620–631, 1998. doi:10.1063/1.168744

[3] G. Clark, “Computational Modelling: Technological Futures,” Government Office for Science, Tech. Rep., 2018.

[4] C. B. Collin, T. Gebhardt, M. Golebiewski, et al., “Computational Models for Clinical Applications in Personalized Medicine—Guidelines and Recommendations for Data Integration and Model Validation,” *Journal of Personalized Medicine*, vol. 12, p. 166, 2022. doi:10.3390/jpm12020166

[5] T. Gatsos, K. A. Elsayed, Y. Zhai, and D. A. Lados, “Review on Computational Modeling of Process–Microstructure–Property Relationships in Metal Additive Manufacturing,” *JOM*, vol. 72, pp. 403–419, 2020. doi:10.1007/s11837-019-03913-x

[6] R. C. Wilson and A. G. Collins, “Ten simple rules for the computational modeling of behavioral data,” *eLife*, vol. 8, 2019. doi:10.7554/elife.49547

[7] J. I. Myung, Y. Tang, and M. A. Pitt, “Chapter 11 Evaluation and Comparison of Computational Models,” in *Methods in Enzymology*. Elsevier, 2009, pp. 287–304. doi:10.1016/s0076-6879(08)03811-1

[8] G. Wilson, J. Bryan, K. Cranston, J. Kitzes, L. Nederbragt, and T. K. Teal, “Good enough practices in scientific computing,” *PLoS computational biology*, vol. 13, p. e1005510, 2017. doi:10.1371/journal.pcbi.1005510

[9] S. A. Niederer, J. Lumens, and N. A. Trayanova, “Computational models in cardiology,” *Nat Rev Cardiol*, vol. 16, pp. 100–111, 2018. doi:10.1038/s41569-018-0104-y

[10] C. J. Greenshields and H. G. Weller, *Notes on Computational Fluid Dynamics*. Reading, UK: CFD Direct Ltd, 2022. https://openlibrary.org/books/OL38336756M

[11] R. Verzicco, “Electro-fluid-mechanics of the heart,” *Journal of Fluid Mechanics*, vol. 941, 2022. doi:10.1017/jfm.2022.272

[12] A. Quarteroni, Ed., *Modeling the Heart and the Circulatory System*, vol. 14. Switzerland: Springer International Publishing, 2015. doi:10.1007/978-3-319-05230-4

[13] A. Lintermann, “Computational Meshing for CFD Simulations,” *Clinical and Biomedical Engineering in the Human Nose*, pp. 85–115, 2021. doi:10.1007/978-981-15-6716-2_6

[14] J. Fish and T. Belytschko, *A first course in finite elements*. Wiley, 2007. http://openlibrary.org/book/OL9728110M

[15] F. Kong and S. C. Shadden, *Whole Heart Mesh Generation for Image-Based Computational Simulations by Learning Free-From Deformations*. Cham: Springer, 2021, vol. 12904. doi: https://doi.org/10.1007/978-3-030-87202-1_53

[16] A. Loewe, “Ein digitales Herz,” *Spektrum der Wissenschaft*, vol. 20, pp. 44–48, 2020.

[17] J. Mayourian, E. A. Sobie, and D. D. Costa, “An Introduction to Computational Modeling of Cardiac Electrophysiology and Arrhythmogenicity,” *Methods Mol Biol.*, vol. 1816, pp. 17–35, 2018.

[18] T. Jadczyk, G. Caluori, A. Loewe, and K. S. Golba, Eds., *Innovative Treatment Strategies for Clinical Electrophysiology*. Singapore: Springer Nature Singapore, 2022. doi:10.1007/978-981-19-6649-1

[19] A. Jafari, E. Pszczolkowski, and A. Krishnamurthy, “A framework for biomechanics simulations using four-chamber cardiac models.” *J Biomech*, vol. 91, pp. 92–101, 2019. doi:10.1016/j.jbiomech.2019.05.019

[20] V. Gurev, P. Pathmanathan, J.-L. Fattebert, et al., “A high-resolution computational model of the deforming human heart.” *Biomech Model Mechanobiol*, vol. 14, pp. 829–49, 2015. doi:10.1007/s10237-014-0639-8

[21] N. A. Trayanova, “Whole-Heart Modeling,” *Circulation Research*, vol. 108, pp. 113–128, 2011. doi:10.1161/circresaha.110.223610

[22] R. Moss, E. M. Wülfers, S. Schuler, A. Loewe, and G. Seemann, “A Fully-Coupled Electro-Mechanical Whole-Heart Computational Model: Influence of Cardiac Contraction on the ECG,” *Front. Physiol.*, vol. 12, 2021. doi:10.3389/fphys.2021.778872

[23] M. Albatat, D. R. King, L. A. Unger, et al., “Electromechanical Model to Predict Cardiac Resynchronization Therapy,” in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018. doi:10.1109/embc.2018.8513539

[24] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dede, and A. Quarteroni, “A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part II: numerical approximation,” *Numerical Analysis (math.NA)*, vol. Xiv:2011.15040v1, 2020.

[25] T. Gerach, S. Schuler, J. Fröhlich, et al., “Electro-Mechanical Whole-Heart Digital Twins: A Fully Coupled Multi-Physics Approach,” *Mathematics*, vol. 9, p. 1247, 2021. doi:10.3390/math9111247

[26] C. M. Augustin, M. A. F. Gsell, E. Karabelas, et al., “Validation of a 3D-0D closed-loop model of the heart and circulation – Modeling the experimental assessment of diastolic and systolic ventricular properties,” *arXiv Quantitative Biology*, 2020. arXiv:2009.08802v2

[27] E. J. Vigmond, C. Clements, D. M. McQueen, and C. S. Peskin, “Effect of bundle branch block on cardiac output: a whole heart simulation study,” *Prog Biophys Mol Biol*, vol. 97, pp. 520–542, 2008. doi:10.1016/j.pbiomolbio.2008.02.022

[28] A. Santiago, M. Zavala Aké, J. Aguado Sierra, et al., “Fully coupled fluid-electro-mechanical model of the human heart for supercomputers.” *Int J Numer Method Biomed Eng*, 2018. doi:10.1002/cnm.3140

[29] E. Karabelas, M. A. F. Gsell, C. M. Augustin, et al., “Towards a Computational Framework for Modeling the Impact of Aortic Coarctations Upon Left Ventricular Load.” *Front Physiol*, vol. 9, p. 538, 2018. doi:10.3389/fphys.2018.00538

[30] M. A. F. Gsell, C. M. Augustin, A. J. Prassl, et al., “Assessment of wall stresses and mechanical heart power in the left ventricle: Finite element modeling versus Laplace analysis.” *Int J Numer Method Biomed Eng*, p. e3147, 2018. doi:10.1002/cnm.3147

[31] F. Regazzoni, M. Salvador, P. C. Africa, M. Fedele, L. Dede, and A. Quarteroni, “A cardiac electromechanics model coupled with a lumped parameters model for closed-loop blood circulation. Part I: model derivation,” *arXiv*, 2020. arXiv:2011.15040

[32] V. Mihalef, R. I. Ionasec, P. Sharma, et al., “Patient-specific modelling of whole heart anatomy, dynamics and haemodynamics from four-dimensional cardiac CT images,” *Interface Focus*, vol. 1, pp. 286–296, 2011. doi:10.1098/rsfs.2010.0036

[33] N. A. Trayanova, A. N. Doshi, and A. Prakosa, “How personalized heart modeling can help treatment of lethal arrhythmias: A focus on ventricular tachycardia ablation strategies in post-infarction patients.” *Wiley Interdiscip Rev Syst Biol Med*, p. e1477, 2020. doi:10.1002/wsbm.1477

[34] S. Galappaththige, R. A. Gray, C. M. Costa, S. Niederer, and P. Pathmanathan, “Credibility assessment of patient-specific computational modeling using patient-specific cardiac modeling as an exemplar,” *PLOS Computational Biology*, vol. 18, p. e1010541, 2022. doi:10.1371/journal.pcbi.1010541

[35] Food and Drug Administration, “Assessing the Credibility of Computational Modeling and Simulation in Medical Device Submissions,” U.S. Department of Health and Human Services, Tech. Rep., 2021. https://www.fda.gov/media/154985/download

[36] R. Mittal, J. H. Seo, V. Vedula, et al., “Computational modeling of cardiac hemodynamics: Current status and future outlook,” *Journal of Computational Physics*, vol. 305, pp. 1065–1082, 2016. doi:10.1016/j.jcp.2015.11.022

[37] A. Zingaro, L. Dede’, F. Menghini, and A. Quarteroni, “Hemodynamics of the heart’s left atrium based on a Variational Multiscale-LES numerical model,” *European Journal of Mechanics - B/Fluids*, 2021. doi:https://doi.org/10.1016/j.euromechflu.2021.06.014

[38] A. Daub, J. Kriegseis, and B. Frohnapfel, “Replication of left ventricular haemodynamics with a simple planar mitral valve model,” *Biomed Tech (Berl)*, 2020. doi:10.1515/bmt-2019-0175

[39] J. Brenneisen, C. Wentzel, F. Karwan, O. Dössel, and A. Loewe, “Fluid dynamics in the human heart: Altered vortex formation and wash-out in mitral regurgitation simulations,” in *Current Directions in Biomedical Engineering*, vol. 7, no. 2. De Gruyter, 2021, pp. 199–202. doi:10.1515/cdbme-2021-2051

[40] J. Brenneisen, A. Daub, T. Gerach, et al., “Sequential Coupling Shows Minor Effects of Fluid Dynamics on Myocardial Deformation in a Realistic Whole-Heart Model.” *Front Cardiovasc Med*, vol. 8, p. 768548, 2021. doi:10.3389/fcvm.2021.768548

[41] O. Evju and K.-A. Mardal, *On the Assumption of Laminar Flow in Physiological Flows: Cerebral Aneurysms as an Illustrative Example*, MS&A, vol. 14. Switzerland: Springer International Publishing, 2015. doi:10.1007/978-3-319-05230-4_7

[42] W. Sun, C. Martin, and T. Pham, “Computational Modeling of Cardiac Valve Function and Intervention,” *Annual Review of Biomedical Engineering*, vol. 16, pp. 53–76, 2014. doi:10.1146/annurev-bioeng-071813-104517

[43] F. Sotiropoulos and I. Borazjani, “A review of state-of-the-art numerical methods for simulating flow through mechanical heart valves.” *Med Biol Eng Comput*, vol. 47, pp. 245–56, 2009. doi:10.1007/s11517-009-0438-z

[44] I. S. Lan, J. Liu, W. Yang, J. Zimmermann, D. B. Ennis, and A. L. Marsden, "Validation of the Reduced Unified Continuum Formulation Against In Vitro 4D-Flow MRI," *eprint*, 2022. arXiv:2203.00721v1

[45] C. P. Blanken, L. M. Gottwald, J. J. Westenberg, et al., "Whole-Heart 4D Flow MRI for Evaluation of Normal and Regurgitant Valvular Flow: A Quantitative Comparison Between Pseudo-Spiral Sampling and EPI Readout," *Journal of Magnetic Resonance Imaging*, vol. 55, pp. 1120–1130, 2022. doi:10.1002/jmri.27905

[46] M. Hirschhorn, V. Tchantchaleishvili, R. Stevens, J. Rossano, and A. Throckmorton, "Fluid-structure interaction modeling in cardiovascular medicine - A systematic review 2017-2019." *Med Eng Phys*, vol. 78, pp. 1–13, 2020. doi:10.1016/j.medengphy.2020.01.008

[47] S. Bhakade, K. Surajkumar, Y. Mohite, and P. Kengar, "A Review on Fluid Structure Interaction Analysis Methodology," *International Journal of Trend in Research and Development*, vol. 3, pp. 617–619, 2016.

[48] E. H. Dowell and K. C. Hall, "Modeling of Fluid-Structure Interaction," *Annual Review of Fluid Mechanics*, vol. 33, pp. 445–490, 2001. doi:10.1146/annurev.fluid.33.1.445

[49] G. Hou, J. Wang, and A. Layton, "Numerical Methods for Fluid-Structure Interaction — A Review," *Communications in Computational Physics*, vol. 12, pp. 337–377, 2012. doi: 10.4208/cicp.291210.290411s

[50] R. Courant, K. Friedrichs, and H. Lewy, "Über die partiellen Differenzengleichungen der mathematischen Physik," *Math Ann*, vol. 100, pp. 32–74, 1928. doi:https://doi.org/10.1007/BF01448839

[51] J. Degroote, K.-J. Bathe, and J. Vierendeels, "Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction," *Computers & Structures*, vol. 87, pp. 793–801, 2009. doi:10.1016/j.compstruc.2008.11.013

[52] M. A. Fernández, "Coupling schemes for incompressible fluid-structure interaction: implicit, semi-implicit and explicit," *SeMA Journal*, vol. 55, pp. 59–108, 2013. doi:10.1007/BF03322593

[53] F. Viola, V. Meschini, and R. Verzicco, "Fluid–Structure-Electrophysiology interaction (FSEI) in the left-heart: A multi-way coupled computational model," *European Journal of Mechanics - B/Fluids*, vol. 79, pp. 212–232, 2020. doi:10.1016/j.euromechflu.2019.09.006

[54] S. S. Abbas, M. S. Nasif, and R. Al Waked, "State-of-the-art numerical fluid–structure interaction methods for aortic and mitral heart valves simulations: A review," *SIMULATION*, p. 003754972110235, 2021. doi:10.1177/00375497211023573

[55] B. E. Griffith and N. A. Patankar, "Immersed Methods for Fluid-Structure Interaction," *Annual Review of Fluid Mechanics*, vol. 52, pp. 421–448, 2020. doi:10.1146/annurev-fluid-010719-060228

[56] C. S. Peskin, "Flow patterns around heart valves: A numerical method," *Journal of Computational Physics*, vol. 10, pp. 252–271, 1972. doi:10.1016/0021-9991(72)90065-4

[57] R. Mittal and G. Iaccarino, "Immersed Boundary Methods," *Annual Review of Fluid Mechanics*, vol. 37, pp. 239–261, 2005. doi:10.1146/annurev.fluid.37.061903.175743

[58] F. Sotiropoulos and X. Yang, "Immersed boundary methods for simulating fluid-structure interaction," *Progress in Aerospace Sciences*, vol. 65, pp. 1–21, 2014. doi:10.1016/j.paerosci.2013.09.003

[59] S.-G. Cai, A. Ouahsine, and Y. Hoarau, "Moving immersed boundary method for fluid–solid interaction," *Physics of Fluids*, vol. 34, p. 053307, 2022. doi:10.1063/5.0088302

[60] W.-Q. Wang, Y. Yan, and F.-B. Tian, “A simple and efficient implicit direct forcing immersed boundary model for simulations of complex flow,” *Applied Mathematical Modelling*, vol. 43, pp. 287–305, 2017. doi:10.1016/j.apm.2016.10.057

[61] F. Xu and S. Kenjereš, “Numerical simulations of flow patterns in the human left ventricle model with a novel dynamic mesh morphing approach based on radial basis function.” *Comput Biol Med*, vol. 130, p. 104184, 2021. doi:10.1016/j.compbiomed.2020.104184

[62] D. Goldstein, R. Handler, and L. Sirovich, “Modeling a no-slip flow boundary with an external force field,” *Journal of Computational Physics (ISSN 0021-9991)*, vol. 105, pp. 354–366, 1993. doi:10.1006/jcph.1993.1081

[63] Y. Kametani, Y. Fukuda, T. Osawa, and Y. Hasegawa, “A new framework for design and validation of complex heat transfer surfaces based on adjoint optimization and rapid prototyping technologies,” *Journal of Thermal Science and Technology*, vol. 15, pp. JTST0016–JTST0016, 1970. doi:10.1299/jtst.2020jtst0016

[64] B. E. Griffith, X. Luo, D. M. McQueen, and C. S. Peskin, “Simulating the Fluid Dynamics of Natural and Prosthetic Heart Valves Using the Immersed Boundary Method,” *International Journal of Applied Mechanics*, vol. 1, p. 137, 2009. doi:10.1142/S1758825109000113

[65] Y. Cheng and H. Zhang, “Immersed boundary method and lattice Boltzmann method coupled FSI simulation of mitral leaflet flow,” *Computers & Fluids*, vol. 39, pp. 871–881, 2010. doi:10.1016/j.compfluid.2010.01.003

[66] J. Adu, L. Yin, H. Zhang, S. Xie, and J. Lu, “Simulation of the dynamic flow field in the left ventricle of the heart during diastolic filling,” *AIP Advances*, vol. 10, p. 025221, 2020. doi:10.1063/1.5126470

[67] Y. Kim, W. Lee, and E. Jung, “An Immersed Boundary Heart Model Coupled with a Multicompartment Lumped Model of the Circulatory System,” *SIAM Journal on Scientific Computing*, vol. 32, pp. 1809–1831, 2010. doi:10.1137/090761963

[68] W.-B. Tay, Y.-H. Tseng, L.-Y. Lin, and W.-Y. Tseng, “Towards patient-specific cardiovascular modeling system using the immersed boundary technique,” *BioMedical Engineering OnLine*, vol. 10, 2011. doi:10.1186/1475-925X-10-52

[69] D. M. McQueen and C. S. Peskin, “A three-dimensional computer model of the human heart for studying cardiac fluid dynamics,” *ACM SIGGRAPH Computer Graphics*, vol. 34, pp. 56–60, 2000. doi:10.1145/563788.604453

[70] D. M. McQueen, C. S. Peskin, and L. Zhu, “The Immersed Boundary Method for incompressible fluid-structure interaction,” *Computational Fluid and Solid Mechanics*, pp. 26–30, 2001. doi:10.1016/B978-008043944-0/50560-6

[71] D. M. McQueen and C. S. Peskin, “Heart Simulation by an Immersed Boundary Method with Formal Second-order Accuracy and Reduced Numerical Viscosity,” in *Mechanics for a New Mellennium*, H. Aref and J. Phillips, Eds. Springer Netherlands, 2002, pp. 429–444. doi:10.1007/0-306-46956-1_27

[72] W. H. Ibrahim, “Recent advances and controversies in adult cardiopulmonary resuscitation,” *Postgraduate Medical Journal*, vol. 83, pp. 649–654, 2007. doi:10.1136/pgmj.2007.057133

[73] S. Fowler, R. Roush, and J. Wise, *Concepts of biology*. Houston, Texas: OpenStax College, 2013. https://openlibrary.org/books/OL26409872M

[74] J. R. Levick, *Introduction to Cardiovascular Physiology*. London: Taylor & Francis Group, 2009. https://openlibrary.org/books/OL28745035M

[75] J. G. Betts, P. DeSaix, E. Johnson, et al., *Anatomy and Physiology*. Huston, Texas: OpenStax, Rice University, 2013. http://openlibrary.org/book/OL27930559M

[76] S. L. Murphy, K. D. Kochanek, J. Xu, and E. Arias, “Mortality in the United States, 2020.” *NCHS Data Brief*, pp. 1–8, 2021. https://www.ncbi.nlm.nih.gov/pubmed/34978528

[77] J. P. Dal Bianco, J. Beaudoin, M. D. Handschumacher, and R. A. Levine, “Basic Mechanisms of Mitral Regurgitation,” *Canadian Journal of Cardiology*, vol. 30, pp. 971–981, 2014. doi:10.1016/j.cjca.2014.06.022

[78] H. Roskam and H. Reindell, *Herzkrankheiten*, vol. 4. Berlin, Heidelberg, New York: Springer, 1996. https://openlibrary.org/books/OL12777014M

[79] J. P. Singh, J. C. Evans, D. Levy, et al., “Prevalence and clinical determinants of mitral, tricuspid, and aortic regurgitation (the Framingham Heart Study),” *The American Journal of Cardiology*, vol. 83, pp. 897–902, 1999. doi:10.1016/s0002-9149(98)01064-9

[80] F. Bursi, M. Enriquez Sarano, S. J. Jacobsen, and V. L. Roger, “Mitral Regurgitation After Myocardial Infarction: A Review,” *The American Journal of Medicine*, vol. 119, pp. 103–112, 2006. doi:10.1016/j.amjmed.2005.08.025

[81] K. Fattouch, P. Lancellotti, and G. D. Angelini, Eds., *Secondary Mitral Valve Regurgitation*. London: Springer London, 2015. doi:10.1007/978-1-4471-6488-3

[82] P. Lancellotti, L. Moura, L. A. Pierard, et al., “European Association of Echocardiography recommendations for the assessment of valvular regurgitation. Part 2: mitral and tricuspid regurgitation (native valve disease).” *Eur J Echocardiogr*, vol. 11, pp. 307–32, 2010. doi:10.1093/ejechocard/jeq031

[83] R. Larsen, *Anästhesie und Intensivmedizin in Herz-, Thorax- und Gefäßchirurgie*. Springer Verlag, 1999. https://openlibrary.org/books/OL9608988M

[84] A. J. Marian and E. Braunwald, “Hypertrophic Cardiomyopathy,” *Circulation Research*, vol. 121, pp. 749–770, 2017. doi:10.1161/circresaha.117.311059

[85] P. M. Elliott, A. Anastasakis, M. Borgers, and M. Borggrefe, “2014 ESC Guidelines on diagnosis and management of hypertrophic cardiomyopathy,” *European Heart Journal*, vol. 35, pp. 2733–2779, 2014. doi:10.1093/eurheartj/ehu284

[86] R. Ariga, E. M. Tunnicliffe, S. G. Manohar, et al., “Identification of Myocardial Disarray in Patients With Hypertrophic Cardiomyopathy and Ventricular Arrhythmias.” *J Am Coll Cardiol*, vol. 73, pp. 2493–2502, 2019. doi:10.1016/j.jacc.2019.02.065

[87] J. Schönberger and G. Ertl, “Monogenetische Herzerkrankungen,” *Medizinische Klinik*, vol. 103, pp. 166–174, 2008. doi:10.1007/s00063-008-1024-0

[88] J. Milei and G. Ambrosio, *Cardiomyopathies*. InTechOpen, 2013. doi:10.5772/56744

[89] C. J. McLeod, J. M. Bos, J. L. Theis, et al., “Histologic characterization of hypertrophic cardiomyopathy with and without myofilament mutations,” *American Heart Journal*, vol. 158, pp. 799–805, 2009. doi:10.1016/j.ahj.2009.09.006

[90] N. Hensley, J. Dietrich, D. Nyhan, N. Mitter, M.-S. Yee, and M. Brady, “Hypertrophic cardiomyopathy: a review.” *Anesth Analg*, vol. 120, pp. 554–69, 2015. doi:10.1213/ANE.0000000000000538

[91] G. Schuler, “19. Hypertrophe Kardiomyopathie,” in *Körperliche Aktivität und Krankheit*. De Gruyter, 2017, pp. 301–313. doi:10.1515/9783110456783-019

[92] G. Mattsson and P. Magnusson, *Cardiomyopathy - Disease of the Heart Muscle*. IntechOpen, 2021. doi:10.5772/intechopen.91489

[93] M. Markl, A. Frydrychowicz, S. Kozerke, M. Hope, and O. Wieben, “4D flow MRI,” *Journal of Magnetic Resonance Imaging*, vol. 36, pp. 1015–1036, 2012. doi:10.1002/jmri.23632

[94] P. Dyverfeldt, M. Bissell, A. J. Barker, et al., “4D flow cardiovascular magnetic resonance consensus statement,” *Journal of Cardiovascular Magnetic Resonance*, vol. 17, 2015. doi:10.1186/s12968-015-0174-5

[95] K. S. Nayak, J.-F. Nielsen, M. A. Bernstein, et al., “Cardiovascular magnetic resonance phase contrast imaging,” *Journal of Cardiovascular Magnetic Resonance*, vol. 17, 2015. doi:10.1186/s12968-015-0172-7

[96] G. Soulat, P. McCarthy, and M. Markl, “4D Flow with MRI,” *Annual Review of Biomedical Engineering*, vol. 22, pp. 103–126, 2020. doi:10.1146/annurev-bioeng-100219-110055

[97] T. A. Gallagher, A. J. Nemeth, and L. Hacein Bey, “An Introduction to the Fourier Transform: Relationship to MRI,” *American Journal of Roentgenology*, vol. 190, pp. 1396–1405, 2008. doi:10.2214/ajr.07.2874

[98] O. Dössel, *Bildgebende Verfahren in der Medizin - Von der Technik zur medizinischen Anwendung*, vol. 2. Auflage. Berlin Heidelberg: Springer Vieweg, 2016. https://openlibrary.org/books/OL27986247M

[99] D. T. Wymer, K. P. Patel, W. F. Burke, and V. K. Bhatia, “Phase-Contrast MRI: Physics, Techniques, and Clinical Applications,” *RadioGraphics*, vol. 40, pp. 122–140, 2020. doi:10.1148/rg.2020190039

[100] A. Sträter, A. Huber, J. Rudolph, et al., “4D-Flow MRI: Technique and Applications,” *RöFo - Fortschritte auf dem Gebiet der Röntgenstrahlen und der bildgebenden Verfahren*, vol. 190, pp. 1025–1035, 2018. doi:10.1055/a-0647-2021

[101] B. Zhuang, A. Sirajuddin, S. Zhao, and M. Lu, “The role of 4D flow MRI for clinical applications in cardiovascular disease: current status and future perspectives,” *Quantitative Imaging in Medicine and Surgery*, vol. 11, pp. 4193–4210, 2021. doi:10.21037/qims-20-1234

[102] A. Natale, P. J. Al Ahmad, and J. DiMarco, *Cardiac Electrophysiology. Clinical Case Review*. London: Springer London Ltd, 2011. https://openlibrary.org/books/OL25565434M

[103] O. Dössel and G. Seemann, “Computer model of the electrical excitation of the heart,” in *Modelling and Control in Biomedical Systems. A Proceedings Volume from the 5th IFAC Symposium Hilton on the Park, Melbourne, Australia, 21-23 August*, D. D. Feng and E. R. Carson, Eds., Oxford: Pergamon, 2003, pp. 179–184. doi:10.1016/S1474-6670(17)33496-1

[104] J. H. Ferziger, M. Peric, and R. L. Street, *Numerische Strömungsmechanik*. Berlin: Springer, 2020. http://openlibrary.org/book/OL29216428M

[105] P. K. Kundu, I. M. Cohen, and D. R. Dowling, *Fluid Mechanics*. Oxford: Elsevier Science & Technology Books, 2016. https://openlibrary.org/books/OL28555722M

[106] F. Durst, *Fluid Mechanics*. Berlin, Heidelberg: Springer Berlin, Heidelberg, 2022. doi:10.1007/978-3-662-63915-3

[107] A. C. Daub, *Numerical Haemodynamics in the Human Heart*. Karlsruhe: KIT Scientific Publishing, 2018. doi:10.5445/KSP/1000080044

[108] I. Demirdzic and M. Peric, “Space conservation law in finite volume calculations of fluid flow,” *International Journal for Numerical Methods in Fluids (ISSN 0271-2091)*, vol. 8, pp. 1037–1050, 1988. doi:10.1002/fld.1650080906

[109] T. Belytschko, W. Kam Liu, and B. Moran, *Nonlinear Finite Elements for Continua and Structures*. Wiley, 2000.

[110] T. Fritz, “Biomechanical Modeling of the Human Heart - Modeling of the Ventricles, the Atria and the Pericardium and the Inverse Problem of Cardiac Mechanics,” PhD thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), Karlsruhe, 2015.

[111] E. Kovacheva, “Model Based Estimation of the Elastomechanical Properties of the Human Heart,” PhD thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2021. doi:10.5445/IR/1000135416

[112] T. Gerach, “Personalized Electromechanical Modeling of the Human Heart: Challenges and Opportunities for the Simulation of Pathophysiological Scenarios,” PhD thesis, Karlsruher Institut für Technologie (KIT), KITopen, 2022. doi:10.5445/IR/1000147806

[113] J. M. Guccione, A. D. McCulloch, and L. K. Waldman, “Passive material properties of intact ventricular myocardium determined from a cylindrical model.” *J Biomech Eng*, vol. 113, pp. 42–55, 1991. doi:10.1115/1.2894084

[114] J. D. Bayer, R. C. Blake, G. Plank, and N. A. Trayanova, “A novel rule-based algorithm for assigning myocardial fiber orientation to computational heart models,” *Ann Biomed Eng*, vol. 40, pp. 2243–2254, 2012. doi:10.1007/s10439-012-0593-5

[115] N. Stergiopulos, J. J. Meister, and N. Westerhof, “Determinants of stroke volume and systolic and diastolic aortic pressure.” *Am J Physiol*, vol. 270, pp. H2050–9, 1996. doi:10.1152/ajpheart.1996.270.6.H2050

[116] J. Bols, J. Degroote, B. Trachet, B. Verhegghe, 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*, vol. 246, pp. 10–17, 2013. doi:10.1016/j.cam.2012.10.034

[117] J. Brenneisen, S. Schuler, E. Kovacheva, T. Gerach, O. Dössel, and A. Loewe, “Influence of Geometrical Properties for the Calculation of a Pressure-Free Whole Heart Geometry,” in *14th WCCM-ECCOMAS Congress 2020*, F. Chinesta, R. Abgrall, O. Allix, and M. Kaliske, Eds. Scipedia, 2021, pp. 1–9. doi:10.23967/wccm-eccomas.2020.177

[118] L. Marx, J. A. Niestrawska, M. A. Gsell, F. Caforio, G. Plank, and C. M. Augustin, “Robust and efficient fixed-point algorithm for the inverse elastostatic problem to identify myocardial passive material parameters and the unloaded reference configuration,” *Journal of Computational Physics*, p. 111266, 2022. doi:10.1016/j.jcp.2022.111266

[119] S. Klotz, I. Hay, M. L. Dickstein, et al., “Single-beat estimation of end-diastolic pressure-volume relationship: a novel method with potential for noninvasive application.” *Am J Physiol Heart Circ Physiol*, vol. 291, pp. H403–12, 2006. doi:10.1152/ajpheart.01240.2005

[120] 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,” *Biomech Model Mechanobiol*, vol. 13, pp. 627–641, 2014. doi:10.1007/s10237-013-0523-y

[121] S. Schuler, L. Baron, A. Loewe, and O. Dössel, “Developing and coupling a lumped element model of the closed loop human vascular system to a model of cardiac mechanics,” in *BMTMedPhys 2017*, vol. 62, no. S1. Dresden: de Gruyter, 2017, p. S69.

[122] 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), 2016.

[123] D. Garcia, P. Pibarot, and L.-G. Durand, “Analytical modeling of the instantaneous pressure gradient across the aortic valve.” *J Biomech*, vol. 38, pp. 1303–11, 2005. doi:10.1016/j.jbiomech.2004.06.018

[124] M. Weiß, “Bewertung und anpassung der mitralklappenimplementierung in einem parametrischen kreislaufmodell basierend auf fluiddynamischen simulationen,” Master’s thesis, Institut für Biomedizinische Technik, Karlsruher Institut für Technologie (KIT), 2021.

[125] J. Brenneisen, D. Müller, A. Stroh, B. Frohnapfel, O. Dössel, and A. Loewe, “Cardiac fluid dynamics based on immersed boundary method for application in hypertrophic cardiomyopathy,” in *7th International Conference on Computational & Mathematical Biomedical Engineering - CMBE2021*, P. Nithiarasu and C. Vergara, Eds., Cardiff, UK, 2022, pp. 439 – 442.

[126] D. Müller, “Adjoint-based geometry optimization for enhancement on thermohydraulic efficiency in wall-bounded flows,” Master’s thesis, Institute of Fluid Mechanics, Karlsruhe Institute of Technology (KIT), 2021.

[127] J. Betz, “Implementierung und Analyse eines Computermodells der Mitralklappe basierend auf der Immersed Boundary Methode,” Bachelor’s thesis, Institut für Strömungsmechanik (ISTM), Institut für Biomedizinische Technik (IBT), Karlsruher Institut für Technologie (KIT), Karlsruhe, 2021.

[128] J. Ingels, B. Neil, and M. Karlsson, *Mitral Valve Mechanics*. Linköping University Electronic Press, 2015. doi:10.3384/book.diva-117057

[129] C. W. Lillehei, “The St. Jude Medical Prosthetic Heart Valve: Results from a Five-Year Multicenter Experience,” in *Update in Heart Valve Replacement*. Heidelberg: Steinkopff, 1986, pp. 3–18. doi:10.1007/978-3-662-10713-3_2

[130] A. Rodríguez de Castro, “A review on the porous medium approaches to model the flow of interdendritic liquid during the solidification of alloys in casting processes: theory and experiments,” *The International Journal of Advanced Manufacturing Technology*, vol. 107, pp. 4097–4121, 2020. doi:10.1007/s00170-020-05241-w

[131] J. Eriksson, C. J. Carlhäll, P. Dyverfeldt, J. Engvall, A. F. Bolger, and T. Ebbers, “Semi-automatic quantification of 4d left ventricular blood flow.” *J Cardiovasc Magn Reson*, vol. 12, p. 9, 2010. doi:10.1186/1532-429X-12-9

[132] V. M. Stoll, A. T. Hess, C. T. Rodgers, et al., “Left ventricular flow analysis.” *Circ Cardiovasc Imaging*, vol. 12, p. e008130, 2019. doi:10.1161/CIRCIMAGING.118.008130

[133] J. Brenneisen, O. Dössel, and A. Loewe, “Influence of pressure boundary condition definition on flow patterns in cardiac simulations,” *Modeling the Cardiac Function*, 2022.

[134] H. Schlichting and K. Gersten, *Grenzschicht-Theorie*. Berlin/Heidelberg: Springer-Verlag, 2006. doi:10.1007/3-540-32985-4

[135] L. Gepstein, G. Hayam, S. Shpun, and S. A. Ben Haim, “Hemodynamic evaluation of the heart with a nonfluoroscopic electromechanical mapping technique,” *Circulation*, vol. 96, pp. 3672–3680, 1997.

[136] C. Wentzel, “Analysis of stenosis and regurgitation as effects of valve defects on the fluid dynamics in the left heart,” Bachelor’s thesis, Institute of Fluid Mechanics, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2020.

[137] F. Karwan, “Simulation und Analyse der Blutflusskomponenten im linken Ventrikel des menschlichen Herzens mittels numerischer Strömungsmechanik,” Bachelor’s thesis, Institut für Biomedizinische Technik, Karlsruher Institut für Technologie (KIT), Karlsruhe, 2021.

[138] A. Vahanian, F. Beyersdorf, F. Praz, et al., “Corrigendum to: 2021 ESC/EACTS Guidelines for the management of valvular heart disease: Developed by the Task Force for the management of valvular heart disease of the European Society of Cardiology (ESC) and the European Association for Cardio-Thoracic Surgery (EACTS),” *European Heart Journal*, 2022. doi: 10.1093/eurheartj/ehac051

[139] M. Souli and J. P. Zolesio, “Arbitrary Lagrangian-Eulerian and free surface methods in fluid mechanics,” *Computer Methods in Applied Mechanics and Engineering*, vol. 191, pp. 451–466, 2001. doi:10.1016/S0045-7825(01)00313-9

[140] S. Irmay, “On the theoretical derivation of Darcy and Forchheimer formulas,” *Transactions, American Geophysical Union*, vol. 39, p. 702, 1958. doi:10.1029/tr039i004p00702

[141] A. F. Bolger, E. Heiberg, M. Karlsson, et al., “Transit of blood flow through the human left ventricle mapped by cardiovascular magnetic resonance.” *J Cardiovasc Magn Reson*, vol. 9, pp. 741–7, 2007. doi:10.1080/10976640701544530

[142] D. Collia, L. Zovatto, and G. Pedrizzetti, “Analysis of mitral valve regurgitation by computational fluid dynamics.” *APL Bioeng*, vol. 3, p. 036105, 2019. doi:10.1063/1.5097245

[143] M. Bonini, M. Hirschvogel, Y. Ahmed, et al., “Hemodynamic Modeling for Mitral Regurgitation,” *The Journal of Heart and Lung Transplantation*, vol. 41, p. S218, 2022. doi:10.1016/j.healun.2022.01.1685

[144] J. Eriksson, A. F. Bolger, T. Ebbers, and C.-J. Carlhäll, “Four-dimensional blood flow-specific markers of LV dysfunction in dilated cardiomyopathy,” *European Heart Journal - Cardiovascular Imaging*, vol. 14, pp. 417–424, 2013. doi:10.1093/ehjci/jes159

[145] J. Brenneisen, E. Kovacheva, T. Gerach, et al., “Characterization of the Fluid Dynamic Pressure Field in the Human Heart as a Basis for Coupled Fluid-Structure Simulations,” in *BMT 2020*, 54th Annual Conference of the German Society for Biomedical Engineering, vol. Poster Session. Berlin: De Gruyter, 2020, p. 259.

[146] W. F. Sherman and A. Grosberg, “Exploring cardiac form and function: A length-scale computational biology approach.” *Wiley Interdiscip Rev Syst Biol Med*, vol. 12, p. e1470, 2020. doi:10.1002/wsbm.1470

[147] A. Quarteroni, T. Lassila, S. Rossi, and R. Ruiz Baier, “Integrated Heart—Coupling multiscale and multiphysics models for the simulation of the cardiac function,” *Comput Methods Appl Mech Engrg*, vol. 314, pp. 345–407, 2017. doi:10.1016/j.cma.2016.05.031

[148] K. Miller, “Doing the heart good: Translating models to the clinic,” *Biomedical Computation Review*, 2014.

[149] A. G. Brown, Y. Shi, A. Marzo, et al., “Accuracy vs. computational time: translating aortic simulations to the clinic.” *J Biomech*, vol. 45, pp. 516–23, 2012. doi:10.1016/j.jbiomech.2011.11.041

[150] F. Sacco, B. Paun, O. Lehmkuhl, et al., “Left Ventricular Trabeculations Decrease the Wall Shear Stress and Increase the Intra-Ventricular Pressure Drop in CFD Simulations,” *Front. Physiol.*, vol. 9, pp. 1–15, 2018. doi:https://doi.org/10.3389/fphys.2018.00458

[151] R. E. Peverill, “Understanding preload and preload reserve within the conceptual framework of a limited range of possible left ventricular end-diastolic volumes.” *Adv Physiol Educ*, vol. 44, pp. 414–422, 2020. doi:10.1152/advan.00043.2020

[152] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding, “Two calculation procedures for steady, three-dimensional flows with recirculation,” *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, vol. 19, p. 60, 1970. doi:10.1007/BFb0112677

[153] L. Wang, L.-P. Wang, Z. Guo, and J. Mi, “Volume-averaged macroscopic equation for fluid flow in moving porous media,” *International Journal of Heat and Mass Transfer*, vol. 82, pp. 357–368, 2015. doi:10.1016/j.ijheatmasstransfer.2014.11.056

[154] H. Asgharzadeh and I. Borazjani, “Effects of Reynolds and Womersley Numbers on the Hemodynamics of Intracranial Aneurysms.” *Comput Math Methods Med*, vol. 2016, p. 7412926, 2016. doi:10.1155/2016/7412926

[155] T. Schenkel, M. Malve, M. Reik, M. Markl, B. Jung, and H. Oertel, “MRI-based CFD analysis of flow in a human left ventricle: methodology and application to a healthy heart.” *Ann Biomed Eng*, vol. 37, pp. 503–15, 2009. doi:10.1007/s10439-008-9627-4

[156] L. Hütter, “An integrated heart model: Aspects of coupling fluid dynamics, elastomechanics and electrophysiology,” Master’s thesis, Institute of Fluid Mechanics, Univ. Politècnica de València; Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2019.

[157] C. Habchi, S. Russeil, D. Bougeard, et al., “Partitioned solver for strongly coupled fluid–structure interaction,” *Computers & Fluids*, vol. 71, pp. 306–319, 2013. doi:10.1016/j.compfluid.2012.11.004

[158] R. Piersanti, F. Regazzoni, M. Salvador, et al., “3D-0D closed-loop model for the simulation of cardiac biventricular electromechanics,” *Comput Meth Appl Mech Eng*, 2021. arXiv:2108.01907v1

[159] C. M. Augustin, M. A. Gsell, E. Karabelas, et al., “A computationally efficient physiologically comprehensive 3D–0D closed-loop model of the heart and circulation,” *Computer Methods in Applied Mechanics and Engineering*, vol. 386, p. 114092, 2021. doi:10.1016/j.cma.2021.114092

[160] A. M. Bavo, A. M. Pouch, J. Degroote, et al., “Patient-specific CFD simulation of intraventricular haemodynamics based on 3D ultrasound imaging.” *Biomed Eng Online*, vol. 15, p. 107, 2016. doi:10.1186/s12938-016-0231-9

[161] J. H. Seo, V. Vedula, T. Abraham, et al., “Effect of the mitral valve on diastolic flow patterns,” *Physics of Fluids*, vol. 26, p. 121901, 2014. doi:10.1063/1.4904094

[162] N. Bessonov, A. Sequeira, S. Simakov, Y. Vassievskii, and V. Volpert, “Methods of Blood Flow Modelling,” *Math. Model. Nat. Phenom*, vol. 11, pp. 1–25, 2016. doi:10.1051/mmnp/201611101

[163] S. N. Doost, L. Zhong, B. Su, and Y. S. Morsi, “The numerical analysis of non-Newtonian blood flow in human patient-specific left ventricle.” *Comput Methods Programs Biomed*, vol. 127, pp. 232–47, 2016. doi:10.1016/j.cmpb.2015.12.020

List of publications and supervised theses

Journal Articles

- **J. Brenneisen**, A. Daub, T. Gerach, E. Kovacheva, L. Huetter, B. Frohnapfel, O. Dössel, A. Loewe *Sequential Coupling Shows Minor Effects of Fluid Dynamics on Myocardial Deformation in a Realistic Whole-Heart Model*, Frontiers in Cardiovascular Medicine, vol. 8, 2021

Refereed Conference Articles

- **J. Brenneisen**, D. Müller, A. Stroh, B. Frohnapfel, O. Dössel, A. Loewe *Cardiac fluid dynamics based on immersed boundary method for application in hypertrophic cardiomyopathy*, Proceedings of the 7th International Conference on Computational and Mathematical Biomedical Engineering - CMBE2021, ISBN 978-0-9562914-6-2, 2022
- **J. Brenneisen**, C. Wentzel, F. Karwan, O. Dössel, A. Loewe *Fluid dynamics in the human heart: Altered vortex formation and wash-out in mitral regurgitation simulations*, Current Directions in Biomedical Engineering 2021;7(2):199-202
- **J. Brenneisen**, S. Schuler, E. Kovacheva, T. Gerach, O. Dössel, A. Loewe *Influence of Geometrical Properties for the Calculation of a Pressure-Free Whole Heart Geometry*, Scipedia Volume 400 - Biomechanics and Mechanobiology, WCCM-ECCOMAS2020, 2021
- M. Kupper, **J. Brenneisen**, O. Stark, S. Krebs, S. Hohmann *Cascaded Fractional Kalman Filtering for State and Current Estimation of Large-Scale Lithium-Ion Battery Packs*, 2018 Chinese Control And Decision Conference (CCDC):5071-5078, 2018

Refereed Conference Abstracts

- **J. Brenneisen**, O. Dössel, A. Loewe *Influence of pressure boundary condition definition on flow patterns in cardiac simulations*, iHEART Congress – Modelling the Cardiac Function, 2022
- **J. Brenneisen**, A. Daub, E. Kovacheva, T. Gerach, L. Huetter, B. Frohnapfel, O. Dössel, A. Loewe *A sequential coupling approach for fluid-structure interaction in a patient specific whole heart geometry*, iHEART Congress – Modelling the Cardiac Function, 2021
- **J. Brenneisen**, E. Kovacheva, T. Gerach, A. Daub, L. Huetter, B. Frohnapfel, O. Dössel, A. Loewe *Characterization of the Fluid Dynamic Pressure Field in the Human Heart as a Basis for Coupled Fluid-Structure Simulations*, 54th Annual Conference of the German Society for Biomedical Engineering, 2020

DFG Grant application

- Contribution to project *Effiziente und robuste Kopplungsmethoden für elektromechanische Modelle des menschlichen Herzens* in the DFG priority programme *Robuste Kopplung kontinuumsbiomechanischer in silico Modelle für aktive biologische Systeme als Vorstufe klinischer Applikationen - Co-Design von Modellierung, Numerik und Nutzbarkeit (SPP 2311).*
  PI: PD Dr.-Ing. Axel Loewe, Prof. Dr. Christian Wieners
  Accepted in July 2021.

Supervised Student Research Projects

- Mathias Franz, *Simulation and Analysis of Fluid-Structure-Interaction in the Presence of Plaque in the Coronary Arteries*, Bachelor Thesis, Institute of Fluid Mechanics and Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2022
- Johannes Betz, *Implementierung und Analyse eines Computermodells der Mitralklappe basierend auf der Immersed Boundary Methode*, Bachelor Thesis, Institute of Fluid Mechanics and Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2021
- Michael Meinzer, *Implementation of the "Immersed Boundary Method"*, Internship, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2021

- Miriam Weiß, *Bewertung und Anpassung der Mitralklappenimplementierung in einem parametrischen Kreislaufmodell basierend auf fluiddynamischen Simulationen*, Master Thesis, Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2021
- Farokh Karwan, *Simulation und Analyse der Blutflusskomponenten im linken Ventrikel des menschlichen Herzens mittels numerischer Strömungsmechanik*, Bachelor Thesis, Institute of Fluid Mechanics and Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2021
- Carlo Wentzel, *Analysis of stenosis and regurgitation as effects of valve defects on the fluid dynamics in the left heart*, Bachelor Thesis, Institute of Fluid Mechanics and Institute of Biomedical Engineering, Karlsruhe Institute of Technology (KIT), 2020