Numeriek modelleren van de vloeistof-structuurinteractie in complexe vasculaire geometrieën

Numerical Modelling of the Fluid-Structure Interaction in Complex Vascular Geometries

Joris Bols

Promotoren: prof. dr. ir. J. Vierendeels, prof. dr. ir. P. Segers
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen: Werktuigkunde-Elektrotechniek

Vakgroep Mechanica van Stroming, Warmte en Verbranding
Voorzitter: prof. dr. ir. J. Vierendeels
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2014 - 2015

UNIVERSITEIT
GENT

ISBN 978-90-8578-729-7
NUR 978, 954
Wettelijk depot: D/2014/10.500/75

Supervisors:

Prof. dr. ir. Jan Vierendeels
Prof. dr. ir. Patrick Segers

Research lab:

Department of Flow, Heat and Combustion Mechanics (FloHeaCom)
Ghent University
Sint-Pietersneiuwstraat 41
B–9000 Gent
Belgium

Members of the exam committee:

Chairman:	
Prof. dr. ir. Patrick De Baets	Faculty of Engineering and Architecture, UGent
Secretary:	
Prof. dr. ir. Joris Degroote	Faculty of Engineering and Architecture, UGent
Reading committee:	
Dr. ir. Nele Famaey	Faculty of engineering, KU Leuven KU Leuven
Prof. dr. Hennie De Schepper	Faculty of Engineering and Architecture, UGent
Prof. dr. ir. Frans van de Vosse	Eindhoven University of Technology, The Netherlands
Other members:	
Dr. ir. Gianluca De Santis	FEops
Prof. dr. ir. Benedict Verhegghe	Faculty of Engineering and Architecture, UGent
Prof. dr. ir. Patrick Segers	Faculty of Engineering and Architecture, UGent
Prof. dr. ir. Jan Vierendeels	Faculty of Engineering and Architecture, UGent

This research was funded by the Special Research Fund of the Ghent University (BOF10/GOA/005) and by the Research Foundation - Flanders (FWO) project nr. G.0275.08 .

Acknowledgements

Driven by scientific curiosity I accepted the challenge of a four-year PhD journey. Where I initially thought that it would be my own responsibility to get the most out of this opportunity, I soon started to realize that I would owe the final result to those who (voluntarily) invested their time, experience, patience, and even love to make this period of my life unforgettable and fruitful. I wish to acknowledge each of them for their contribution to this dissertation, and to my personal and professional development in general.

First of all, I would like to thank my supervisors prof. Jan Vierendeels and prof. Patrick Segers for offering me this opportunity, for sharing their knowledge with a remarkable sense of devotion and enthusiasm and for giving me the freedom to define my own research path. Furthermore, I would like to thank prof. Joris Degroote for being my search engine when Google let me down and for counselling me at every stage of this PhD research: from theory over implementation to proofreading. I am also greatly indebted to Gianluca De Santis for introducing me to the art of meshing and for being my pyFormex mentor by doing so. I am grateful to Benedict Verhegghe for letting me think in a pythonic way and improving my Linux skills. I would also like to thank the (other) members of my examination committee for spending time on evaluating my dissertation, and for providing valuable comments and suggestions.

Many thanks go to Bram Trachet for introducing me to all research aspects related to hemodynamics in mice, it was a pleasure to collaborate with you. Furthermore, my gratitude goes to Jeroen De Ridder for challenging me with constructive criticism on a daily base and for turning a $30\,m^2$ office into a productive working environment. Although his trademark was p(l)aying a Duvel, Yves Maenhout deserves my gratitude for being the 24/7 IT specialist where most PhD students can only dream about. Additionally, I would like to dedicate a special thank you to Saskia, Griet, Annie and Rita for all the administrative and practical arrangements. Many thanks to my colleagues and friends at both the Technicum and the UZGent, I greatly enjoyed all ‘events’ and the many interesting office and lunch discussions!

Thanks and love to my father Rik, my mother Hilde and my brother Dries. They have always encouraged me and they have always given me what I needed, whether it was their love and affection or material support. Paul and Christine, thank you for making me feel home at the Vrijheidslaan. Last, but certainly not least, I must acknowledge that I could not have completed this PhD journey without Liesbeth by my side. There are no words that can express my gratitude and appreciation for all you’ve done and been for me, just so I could focus on completing my dissertation. Besides your love, patience and support, you have given me so many happy and beautiful memories throughout this journey, I love you!

Joris Bols
Gent, August 2014

Samenvatting
Summary

Samenvatting

Introductie

Een complex netwerk van bloedvaten in het cardiovasculair systeem verzorgt de distributie van het bloed naar het ganse lichaam en het transport terug naar het hart. Het is reeds eeuwen bekend dat vloeistofmechanica en structuurmechanica een belangrijke rol spelen in dit transportfenomeen. Om het vaatstelsel te bestuderen, kunnen onderzoekers zich baseren op experimenten uitgevoerd op levende organismen (*in vivo*), op technieken die buiten het lichaam van het organisme toepasbaar zijn (*in vitro*, op experimentele labo-opstellingen of *ex vivo*) of op computer simulaties (*in silico*). Met deze laatstgenoemde techniek kan informatie bekomen worden die niet (eenvoudig) op te meten is in een experimentele of klinische omgeving. Gezien deze modellen eveneens toelaten het geïsoleerd effect van een afzonderlijke parameter te bestuderen, kunnen ze bijdragen tot een beter begrip en inzicht in de fysiologische processen en mechanismen die aanleiding geven tot cardiovasculaire aandoeningen, nog steeds de belangrijkste doodsoorzaak in de Westerse wereld. De ontwikkeling van numerieke methodes biedt bovendien een computationele omgeving aan waarbinnen nieuwe en reeds bestaande theorieën, medische procedures en toestellen kunnen gevalideerd, getest en geoptimaliseerd worden. Deze technieken zijn dus zowel kostenefficiënt als patiëntvriendelijk, en laten daarenboven toe om dierexperimenten te vermijden of in aantal te verminderen. Met de voortdurende vooruitgang van computationele methoden, rekenkracht en medische beeldvormingstechnieken, vormt computationele biomechanica een veelbelovend hulpmiddel in de (vasculaire) gezondheidszorg, dat mogelijks in de klinische praktijk gebruikt zal worden.

De laatste decennia is een duidelijke trend waarneembaar in de richting van het gebruik van meer realistische modellen van het vaatstelsel, op basis van *in vivo* metingen. Deze modellen zijn gebaseerd op patiëntspecifieke geometrieën en/of brengen de interactie tussen de bloedstroming en de vaatwand in rekening. Dit brengt helaas met zich mee dat numerieke modellen steeds moeilijker te ontwikkelen en op te lossen zijn. Enerzijds verhoogt de complexiteit van de geometrieën de gebruikerstijd met betrekking tot de opmaak van het model, anderzijds zal een uitbreiding van het rekendomein en de inclusie van de fluïdum-structuurinteractie (FSI) de rekentijd doen toenemen. Dit is in het bijzonder het geval wanneer een vasculaire pathologie zoals een lokale vernauwing (stenose) of een lokale dilatatie (aneurysma) van de arterie beschouwd wordt.

In een poging de hoge gebruikerstijd en rekentijd die gerelateerd zijn aan de complexe geometrieën te beperken, wordt in Deel I van dit werk een nieuwe aanpak geïntroduceerd die toelaat de rekenroosters van het stromingsdomein van complexe vasculaire structuren volledig op te bouwen uit hexaëders. De aldus gegenereerde roosters zijn bijzonder computationeel efficiënt doordat de roostercellen gealigneerd zijn met de voornaamste stromingsrichting en doordat lokale (anisotrope) roosterverfijningen mogelijk zijn. Verder laat de methode een hoge cel kwaliteit toe in regio's met een hoge kromming. Dit komt de robuustheid ten goede, waardoor een volledig geautomatiseerde aanpak mogelijk wordt die de waardevolle gebruikerstijd kan minimaliseren. Deel II van dit proefstuk behandelt de spanningen en rekken die optreden in de arteriële wand ten gevolge van de interne belasting aanwezig op het moment van beeld acquisitie. Verwaarlozing van deze spanningen zou immers leiden tot een foutieve begroting van de spanningen in de wand en van de vervormingen die optreden bij de simulatie van de fysiologische druk in een beeldgebaseerde vasculaire structuur. Om deze onbekende spanningen in rekening te brengen, wordt een methodiek ontwikkeld die eenvoudig implementeerbaar is en toelaat de onbelaste toestand van de beeldgebaseerde geometrieën te achterhalen. De implementatie is bovendien toepasbaar op softwarepakketten waarbij er geen toegang is tot de broncode. De technieken ontwikkeld in de eerste twee delen worden ten slotte toegepast in Deel III van dit proefschrift dat focust op de ontwikkeling van een fluïdum-structuurinteractie model van een abdominale muizenaorta gebaseerd op *in vivo* metingen.

Deel I. Generatie van ongestructureerde hexaërder roosters van hoge kwaliteit voor complexe vasculaire geometrieën

Hoofdstuk 1. Introductie tot hexaëder roostergeneratie in het vaatstelsel

De trend in de richting van meer realistische numerieke modellen van (pathologische) patiëntspecifieke vasculaire structuren brengt een uitbreiding van het computationeel domein en een verhoogde complexiteit van de geometrieën met zich mee, wat een nadelig gevolg heeft voor zowel de reken- als gebruikerstijd. Uit de literatuur blijkt dat hexaëder roosters efficiënt de rekentijd en de nodige rekenkracht reduceren. Deze winst in rekentijd gaat echter ten koste van de gebruikerstijd (voor het creëren van het rekenrooster), in het bijzonder wanneer de computationele analyses betrekking hebben op complexe geometrieën zoals de nek van een aneurysma, stenoses of bifurcaties. Bovendien laten dergelijke roosters over het algemeen geen lokale verfijning toe. In een poging om deze beperkingen te reduceren wordt een nieuwe aanpak voor de generatie van hexaëder roosters voorgesteld waarbij een automatische creatie van een multi-blok structuur gecombineerd wordt met een rooster-gebaseerde methode.

Hoofdstuk 2. De uitgebreide Treemesh methode

In dit hoofdstuk wordt een strategie voorgesteld om geautomatiseerd een multiblok structuur van hexaëders te genereren voor een vasculair netwerk met verschillende vertakkingen. Dit wordt mogelijk gemaakt door de topologie van deze vertakkingen te benaderen adhv. moedertakken, gevolgd door de constructie en alignatie van vaatomsluitende vierkanten en hun inter-connectie tot een reeks hexaëder blokken. Een verfijnde blok-structuur en een radiaal gecomprimeerd oppervlakte rooster van de vasculaire boom worden vervolgens als uitgangspunt voor de welbekende roostergebaseerde methode genomen. Deze aanpak resulteert in een reeks van centerlijn gealigneerde hexaëder volumes voor de kern van het fluïdum domein. Deze ongestructureerde kern van hexaëders sluit nauw aan bij de originele geometrie en laat de creatie toe van een grenslaagrooster van hoge kwaliteit tussen het buitenoppervlak van de hexaëder kern en het originele oppervlakte rooster.

Hoofdstuk 3. Validatie van de uitgebreide Treemesh methode gebaseerd op de resultaten van een CFD variabiliteitsstudie

In de literatuur worden numerieke studies beschreven waarin beeldgebaseerde computationele fluïdum dynamica (CFD) resultaten gevalideerd worden ten opzichte van *in vitro* metingen en *in vivo* opgemeten data. Slechts een paar van deze studies rapporteerden de sensitiviteit van de CFD resultaten aan het gebruikte softwarepakket en de oplossingsmethode. Door deel te nemen aan een CFD variabiliteitsstudie konden we onze oplossingen toetsen aan de resultaten bekomen door andere experten binnen het vakgebied. Dit liet toe onze CFD oplossingsmethode in zijn geheel en de roostergeneratie in het bijzonder te valideren, zonder voorafgaande kennis te hebben van de uiteindelijke resultaten wat de modelleringsaanpak zou kunnen beïnvloeden. Dit hoofdstuk bespreekt aldus onze bijdrage tot een studie waarin de variabiliteit van de berekende druk en stroming in een beeldgebaseerde geometrie van een aanzienlijk carotis-aneurysma in combinatie met een stenose proximaal van dit aneurysma bestudeerd wordt.

Hoofdstuk 4. Toepassing van de uitgebreide Treemesh methode op een FSI model van aorta-coarctatie

In voorgaande hoofdstukken werd de robuustheid van de ontwikkelde roostergeneratie methode getest op complexe geometrieën zoals boomstructuren met een driesprong, een aneurysma of een milde stenose. Om dit plaatje te vervolledigen wordt er in dit hoofdstuk een rooster sensitiviteitsanalyse in een zeer ernstige stenose beschouwd. Het beschouwde fluïdum-structuurinteractie model is representatief voor een aorta met aangeboren aorta coarctatie. Dit voorbeeld illustreert de mogelijkheden van de uitgebreide Treemesh methode. Door een verstandig gebruik van anisotrope roosterverfijningen kan de rekentijd aanzienlijk gereduceerd worden met behoud van nauwkeurigheid.

Deel II. Inverse modellering van beeld-gebaseerde patiëntspecifieke bloedvaten

Hoofdstuk 5. Introductie tot het voorspanningsprobleem in arteriën en limitaties van bestaande methodologiën

In het vakgebied van de computationele cardiovasculaire biomechanica is de numerieke simulatie van de vaatwandvervormingen en -spanningen een vaak bestudeerd probleem. Gezien de wandgeometrie geëxtraheerd wordt uit medische beelden, is de aanwezige spanningsverdeling in de vaatwand onbekend. De simulatie van de werkelijke fysiologische druk binnenin de vaatwand zonder het in rekening brengen van de *in vivo* spanningen zal echter resulteren in een onnauwkeurige begroting van de spanningen en de wandvervormingen.

Hoofdstuk 6. De achterwaartse verplaatsingsmethode

Om de computationele modellen te verbeteren en de aangehaalde onnauwkeurigheden te vermijden stelt dit hoofdstuk de BDM methode voor (afkorting voor backward displacement method). Vertrekkende van de *in vivo* opgemeten geometrie van de belaste structuur bepaalt deze methode de geometrie in afwezigheid van een interne druk, de zogenaamde nul-druk geometrie, en het *in vivo* spanningstensorveld. Na de introductie van de achterwaartse verplaatsingsfunctie $\mathcal{B}$ en de functie $\mathcal{S}$, die de structuursolver voorstelt, kan het invers probleem beschreven worden aan de hand van een vaste punt probleem. De benadering van de nul-druk geometrie wordt hierbij geüpdatet met behulp van vaste punt iteraties waarin de i^{de} achterwaartse verplaatsingsfunctie toegepast wordt op de output van de i^{de} voorwaartse structuuranalyse. Om de BDM methode te versnellen kan het probleem ook opgelost worden met een quasi-Newton algoritme met een benadering van de inverse Jacobiaan aan de hand van een kleinste kwadraten (least squares) model (QN-ILS). Deze techniek brengt de informatie bekomen in de voorgaande iteraties in rekening om de convergentiesnelheid te verbeteren. In beide oplossingsstrategieën kan het gemiddelde residu bepaald worden aan de hand van de gemiddelde afstand aanwezig tussen de beeldgebaseerde geometrie en de geometrie bekomen als oplossing van het voorwaartse probleem. Wanneer convergentie bereikt is, wordt (i) de nul-druk geometrie bekomen en (ii) de resulterende *in vivo* opgemeten geometrie teruggevonden die in evenwicht is met het *in vivo* spanningsveld en de *in vivo* belasting. Bovendien hoeven bij deze aanpak enkel de coördinaten van de knooppunten geüpdatet te worden, wat een eenvoudige implementatie toelaat wanneer er met bestaande softwarepakketten voor structuurberekeningen gewerkt wordt, zelfs indien er geen toegang is tot de broncode (wat bij de meeste commerciële pakketten het geval is).

Hoofdstuk 7. Numerieke validatie en applicatie van de achterwaartse verplaatsingsmethode

Twee voorbeelden worden gebruikt om de BDM methode te illustreren. Eerst wordt een vereenvoudigd model van een muisaorta (voorgesteld door een rechte cilindrische buis) beschouwd voor validatie doeleinden en om het belang van voorspanning in de beeldgebaseerde geometrie te evalueren. Het tweede voorbeeld betreft een muisspecifieke abdominale aorta met vier aftakkingen om na te gaan of de BDM methode ook inzetbaar is om de nul-druk geometrie te achterhalen bij meer complexe geometrieën. De keuze voor toepassing op de aorta van een muis kadert in lopend biomechanisch onderzoek naar de etiologie van aneurysmavorming in muismodellen, en *in vivo* data was onmiddellijk beschikbaar. Beide voorbeelden omvatten de vergelijking van de convergentie bekomen met de vaste punt methode enerzijds en de quasi-Newton methode anderzijds. Om het materiaalgedrag van de vaatwand te modelleren wordt de BDM methode ten slotte aangewend in een optimalisatie procedure voor materiaal parameter identificatie op basis van *in vivo* opgemeten data.

Deel III. Van medisch beeld tot computationeel FSI model

Hoofdstuk 8. Fluïdum-structuurinteractie model van een abdominale muizenaorta

In dit hoofdstuk wordt een fluïdum-structuurinteractie simulatie van een abdominale muizenaorta ontwikkeld, met opnieuw de motivatie van lopend biomechanisch onderzoek in muismodellen en de beschikbaarheid van data. Het oplossen van de hemodynamica en de arteriële wandmechanica op zich zijn reeds complexe problemen, maar de koppeling van beiden is nog uitdagender. De modelleringsaspecten beschouwd in Deel I en II van dit werk worden opgenomen om hun toepasbaarheid in fluïdum-structuurinteractie simulaties van complexe vasculaire structuren aan te tonen. Eerst wordt de *in vivo* data acquisitie en de toepasbaarheid van de uitgebreide Treemesh methode besproken. Hieruit resulteert een rekenrooster voor zowel het stromingsdomein als het structuurdomein. Vervolgens wordt een CFD model geconstrueerd en de gevoeligheid van de stromingsresultaten aan de roosterdensiteit geanalyseerd. Ook de vergelijking wordt gemaakt met de resultaten bekomen voor een tetraëder rooster. Daarna wordt een computationeel structuurmechanica model van de arteriële wand opgesteld en op spanning gebracht aan de hand van de BDM methode, en dit voor materiaal parameters die zodanig bepaald werden dat de *in vivo* opgemeten diametervariatie volgt uit een fysiologische drukvariatie. Ten slotte worden de stromingssimulaties en de structuursimulatie (met voorspanning) gekoppeld door het FSI probleem op te lossen aan de hand van een gepartitioneerde methode. Behalve drukvallen, snelheidsvelden en wandschuifspanningen, voorziet de FSI simulatie ook resultaten betreffende de statische druk, de vervorming van de arteriële wand en de spanningen in de arteriële wand.

Conclusies

Het finale hoofdstuk van dit proefschrift omvat een overzicht van de belangrijkste verwezenlijkingen binnen dit onderzoek met betrekking tot de initiële doelstellingen. Gebaseerd op de belangrijkste beperkingen van dit werk, worden er ten slotte aanbevelingen gemaakt voor toekomstige uitbreidingen en verder onderzoek.

Summary

Introduction

In the cardiovascular system a complex network of vessels is responsible for the transportation of blood throughout the body and back to the heart. It has been known for centuries that fluid mechanics and solid mechanics play a fundamental role in this transport system. To study the vasculature, researchers rely on data that can be obtained by careful experiments on living organisms (*in vivo*), outside of living organisms (*in vitro*, in mock loop systems or *ex vivo*), or by computer simulations (*in silico*). The latter are usually designed to provide researchers with information that cannot (easily) be measured in an experimental or clinical setting. Combined with the possibility to study the isolated effect of a single parameter, computer models may contribute to a better comprehension and insight into the physiological processes and mechanisms leading to cardiovascular diseases, which are currently the leading cause of death in the western world. Moreover, the development of numerical models offers a computational environment in which new – as well as existing – theories, medical procedures and devices can be validated, tested and optimized, which is cost-effective, patient-friendly as well as animal friendly as experiments on animals might be avoided. Given the permanent progress of computational methods, computing power and medical imaging techniques, computational (vascular) biomechanics are expected to advance (vascular) healthcare, strengthening the belief that computational biomechanical models will eventually be used in clinical practice.

In the last decade, a clear trend can be observed towards the development of realistic models of parts of the vasculature, based on *in vivo* measurements. This is done by studying patient-specific geometries and/or including the coupling between the behaviour of the blood flow and the vessel wall in the models. This, however, also implies that the numerical models become more challenging to develop and to solve. From a preprocessing point of view (extracting data from images, image segmentation and mesh creation), complex geometries tend to increase the operator time while from a computational point of view, larger computational domains and fluid-structure interaction (FSI) increase the computation time. This is especially true when the domain of interest involves vascular pathologies such as a local narrowing (stenosis) or a local dilatation (aneurysm) of the arterial wall.

In an attempt to overcome these issues of high operator times and high computation times when addressing the bio(fluid)mechanics of complex geometries, Part I of this work introduces a novel approach to generate full-hexahedral fluid meshes of complex vascular structures. The generated grids feature computational efficiency by aligning the grid lines with the predominant direction of the flow and allowing local (anisotropic) refinements. Moreover, high-quality cells are generated even in regions of high curvature, which improves the robustness of the method and allows for automation of the procedure to save valuable operator time. Part II of this dissertation deals with the *in vivo* stress and strain fields present in the arterial wall (due to internal pressure load) at the moment of medical image acquisition. Neglecting their presence results in incorrect values for the stress in the arterial wall and the deformation of the arterial wall when simulating the physiological pressure load inside image-based vascular structures. To account for this unknown stress field when working with a black box structural solver, a methodology which allows for a straightforward implementation has been formulated to compute the unloaded state of image-based geometries. Finally, the methodologies of the first two parts are applied in Part III which focuses on the development of a fluid-structure interaction model of an abdominal mouse aorta, based on *in vivo* measurements.

Part I. High-quality unstructured hexahedral mesh generation for complex vascular trees

Chapter 1. Introduction to hexahedral mesh generation in the vasculature

The trend towards the development of realistic numerical models of (pathologic) patient-specific vascular structures brings along larger computational domains and more complex geometries, increasing both the computational time and the operator time. Literature and previous experience teach that hexahedral grids effectively lower the computational run time and the required computational infrastructure, but at a high cost in terms of operator time (to generate the mesh), especially when the computational analyses are targeting complex geometries such as aneurysm necks, severe stenoses and bifurcations. Moreover, such grids generally do not allow local refinements. As an attempt to overcome these limitations, a novel approach to hexahedral meshing is proposed in which the automated generation of multi-block structures is combined with a grid-based method.

Chapter 2. The extended Treemesh method

In this chapter, a strategy is proposed to generate a hexahedral multi-block structure of a treelike network by expressing the branching topology in terms of 'parent vessels', followed by the construction and alignment of vessel-enclosing squares and their interconnection into a set of hexahedral blocks. A refined multi-block structure and a radially compressed surface mesh of the vascular tree are then provided as an input to the well-known grid-based method. The result is an unstructured set of (centerline-aligned) hexahedral volume elements for the core of the fluid domain. This unstructured hex-core resembles the volume of the original geometry well and allows for the creation of a high-quality boundary layer grid between the interface of the hex-core and the original surface mesh.

Chapter 3. Validation of the extended TreeMesh method based on the results of a CFD variability study

Numerous studies have been performed in which image-based computational fluid dynamics (CFD) predictions are validated against careful in vitro measurements and *in vivo* measured data, but only few studies in the biomedical field report the sensitivity of CFD predictions to different solvers and solution strategies. By participating in a blinded CFD variability study we were able to compare our solutions to those obtained by other experts in this field. This allowed us to validate our CFD solution strategy as a whole and our meshing procedure in particular, without having prior knowledge of the final outcome of the study influencing our modelling approach. As such, this chapter discusses our contribution to a study which sought to assess variability in the prediction of the pressure and the flow in an image-based geometry of a giant carotid aneurysm with a proximal stenosis.

Chapter 4. Application of the extended Treemesh method to an FSI-model of aortic coarctation

In previous chapters, the robustness of the meshing approach was tested on common complex geometries, such as treelike structures including a trifurcation, an aneurysm and a mild stenosis. To complete the picture with a severe stenosis, chapter 4 presents a grid sensitivity analysis of a fluid-structure interaction simulation in a model of aortic coarctation. In this application, the true potential of the extended Treemesh method is shown by obtaining anisotropically refined grids which lower the computation time while maintaining the same level of accuracy.

Part II. Inverse modelling of image-based patient-specific blood vessels

Chapter 5. Introduction to the prestress problem in arteries and limitations of existing methodologies

In the field of computational cardiovascular biomechanics, one often comes across the numerical simulation of the mechanics of the blood vessel wall. When the vessel geometry is obtained *in vivo* via medical imaging, the stress distribution throughout the vessel wall is unknown. However, simulating the full physiological pressure load inside the blood vessel without incorporating *in vivo* stresses will result in an inaccurate stress field and an incorrect deformation of the vessel wall.

Chapter 6. The backward displacement method

In order to correct computational models for the aforementioned inaccuracies, this chapter proposes the backward displacement method (BDM) to solve for the zero-pressure geometry and the *in vivo* stress tensor field starting from the *in vivo* measured geometry of the loaded structure. After introducing the backward displacement function $\mathcal{B}$ and the function $\mathcal{S}$, which represents the structural solver, the inverse problem can be described by a fixed point problem. The approximation of the zero-pressure geometry gets updated by means of fixed point iterations, in which the i^{th} backward displacement function is applied onto the output of the i^{th} forward structural analysis. In order to accelerate the backward displacement method, the corresponding root finding problem can be solved using a quasi-Newton algorithm with an approximation for the inverse of the Jacobian from a least squares model (QN-ILS). This technique incorporates information obtained during previous iterations to increase the convergence rate. In both strategies, the mean residual can get evaluated as the mean distance that is still present between the image-based geometry and the geometry resulting from the forward problem. When convergence is reached (i) a zero-pressure geometry is found and (ii) the resulting *in vivo* measured geometry is fully recovered and in equilibrium with an *in vivo* stress field and the *in vivo* load. Furthermore, only the nodal coordinates of the mesh need to be updated, allowing for a straightforward implementation in combination with existing structural solvers, even if the solver is a black box and no access is granted to the source code (as is the case with most commercial packages).

Chapter 7. Numerical validation and application of the backward displacement method

Two examples are used to demonstrate the backward displacement method. First, a simplified model of a mouse aorta (represented by a straight cylindrical tube) is used for validation purpose and to evaluate the importance of *in vivo* stress incorporation in the *in vivo* measured geometry. The second example concerns a mouse-specific abdominal aorta with four side branches to explore the ability of the backward displacement method to restore a more complex cardiovascular structure at its zero-pressure state. Both examples cover the comparison of the convergence resulting from the fixed point method and the quasi-Newton method. Finally, based on the backward displacement method, a material parameter optimization framework was set up to obtain material parameters which mimic the distension of the arterial wall using *in vivo* obtained data only.

Part III. From medical image to computational FSI model

Chapter 8. Numerical modelling of an abdominal mouse aorta

In this chapter, the modelling aspects covered in Part I and Part II of this work are applied to model the fluid-structure interaction of an abdominal mouse aorta. The choice for an abdominal mouse aorta was based on the fact that these numerical tools will, among others, be used for the assessment of aortic biomechanics in mouse models developing aneurysms and on the in-house availability of mouse-specific datasets. Solving the hemodynamics and the arterial wall mechanics separately is already complex, but the coupled nature of the problem makes it even more challenging. First, the *in vivo* data acquisition is discussed along with the application of the extended Treemesh method which results in a computational grid. Then, a CFD model is constructed for which the sensitivity of the flow results on the grid density is analyzed and compared to the results obtained with tetrahedral grids. Next, a computational structural mechanics (CSM) model for the arterial wall is set up and prestressed using the backward displacement method. This is done for material parameters which were identified such that the *in vivo* measured distension is obtained in the physiological pressure range. Finally, the flow simulation and the structural simulation (including prestress) are coupled by solving the FSI problem using a partitioned approach. Besides pressure drops, velocity fields, and wall shear stresses, the FSI simulation also provides data on the static pressure, the deformation of the arterial wall and the stress distribution in the arterial wall.

Conclusions

The last chapter of this dissertation includes an overview of the major findings with respect to the initial research goal. Based on the most important limitations of this work, final recommendations are made towards future research.

Contents

Part I – High-quality unstructured hexahedral mesh generation for complex vascular trees

PART II – INVERSE MODELLING OF IMAGE-BASED PATIENT-SPECIFIC BLOOD VESSELS

Part III – From medical image to computational FSI model

Abbreviations and Symbols

Abbreviations

3 D	Three-dimensional
AA	(distal) Abdominal aorta
AAA	Abdominal aortic aneurysm
ALE	Arbitrary Lagrangian-Eulerian
ApoE	Apolipoprotein E
ASCII	American Standard Code for Information Interchange
AV	Cycle averaged
B-mode	Brightness mode
BC	Boundary condition
BDM	Backward displacement method
BDMPO	BDM-based material parameter optimization
BL	Boundary layer
C3D20H	Quadratic hexahedral, hybrid elements
C3D8H	Linear hexahedral, hybrid elements
CA	Celiac artery
CFD	Computational fluid dynamics
CPU	Central processing unit
CS	Control surface
CSM	Computational structural mechanics
CT	Computer tomography
CV	Control volume
DICOM	Digital imaging and communication in medicine
ECG	Electrocardiogram
FEA	Finite element analysis
FIFO	First in - first out
FP	Fixed point algorithm
FSI	Fluid-structure interaction
GUI	Graphical user interface
ILS	Approximation for the inverse of the Jacobian from a least squares model
IQN	Interface quasi-Newton algorithm
IQR	Interquartile range

LRA	Left renal artery
M-mode	Motion mode
MA	Mesenteric artery
MBGB	Multi-block grid-based
MBS	Multi-block structure
MRI	Magnetic resonance imaging
PK	Peak-systolic
QN	Quasi-Newton algorithm
RAM	Random access memory
RRA	Right renal artery
SIMPLE	Semi-implicit method for pressure linked equations
SS	Steady state
STL	Stereolithography
TA	Thoracic aorta
TAWSS	Time-averaged wall shear stress
TM	Treemesh method
US	Ultrasound
VMTK	Vascular modelling toolkit
WK	Windkessel model
WSS	Wall shear stress
XTM	Extended Treemesh method

Symbols

ϵ	Convergence criterion	
λ	Stretch	
$\mathbb{N}$	Natural numbers	
$\mathbb{R}$	Real numbers	
$\mathcal{B}$	Backward displacement function	
$\mathcal{F}$	Flow solver	
$\mathcal{J}_X$	Jacobian matrix of the residual operator $\mathcal{R}_X$	
$\mathcal{R}$	Residual operator	
$\mathcal{S}$	Structural Solver	
μ	Dynamic viscosity	$[Pa \cdot s]$
μ	Shear modulus	$[Pa]$
Ω	Configuration	
ω_p	Under relaxation factor for the pressure	
ϕ	Parameter	
π	Pi	
ψ	Material parameters	
ρ	Density	$[kg/m^3]$
σ	Cauchy stress tensor	$[Pa]$
τ	Viscous stress tensor	$[Pa]$
$\boldsymbol{F}$	Deformation gradient tensor	

$\boldsymbol{I}$	Unit tensor	
$\boldsymbol{B}$	Curved block structure for a branch	$(\mathbb{R}^{N_B \times 8 \times 3})$
$\boldsymbol{c}$	Coordinates of a centerline node	$[m]$
$\boldsymbol{c}$	Least-squares coefficient vector	
$\boldsymbol{C}_A$	Additional connection block structure	$(\mathbb{R}^{N_A \times 8 \times 3})$
$\boldsymbol{C}_B$	Branch connection block	$(\mathbb{R}^{1 \times 8 \times 3})$
$\boldsymbol{C}_V$	Vessel connection block	$(\mathbb{R}^{1 \times 8 \times 3})$
$\boldsymbol{D}$	Data structure	
$\boldsymbol{d}$	Distance vector	$[m]$
$\boldsymbol{f}$	Body forces	$[N/m^3]$
$\boldsymbol{L}$	Buffer list for branch connection blocks	$(\mathbb{R}^{N_{C_B} \times 8 \times 3})$
$\boldsymbol{M}$	Multi-block structure	$(\mathbb{R}^{N_M \times 8 \times 3})$
$\boldsymbol{n}$	Bifurcation normal	$[m]$
$\boldsymbol{Q}$	Orthogonal matrix	
$\boldsymbol{Q}$	Set of consecutive squares	$(\mathbb{R}^{N_Q \times 4 \times 3})$
$\boldsymbol{R}$	Upper triangular matrix	
$\boldsymbol{r}$	Residual vector	$[m]$
$\boldsymbol{T}$	Traction vector	$[Pa]$
$\boldsymbol{U}$	Displacement vector	$[m]$
$\boldsymbol{V}$	Column matrix that contains the changes of input	
$\boldsymbol{v}$	Velocity vector	$[m/s]$
$\boldsymbol{W}$	Column matrix that contains the changes of output	
$\boldsymbol{X}$	Undeformed material coordinates	$[m]$
$\boldsymbol{x}$	Deformed material coordinates	$[m]$
$\boldsymbol{x}$	Interface displacement vector	$[m]$
$\boldsymbol{y}$	Interface stress vector	$[N/m^2]$
C_{ij}	Coefficients from a strain energy function	$[Pa]$
d_0	Distance between centerline nodes	$[m]$
e	Error	
f	(Objective) Function	
f	Factor	
f	Parameter	
h	Wall thickness	[m]
I_1, I_2	First and second invariant of the left Cauchy-Green deformation tensor	
k	Kilo (refers to ×1000)	
l	Length	$[m]$
N	Number of elements/cells	
p	Pressure	$[mmHg]$
Q	Flow	$[ml/s]$
R^i	Distance to the centerline (before compression)	$[m]$
r^i	Distance to the centerline (after compression)	$[m]$
t	Time	$[s]$
u, v, w	Velocity components	$[m/s]$
wss	Wall shear stress	$[Pa]$
x, y, z	Coordinates	$[m]$

A	Cross sectional area	$[m^2]$
A	Surface area	$[m^2]$
C	Capacitor	$[ml/mmHg]$
D	Diameter	$[m]$
d	Distension	$[m]$
E	Young's modulus	[Pa]
R	Resistance	$[mmHg \cdot s/ml]$
r	Radius	$[m]$
Re	Reynolds number	
T	Period	$[s]$
Z	Impedance	$[mmHg \cdot s/ml]$

Subscripts

θ	Circumferential direction
a	Axial direction
c	Closest face
$child$	Child branch
dia	End-diastole
$edge$	Refers to the sides of a square
f	Fluid
$inlet$	Refers to the inlet of the fluid domain
iso	Iso-contour
m	Measurement
max	Maximum
min	Minimum
N, E, S, W	Neighbouring cell centroids
n, e, s, w	Neighbouring cell face centroids
p	Pressure
p	Projection
R	Ratio
r	Radial direction
ref	Reference
s	Simulation
s	Solid
$start$	Initial
sys	Peak systole
t	Transversal direction
U	Displacement output
v	Velocity
wss	Wall shear stress

Superscripts

i	Index denoting the order of the parent vessel
i	Iteration counter
j	Index denoting the bifurcation point relative to parent vessel order i
j'	Index denoting the bifurcation point relative to parent vessel order $i+1$
k	Index denoting the vessel sprouting at branching point j^i
l	Index denoting the branch of the $k^{j^i}-th$ vessel
m	Index denoting the centerline node of the $l^{k^{j^i}}-th$ branch
$*$	Converged value
-1	Final index of a list

Operators

Δ	Difference
$\int$	Integral
$\nabla\cdot$	Divergence
∇	Gradient
∂	Partial derivative
$\sum$	Sum

Units

μm	Micrometer
bpm	Beats per minute
cm	Centimeter
g	Gram
GB	Gigabyte
GHz	Gigahertz
h	Hour
kg	Kilogram
l	Litre
min	Minute
ml	Millilitre
mm	Millimeter
$mmHg$	Millimeters of mercury
N	Newton
Pa	Pascal
s	Second

Introduction

Introduction

BACKGROUND

In the cardiovascular system a complex network of vessels is responsible for the transportation of blood throughout the body and back to the heart. It has been known for centuries that fluid mechanics and solid mechanics play a fundamental role in this transport system [60, 87]. To study the vasculature, researchers rely on data that can be obtained by careful experiments on living organisms (*in vivo*), outside of living organisms (*in vitro*, in mock loop systems or *ex vivo*), or by computer simulations (*in silico*). Over the years, advances in our knowledge on both hemodynamics and arterial wall mechanics significantly improved *in silico* or numerical models, encouraging researchers to use numerical methods to investigate phenomena which are difficult to describe using *in vivo* measurements or *in vitro* setups. Moreover, numerical models are able to provide researchers with information that cannot (easily/accurately) be measured in a clinical setting, such as

- the velocity of the blood flow near the arterial wall and the wall shear stress derived therefrom;
- the local pulse pressure and the pressure drop along a vessel;
- the stress distribution throughout the arterial wall and the relationship with the strain.

This valuable information, together with the possibility to study the isolated effect of a single parameter, contributes to a better comprehension and insight into the physiological processes and mechanisms leading to cardiovascular diseases, which are currently the leading cause of death in the western world. Moreover, the development of numerical models offers a computational environment in which new – as well as existing – theories, medical procedures and devices can be validated, tested and optimized, what is both cost-effective, patient-friendly as well as animal-friendly as experiments on animals might be avoided. Given the permanent progress of computational methods, computing power and medical imaging techniques, computational (vascular) biomechanics are expected to advance (vascular) healthcare, strengthening the belief that computational biomechanical models will eventually be used in clinical practice.

In the last decade, a clear trend can be observed towards the development of more realistic models of parts of the vasculature, which include *in vivo* obtained, patient-specific conditions. These models cannot do without non-invasive medical imaging techniques such as X-ray computed tomography (CT), magnetic resonance imaging (MRI) and ultrasound imaging. The latter two techniques do not only allow for *in vivo* visualization of three-dimensional (3D) patient-specific geometries, but they are also able to provide patient-specific in- and outflow boundary conditions for computational fluid dynamics (CFD) models. Whenever a pressure outlet boundary condition is applied instead, the measured outflow can be used for validation purposes. This is also true for the *in vivo* measured distension of the arterial wall, when it is not used to further complete the computational structural mechanics (CSM) model by determining the material parameters of a constitutive law. The trend towards more realistic models is not only limited by including patient-specific data, but numerical models and methods are improved too. Given the importance of the arterial wall properties on the character of the blood flow and the influence of the hemodynamics on the intramural stresses, more and more vascular models take into account the fluid-structure interaction (FSI) between the blood flow and the (compliant) arterial wall.

Aim and motivation

The main objective of this thesis is to further improve the current computational models in vascular biomechanics. As there are still many unresolved problems in this field [72], the focus goes to patient-specific geometrical modeling. Although it is obvious that numerical simulations rely heavily on the advances in medical imaging and 3D vessel reconstruction, the research goal here is to develop computational strategies which allow to generate more efficient and more accurate fluid-structure interaction models of complex, image-based vascular structures.

In general, the increase in accuracy brought by patient-specific FSI models is, however, achieved at the expense of geometrical and numerical complexity. This means that the numerical models become more challenging to develop (regarding the extraction of the geometry from medical images, the discretization of the volume of interest and the application of boundary conditions) and to solve (regarding the implementation of numerical methods, and the amount of equations to be solved). On the one hand, from a preprocessing point of view, complex geometries tend to increase the operator time whereas on the other hand, from a computational point of view, larger computational domains and fluid-structure interaction increase the computation time. This is especially true when the domain of interest involves a vascular pathology such as a stenosis (i.e. a local narrowing of the arterial wall) or an aneurysm (i.e. a local dilatation of the arterial wall).

Structure overview

Apart from this introduction and a final conclusion, the dissertation is divided into three distinct parts:

- Unstructured hexahedral mesh generation of complex vascular trees;
- Inverse modelling of image-based patient-specific blood vessels;
- From medical image to computational FSI model of an abdominal mouse aorta.

The first part introduces a novel approach (the extended Treemesh method, XTM) to generate full-hexahedral fluid meshes of complex vascular structures. The generated grids feature computational efficiency by aligning the grid lines with the predominant direction of the flow and allowing local (anisotropic) refinements. Moreover, high-quality cells are generated even in high-curvature regions, what improves the robustness of the method and allows for automation of the procedure to save valuable operator time.

The second part describes a novel methodology (the backward displacement method, BDM) to compute the unloaded state of image-based geometries to account for the *in vivo* stress and strain field present in the arterial wall (due to internal pressure load) at the moment of medical image acquisition. Neglecting their presence results in incorrect values for the stress in the arterial wall and the deformation of the arterial wall when simulating the physiological pressure load inside image-based vascular structures.

Finally, the extended Treemesh method of Part I and the backward displacement method of Part II are applied in the third part to develop an accurate and efficient fluid-structure interaction model of an abdominal mouse aorta with *in vivo* obtained data.

Each part starts with an introductory chapter, including an overview of the literature as well as a brief outline to summarize its content. In the first two parts, this introduction is followed by a description of the novel methodology, a validation procedure and an example, whereas in the third part the CFD model, the CSM model and FSI model are subsequently discussed.

One

High-quality unstructured hexahedral mesh generation for complex vascular trees:

The eXtended TreeMesh method

Chapter 1

Introduction to hexahedral mesh generation in the vasculature

1.1 Introduction

The trend towards the development of realistic numerical models of (pathologic) patient-specific vascular structures brings along larger computational domains and more complex geometries, increasing both the computation time and the operator time. As depicted in literature (see Section 1.2), hexahedral grids can effectively lower the computational run time and the required computational infrastructure, but at a high cost in terms of operator time, especially when the computational analyses are targeting complex geometries such as aneurysm necks, severe stenoses and bifurcations. Moreover, such grids generally do not allow local refinements. As an attempt to overcome these limitations, a novel approach to hexahedral meshing is proposed in Chapter 2, which combines the automated generation of a vessel enclosing multi-block structure with a grid-based method. This strategy is further referred to as the *multi-block grid-based (MBGB)* method and the corresponding software implementation is further referred to as the *extended Treemesh method (XTM)*. The robustness of this novel approach is demonstrated on common complex geometries, such as treelike structures, aneurysms and stenoses together with the procedure to introduce local refinements and the ingredients to automate the whole approach. A validation of the method is provided by means of a comparison with results of a CFD variability study (Chapter 3). The performance of the generated grids, the effect of local grid refinements and the applicability of the grids in fluid-structure interaction (FSI) models is assessed and analyzed using a numerical example of an aorta with aortic coarctation (Chapter 4). Furthermore, the in this Part developed extended Treemesh method is applied in the workflow towards a numerical model of an abdominal mouse aorta in Part III, where the performance of the generated grids is further explored.

1.2 Literature overview

Computational grids comprising of hexahedral instead of tetrahedral volume elements have been reported to require less cells to obtain a mesh independent result and shorter computation times, especially when simulating the flow in segments of the cardiovascular or the respiratory system using computational fluid dynamics (CFD) [16, 42, 82]. This has been ascribed to the fact that adequate hexahedral grids consist of well organized cells along the predominant direction of the flow, aligning the cell face normals with the physical flux [3, 40] and allowing an efficient distribution of the cells as they can be stretched or subdivided anisotropically without deteriorating the cell quality [10, 16]. However, the construction of an adequate hexahedral mesh for a vascular tree is a complex and time-consuming task for the operator [4, 17, 42, 82]. This problem is amplified when the geometry presents complex features such as multiple bifurcations, high-curvature regions, stenoses and aneurysms.

Mesh generation methods based on sweeping a quadrilateral decomposition of the cross section for creating (block) structured hexahedral grids of tubular structures [9, 26, 63] and bifurcations [39, 41, 88] have been proposed in the last decade. This led to the development of robust and automated strategies where the sweeping is based on a skeleton [3], on the solution of a thermal conduction problem [80] or on geometrical features and operator choices using a graphical user interface [15, 17]. Decomposition templates for various branching configurations have been proposed by Zhang et al. [89]. However, a configuration with a single bifurcation may already lead to skewed elements at this bifurcation region, limiting the use of these sweeping techniques to 'simple' bifurcations only. To minimize the generation of skewed elements at the bifurcation and to prevent sweeping them through the model, a triangular-based prismatic cut – located at the center of the branching point and remeshed with hexahedrons – has been suggested by Antiga et al. [3].

When geometries comprise multiple branching points or n-furcations, hexahedral meshing strategies based on mapping instead of sweeping can be used. These strategies combine a preliminary generated multi-block structure with a volumetric block refinement to create the grid points. A body-fitted grid is then generated by mapping the volume mesh to the volume of interest and by mapping the border mesh onto the surface of the image-based geometry [31]. A boundary layer grid can be included by introducing building blocks with a so-called ‘butterfly’ pattern, in which four blocks surround a fifth block at the core [42, 90]. This approach can be semi-automated by building multi-block structures inside and outside the vessel geometry as demonstrated by De Santis et al. (the *Treemesh* method) [15, 18]. However, current procedures for generating adequate multi-block structures are still not fully automated or lack robustness, and manual interventions are required to correct self-intersecting or low quality blocks [18, 54, 55] leading to intensive and extensive operator tasks. Moreover, block corrections to assure a proper mapping in high-curvature regions of complex geometries – by pursuing small variations in the local geodesic distance between the geometry and the multi-block structure – typically deteriorate the quality of the blocks and the cells of the final mesh.

The rationale of this work is to address these shortcomings by providing a robust and high-quality hexahedral meshing procedure – even in high-curvature regions – that allows automated mesh generation of complex vascular trees and is able to handle any combination of vessel segments (straight or curved), side branches (bifurcations and more general n-furcations), stenoses and aneurysms.

CHAPTER 2

The extended Treemesh method [1]

This chapter presents a novel procedure for generating a multi-block structure by auto-connecting curved segments of consecutive hexahedral blocks (Section 2.1). The multi-block structure is refined and used together with a radially compressed geometry as input for a grid-based method to generate an unstructured grid for the core of the fluid domain. This unstructured hex-core resembles the volume of the original geometry well and allows the creation of a high-quality boundary layer grid (Section 2.2). The presented meshing strategy is able to handle aneurysms by extending the multi-block structure with additional blocks (Section 2.3) and allows local grid refinements and automation (Section 2.4).

The methods proposed in this chapter were implemented using pyFormex, a python-based open-source software package for generating, transforming and manipulating large geometrical models of three-dimensional (3D) structures by sequences of mathematical operations [51]. The framework in pyFormex was combined with output from a geometrical analysis using the VMTK software, which is a collection of libraries and tools for 3D reconstruction, geometric analysis, mesh generation and surface data analysis for image-based modeling of blood vessels. [83].

[1] This chapter is based on the article *"Unstructured hexahedral mesh generation of complex vascular trees using a multi-block grid-based approach."*, as submitted (in human readable format using the `-h` option) to the Journal *Computer Methods in Biomechanics and Biomedical Engineering*, 2014.

2.1 Multi-block structure generation

A multi-block structure is defined as an interconnection of hexahedral blocks. Ideally, every block has straight edges with angles of $\frac{\Pi}{2}$ radians. Such structures are frequently used in numerous application fields and commercial grid generators as a first step towards the generation of block structured hexahedral meshes using a mapping approach. In this work, the multi-block structures will be further processed – instead of being mapped – to get versatile unstructured hexahedral meshes, able to cover any combination of vessel segments (straight or curved), side branches (bifurcations, trifurcations and more general 'n-furcations'), aneurysms and stenoses.

In this section a strategy is proposed to generate a hexahedral multi-block structure of a treelike network by expressing the branching topology in terms of 'parent vessels' (Section 2.1.1), followed by the construction and alignment of vessel-enclosing squares (Section 2.1.2) and their interconnection into a set of hexahedral blocks (Section 2.1.3). Although the here proposed strategy is able to handle fusiform aneurysms and mild stenoses, the reader is referred to Section 2.3 and Section 2.4.2 for more complex cases such as saccular type of aneurysms and severe stenoses.

2.1.1 Branching topology

Given the inlet and outlets of the geometry (of which the surface mesh is provided e.g. in the STL-format), a centerline with a constant prescribed distance d_0 between the centerline nodes is computed, together with the radii of the maximum inscribed spheres at the centerline nodes, and the distances between the surface geometry and the centerline [2, 5]. Each set of centerline nodes, corresponding to a vessel segment between two consecutive branching points or between a branching point and an inlet or outlet, is further referred to as a 'branch' with an arbitrary, but unique, branch label.

The connectivity between the branches is extracted from the centerline data and used to define the complete branching topology in terms of 'parent vessels'. Starting with the branch at the inlet, consecutive branches are gathered into one (non-bifurcating) parent vessel, the so-called zeroth order parent vessel. This is done such that for every branching point the angle between the direction vector of the incoming parent branch and the direction vector of the outgoing parent branch is minimal. The remaining child branches are listed per branching point using a FIFO-buffer, and set as start branches to retrieve the first order parent vessels. This procedure is repeated as long as the FIFO-buffer is filled with start branch labels of next order parent vessels. An example is provided in Figure 2.1(a), where the branches are colored by parent vessel order and labeled in the same order as in which they are handled by the FIFO-buffer.

For further processing, the branching topology is unambiguously stored in a four-dimensional data structure $\boldsymbol{D}$ of branch labels, Figure 2.1(b). It is important to note that the order in which the branch labels are listed denotes the connectivity of the branches. The data structure is generated by applying Algorithm 2.1. In this algorithm, the variables are defined as follows:

- b represents a branch label and b_{inlet} represents the branch label at the inlet. A branch label b is denoted by $D^{i,j,k,l}$ if it represents the l^{th} branch of the k^{th} vessel at the j^{th} branching point of order i. This is demonstrated in Figure 2.1(c).
- $\boldsymbol{D}$ represents a four-dimensional data structure of branch labels. This structure of nested lists can be expanded into four types of one-dimensional lists of objects:
  - $\boldsymbol{D}$ represents a one-dimensional list with N objects, listing the sets of parent vessel sets $\boldsymbol{D}^i$ for different orders of i ($\in [0, N-1]$).
  - $\boldsymbol{D}^i$ represents a one-dimensional list with N^i objects, listing the parent vessel sets $\boldsymbol{D}^{i,j}$ of order i for different branching points j ($\in [0, N^i - 1]$).
  - $\boldsymbol{D}^{i,j}$ represents a one-dimensional list with $N^{i,j}$ objects, listing the i^{th} order parent vessels $\boldsymbol{D}^{i,j,k}$ ($k \in [0, N^{i,j} - 1]$) at branching point j.
  - $\boldsymbol{D}^{i,j,k}$ represents a one-dimensional list with $N^{i,j,k}$ objects, listing the branch labels $D^{i,j,k,l}$ ($l \in [0, N^{i,j,k} - 1]$) of the k^{th} vessel of order i at the j^{th} branching point.
- $\boldsymbol{b}^i_{start}$ represents a two-dimensional list of start branch labels, listed per branching point j. This FIFO-buffer is updated by the following variables:
  - $\boldsymbol{b}^{i+1}_{start}$ represents a one-dimensional list of N^{i+1} objects, listing the child branch sets $\boldsymbol{b}^{i+1,j'}_{child}$ at different branching points j' of the next parent vessel order $(i+1)$.
  - $\boldsymbol{b}^{i+1,j'}_{child}$ represents the one-dimensional list of child branch labels at a branching point j' of the next order $(i+1)$.

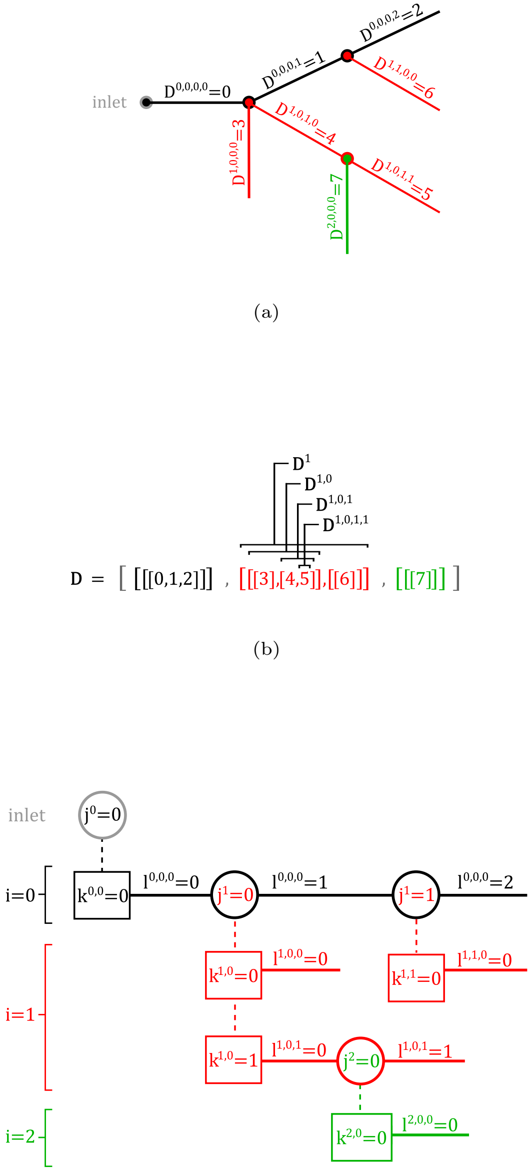


Figure 2.1: (a) Skeleton representation of the treelike structure corresponding to (b) the data structure $\boldsymbol{D}$. (c) Schematic representation of the branching topology in terms of ‘parent vessels’ to demonstrate the way of indexing.

Algorithm 2.1 Algorithm to unambiguously define the branching topology in terms of ‘parent vessels’

```
initialize empty lists $\boldsymbol{D}$
initialize list $\boldsymbol{b}^0_{start}$ to $[[b_{inlet}]]$
initialize parent vessel order $i$ to 0
while size of $\boldsymbol{b}^i_{start} > 0$ do
    initialize empty list $\boldsymbol{D}^i$
    initialize empty list $\boldsymbol{b}^{i+1}_{start}$
    for branching point $j = 0$ to size of $\boldsymbol{b}^i_{start}$ do
        initialize empty list $\boldsymbol{D}^{i,j}$
        for start branch $k = 0$ to size of $\boldsymbol{b}^i_{start}[j]$ do
            initialize list $\boldsymbol{D}^{i,j,k}$ to $[\boldsymbol{b}^i_{start}[j][k]]$
            while last branch $\boldsymbol{D}^{i,j,k}[-1]$ ends in branching point $j'$ do
                initialize empty list $\boldsymbol{b}^{i+1,j'}_{child}$
                for branches connected to branching point $j'$ do
                    if branch is next branch of parent vessel $\boldsymbol{D}^{i,j,k}$ then
                        append branch label $b$ to $\boldsymbol{D}^{i,j,k}$
                    else
                        append $b$ to $\boldsymbol{b}^{i+1,j'}_{child}$
                    end if
                end for
                append $\boldsymbol{b}^{i+1,j'}_{child}$ to $\boldsymbol{b}^{i+1}_{start}$
            end while
            append $\boldsymbol{D}^{i,j,k}$ to $\boldsymbol{D}^{i,j}$
        end for
        append $\boldsymbol{D}^{i,j}$ to $\boldsymbol{D}^i$
    end for
    append $\boldsymbol{D}^i$ to $\boldsymbol{D}$
    $i = i + 1$
end while
```

Note 2.1.1. The indices i, j, k, l implicitly indicate the connectivity of the branches. In order not to overload the notations, their dependency $i, j(i), k(i,j), l(i,j,k)$ is omitted. However, the index of a next order's branching point, $j(i+1)$, is of interest and therefore simplified by the notation $j' \in [0, N^{i+1} - 1]$, with N^{i+1} the total number of branching points from which parent vessels of order $i+1$ bifurcate. The zeroth order parent vessel ($i = 0$) does not sprout from a branching point, as such i cannot be negative. However, to initialize Algorithm 2.1, the inlet branch is listed at a virtual branching point j^0 to obtain the start branch label $\boldsymbol{b}^0_{start}$ (step 2 in Algorithm 2.1). This initiates the construction of the only ($N^0 = 1$) zeroth order parent vessel.

Example 2.1.1. (Branching topology of a complex vascular tree)
As an example, consider the vascular tree of Figure 2.2. First, a geometrical analysis is performed using the software package VMTK in order to extract a centerline and group the centerline nodes in branches. By applying the pseudo code of Algorithm 2.1, the randomly assigned branch labels are listed in parent vessels of a certain order, Figure 2.3(a), and unambiguously stored in a data structure. This data structure $\boldsymbol{D}$ and the corresponding indices are schematically presented in Figure 2.3(b). Here, the parent vessels of order i are aligned horizontally and they are represented by the same color; the j^{th} branching point of order i is represented by a circular symbol; the k^{th} bifurcating parent branch is represented by the k^{th} square at the lower end of a vertical dashed line; and the l^{th} branch of a parent vessel is represented by a horizontal segment between two branching points or between a branching point and an inlet or outlet. Note that the geometry of Figure 2.2 only includes bifurcations, i.e. branching points with only one side branch.

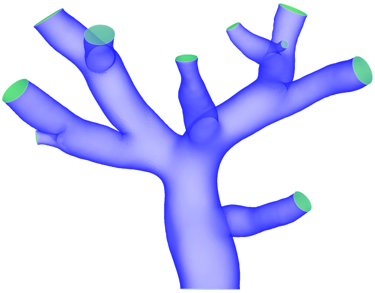
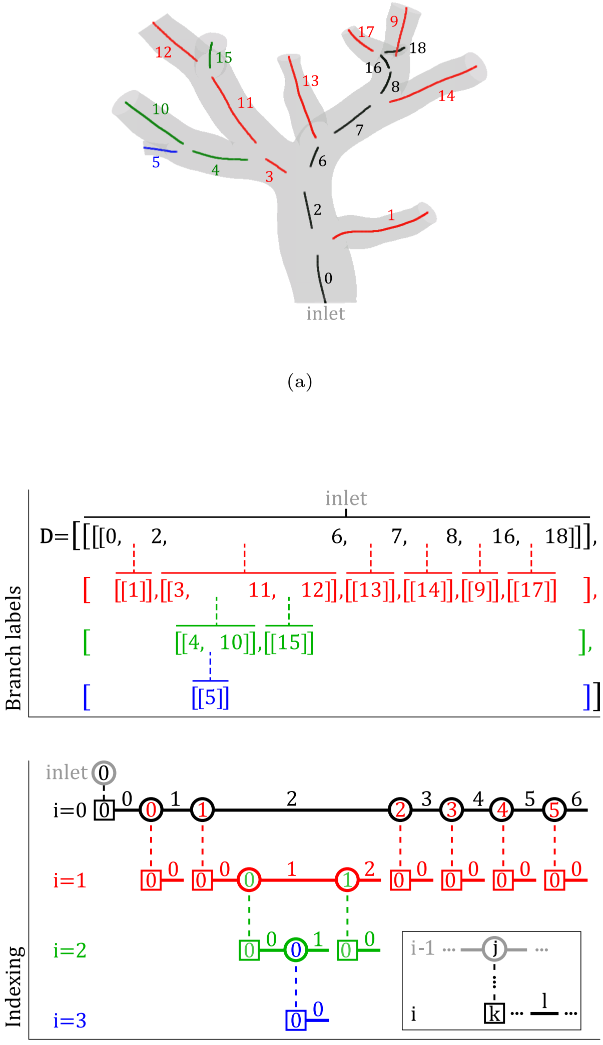


Figure 2.2: Geometry of a complex vascular tree with 1 inlet, 10 outlets, 16 branches and 9 bifurcations. This treelike structure is used to illustrate the methods described in Section 2.1 and Section 2.2.

2.1.2 Square alignment

For each branch, squares are constructed of which the centroids are located at the centerline nodes, of which the normals are oriented parallel to the local axial direction vector and of which one side is oriented parallel to the local transversal direction vector. The local axial direction vectors, $\boldsymbol{e}^{\,m}_{a}$, are set tangent to the centerline, Equation (2.1a), with $\boldsymbol{c}^{\,m}(\in \mathbb{R})$ the coordinates of centerline node



(a)

(b)

Figure 2.3: (a) Branches and branch labels colored by increasing order of the parent vessel i: black, red, green, blue. (b) Schematic representation of the branching topology in the data structure $\boldsymbol{D}$ (top: branch labels $D^{i,j,k,l}$; bottom: corresponding indices $\{i, j, k, l\}$, with respect to Figure 2.1(c))

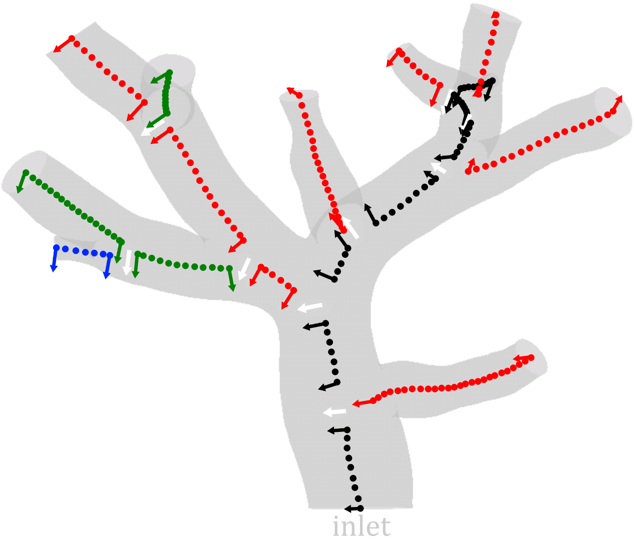


Figure 2.4: Branches of a vascular tree (coloured by increasing parent vessel order: black, red, green, blue), bifurcation normals (white) and projected bifurcation normals (coloured by parent vessel order).

$m \in [0, N^{i,j,k,l} - 1]$ and with $N^{i,j,k,l}$ the number of centerline nodes for the branch with label $D^{i,j,k,l}$. The local transversal direction vectors, $\boldsymbol{e}_t^m$, get calculated such that a smooth transition of the squares is realized in every branch and across the branching points. Therefore, a bifurcation normal, $\boldsymbol{n}^{i,j'}$, is constructed at every branching point between the set of two consecutive branches $[D^{i,j,k,l}, D^{i,j,k,(l+1)}]$ of the same parent vessel i. Each bifurcation normal is constructed normal to the plane defined by the direction vectors of two branches leaving the branching point j': the direction vector $\boldsymbol{v}^{i,j,k,(l+1)}$ of the outgoing parent vessel branch $D^{i,j,k,(l+1)}$, and the direction vector $\boldsymbol{v}^{(i+1),j',0,0}$ of the child branch $D^{i,j',0,0}$. When multiple child branches sprout from the same branching point, the selected child branch is the one for which the direction vector makes the smallest angle with the direction vector of either the incoming parent branch or the outgoing parent branch. This choice will reduce the skewness of the in Section 2.1.3 generated vessel connection blocks, see Equation (2.4). The overall distortion of the final multi-block structure will be further reduced by rotating the bifurcation normals in steps of $\frac{\Pi}{2}$ radians around the direction vector of the outgoing parent branch, until the deflection angle with the upstream bifurcation normal is minimal. When $l > 0$, the upstream bifurcation normal belongs to the same parent vessel and is simply defined by $\boldsymbol{n}^{i,(j'-1)}$, however, when $l = 0$ the upstream bifurcation normal is the one at the branching point from which the branch $D^{i,j,k,0}$ bifurcates, *i.e.* $\boldsymbol{n}^{(i-1),j}$. The bifurcation normals are further processed by a projection onto the planes which are perpendicular to the direction vector of the surrounding branches and by a projection onto the in- and outlet faces of the surrounding in- and outlet branches. Then, they are linearly interpolated along the centerline by Equation (2.1b), $\boldsymbol{n}_p^m$, and used to compute the local transversal direction vector, $\boldsymbol{e}_t^m$, at every centerline node m, Equation (2.1c),

$$\boldsymbol{e}_a^m = \begin{cases} \frac{\boldsymbol{c}^{m+1} - \boldsymbol{c}^m}{\|\boldsymbol{c}^{m+1} - \boldsymbol{c}^m\|} & , \quad m \in \left[0, N^{i,j,k,l} - 2\right] \\ \boldsymbol{e}_a^{N^{i,j,k,l}-2} & , \quad m = N^{i,j,k,l} - 1 \end{cases} \tag{2.1a}$$

$$\boldsymbol{n}_p^m = \boldsymbol{n}_p^0 + \frac{m}{N^{i,j,k,l} - 1}\left(\boldsymbol{n}_p^{N^{i,j,k,l}-1} - \boldsymbol{n}_p^0\right), \quad m \in \left[0, N^{i,j,k,l} - 1\right] \tag{2.1b}$$

$$\boldsymbol{e}_t^m = \begin{cases} \frac{\boldsymbol{n}_p^m \times \boldsymbol{e}_a^m}{\|\boldsymbol{n}_p^m \times \boldsymbol{e}_a^m\|} & , \quad \boldsymbol{n}_p^m \times \boldsymbol{e}_a^m \neq \mathbf{0} \\ \frac{\mathbf{e}_t^{m-1} - \left(\mathbf{e}_t^{m-1} \cdot \mathbf{e}_a^m\right) \cdot \mathbf{e}_a^m}{\|\mathbf{e}_t^{m-1} - \left(\mathbf{e}_t^{m-1} \cdot \mathbf{e}_a^m\right) \cdot \mathbf{e}_a^m\|} & , \quad \boldsymbol{n}_p^m \times \boldsymbol{e}_a^m = \mathbf{0} \end{cases} \tag{2.1c}$$

Each square $\boldsymbol{Q}^{i,j,k,l,m} \in \mathbb{R}^{4\times 3}$ consists of four nodes with three coordinates per node. To radially enclose the surface geometry, they get an edge length relative to the local diameter of the maximum inscribed sphere with a *square scaling factor*

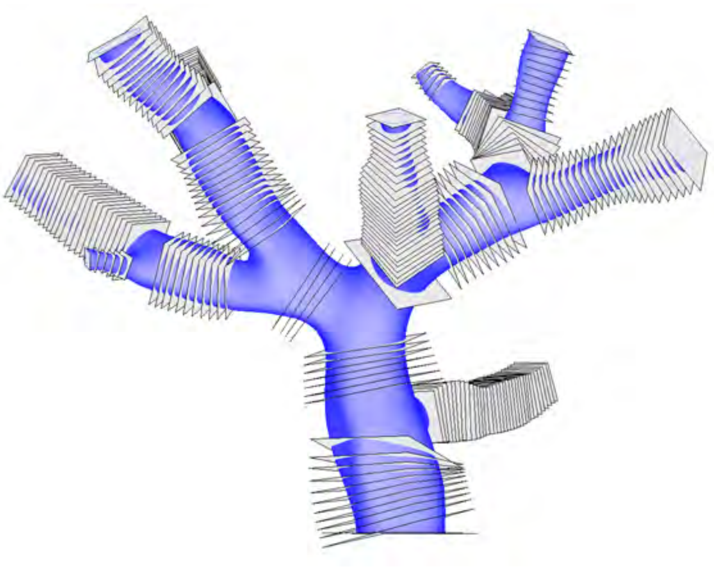


Figure 2.5: Squares positioned along the branches of the vascular tree of Figure 2.2.

f_s [18]. After constructing the squares, corrections are performed by mutually comparing the squares surrounding a branching point and removing those for which the corresponding plane intersects one of the other squares. Compared to the removal of intersecting squares only, this criterion avoids the construction of vessel connection blocks with negative volumes, see Equation (2.4).

Example 2.1.2. (Square alignment along a complex vascular tree)
The bifurcation normals, $\boldsymbol{n}^{i,j'}$, are computed for all nine branching points in the example geometry of Figure 2.2 and shown in white in Figure 2.4. Six of them ($j^1 \in [0, 5]$) are located on the zeroth order parent vessel, two of them ($j^2 \in [0, 1]$) are located on the first order parent vessels, and the final branching point ($j^3 = 0$) is located on the second order parent vessel as illustrated in the indexing panel of Figure 2.3(b).
The projected bifurcation normals, $\boldsymbol{n}_p^0$ and $\boldsymbol{n}_p^{N^{i,j,k,l}-1}$, are generated for every branch and shown in the color of the corresponding parent vessel in Figure 2.4. The projected bifurcation normals are linearly interpolated throughout the branches in order to compute a local transversal direction vector at every centerline node and to align the squares along the centerline, Figure 2.5. The radii of the maximum inscribed spheres at the centerline nodes, which are used to scale the edge lengths, are computed by the VMTK software.

2.1.3 Block generation and interconnection

The squares, as defined in Section 2.1.2, are connected into a multi-block structure (MBS) of hexahedral blocks, based on the branching topology as defined in Section 2.1.1.

First, a block structure is built for the zeroth order parent vessel by interconnecting the consecutive squares of its branches. This procedure starts with the generation of a *curved block structure* for its first branch. As long as this branch is not the last branch of the parent vessel, this step is followed by the generation of a curved block structure for this next (downstream) branch and a *branch connection block*, connecting the curved block structures of both branches. Similar procedures are used to build the parent vessel block structures of order $i > 0$ with the difference that additional connection blocks, towards the corresponding parent vessels of order $i - 1$, are built too. These *vessel connection blocks* are constructed by using the first branch connection block from a FIFO buffer. This branch connection block was added to the FIFO buffer when building the parent vessel block structures of order $i-1$ and is removed from the buffer when moving to the next branching point $j+1$ or when moving to the first branching point ($j'=0$) of the next order ($i+1$).

The generated vessel connection blocks are then axially refined, as if they would consist of an interconnection of squares with an inter-centroid distance which approximates the predefined distance between the centerline nodes, d_0. Finally, the multi-block structure of the vascular tree is axially smoothed – keeping the faces at the inlet, the outlets and the branch connection blocks in place – to avoid self-intersection and to improve the (axially refined) vessel connection blocks.

The construction of the multi-block structure can be automated by applying Algorithm 2.2. In this algorithm, the variables are defined as follows:

- $\boldsymbol{M} \in \mathbb{R}^{N_M \times 8 \times 3}$ represents the *multi-block structure*, with N_M the number of blocks present in the multi-block structure (each with 8 nodes and 3 coordinates per node). By looping through the in Algorithm 2.1 composed data-structure $\boldsymbol{D}$ and sequentially adding the k^{th} bifurcating vessel, at the j^{th} branching point from which the i^{th} order parent vessel bifurcates, to the multi-block structure $\boldsymbol{M}$.
- $\boldsymbol{Q}^{\,i,j,k,l} \in \mathbb{R}^{N_Q \times 4 \times 3}$ represents the *set of squares* corresponding to the branch with label $D^{\,i,j,k,l}$, with N_Q the number of consecutive squares.
- $\boldsymbol{B}^{\,i,j,k,l} \in \mathbb{R}^{N_B \times 8 \times 3}$ represents the *curved block structure* for the branch with label $D^{\,i,j,k,l}$, with $N_B = N_Q - 1$ the number of blocks. It is constructed by interconnecting its consecutive squares, Equation (2.2). The function $f_r : \mathbb{R}^{n \times 4 \times 3} \to \mathbb{R}^{(n-1) \times 8 \times 3}$ reshapes n consecutive squares into a block structure with $n-1$ hexahedral blocks.

$$\boldsymbol{B}^{i,j,k,l} = f_r(\boldsymbol{Q}^{\,i,j,k,l}) \tag{2.2}$$

- $\boldsymbol{C}_B^{i,j,k,l} \in \mathbb{R}^{1 \times 8 \times 3}$ represents a *branch connection block*. When the branch with label $D^{\,i,j,k,l}$ is not the last branch of the vessel, a branch connection block is constructed between its last square, and the first square of the next branch, Equation (2.3). As $\boldsymbol{Q}^{\,i,j,k,l}$ was defined as a set of squares, the last square of a branch and the first square of the next branch are denoted by $\boldsymbol{Q}^{\,i,j,k,l,-1}$ and $\boldsymbol{Q}^{\,i,j,k,l+1,0}$ respectively.

$$\boldsymbol{C}_B^{i,j,k,l} = f_r\big([\boldsymbol{Q}^{i,j,k,l,-1}, \boldsymbol{Q}^{i,j,k,l+1,0}]\big) \tag{2.3}$$

- $\boldsymbol{L} \in \mathbb{R}^{N_{C_B} \times 8 \times 3}$ represents the *buffer list* in which the different branch connections blocks, $\boldsymbol{C}_B^{i,j,k,l}$, get stored once they are computed, with N_{C_B} a

Algorithm 2.2 Algorithm to auto-generate a multi-block structure of a vascular tree based on a set of squares and their connectivity

```
compute data-structure $\boldsymbol{D}$ using Algorithm 2.1
initialize empty list $\boldsymbol{M}$ for the multi-block structure
initialize empty list $\boldsymbol{L}$ for the branch connection blocks buffer
for $i$ from 0 to (size of $\boldsymbol{D}$) $-1$ do
    for $j$ from 0 to (size of $\boldsymbol{D}^i$) $-1$ do
        for $k$ from 0 to (size of $\boldsymbol{D}^{i,j}$) $-1$ do
            if $i > 0$ then
                compute vessel connection block $\boldsymbol{C}_V^{i,j,k,l}(\boldsymbol{L}[0])$, Eq. (2.4)
                append $\boldsymbol{C}_V^{i,j,k,l}$ to $\boldsymbol{M}$
            end if
            for $l$ from 0 to (size of $\boldsymbol{D}^{i,j,k}$) $-1$ do
                compute curved block structure $\boldsymbol{B}^{i,j,k,l}(\boldsymbol{Q}^{i,j,k,l})$ for the branch with
                label $D^{i,j,k,l}$, Eq. (2.2)
                append $\boldsymbol{B}^{i,j,k,l}$ to $\boldsymbol{M}$
                if l $<$ (size of $\boldsymbol{D}^{i,j}$) $-1$ then
                    compute branch connection block $\boldsymbol{C}_B^{i,j,k,l}$, Eq. (2.3)
                    append $\boldsymbol{C}_B^{i,j,k,l}$ to $\boldsymbol{M}$
                    append $\boldsymbol{C}_B^{i,j,k,l}$ to $\boldsymbol{L}$
                end if
            end for
        end for
        if $i > 0$ then
            remove first element from the buffer $\boldsymbol{L}$
        end if
    end for
end for
```

variable amount of branch connection blocks present in the FIFO buffer. These branch connection blocks are extracted from the FIFO buffer to connect parent vessels of order $i > 0$ to parent vessels of the previous order $i-1$ by means of vessel connection blocks.

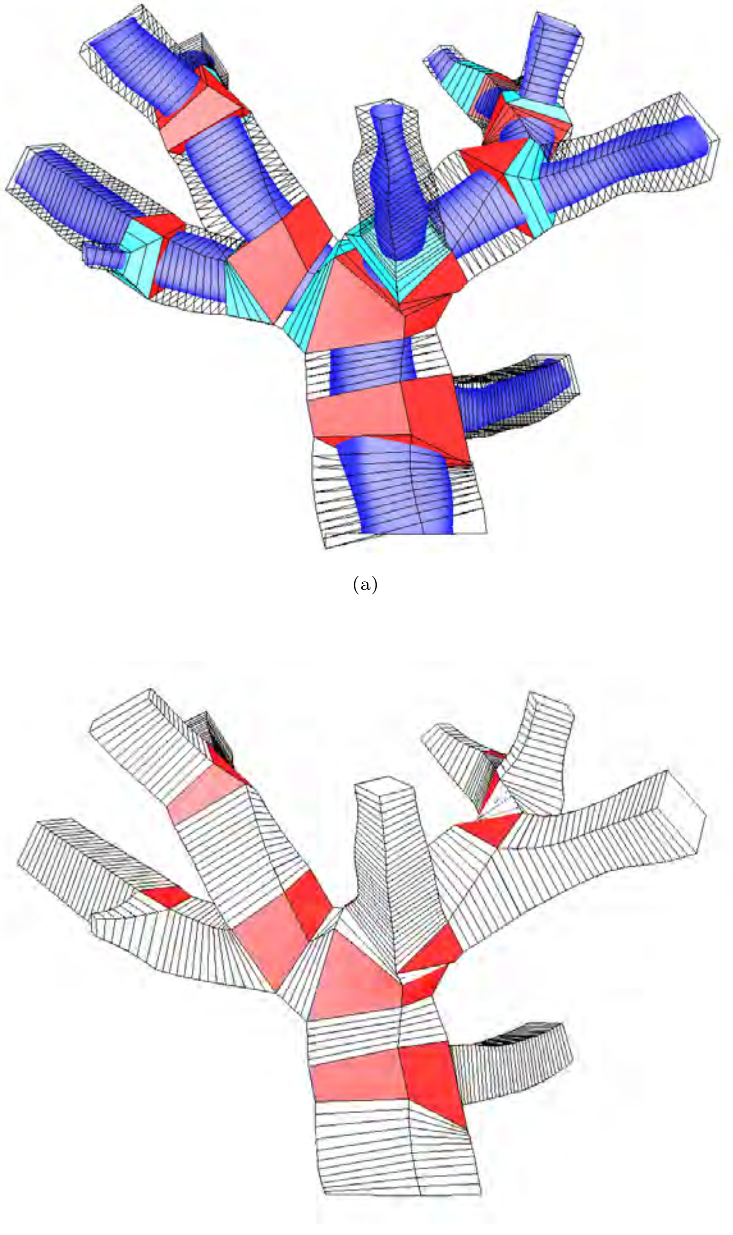
- $\boldsymbol{C}_V^{i,j,k} \in \mathbb{R}^{1\times 8\times 3}$ represents a *vessel connection block* which is constructed by connecting the first square of the first parent vessel branch, $\boldsymbol{Q}^{i,j,k,0,0}$, to the closest face of the first branch connection block in the buffer list $\boldsymbol{L}[0] = \boldsymbol{C}_B^{i-1,j'',k'',l}$, Equation (2.4). With j'' and k'' simplified notations for j^{i-1} and $k^{i-1,j''}$, and with the function $f_c : \left(\mathbb{R}^{(1\times 8\times 3)}, \mathbb{R}^{(1\times 4\times 3)}\right) \to \mathbb{R}^{1\times 4\times 3}$

to return the closest face.

$$\boldsymbol{C}_V^{i,j,k}\big(\boldsymbol{L}[0]\big) = f_r\Big(\Big[f_c\big(\big[\boldsymbol{L}[0], \boldsymbol{Q}^{i,j,k,0,0}\big]\big), \boldsymbol{Q}^{i,j,k,0,0}\Big]\Big) \tag{2.4}$$

Note 2.1.2. Regarding memory usage, the multi-block structure $\boldsymbol{M} \in \mathbb{R}^{N_M \times 8 \times 3}$ is implemented by means of a *node set* $\in \mathbb{R}^{N_N \times 3}$, with N_N the total number of nodes in the multi-block structure, and an *element set* $\in \mathbb{R}^{N_M \times 8}$, which defines the nodal connectivity for each hexahedral element. This provides a more efficient storage as coordinates of nodes which are shared by multiple elements only have to be stored once.

Example 2.1.3. (Multi-block structure of a complex vascular tree)
Based on the branching topology in Figure 2.3, the set of squares in Figure 2.5 are interconnected by applying the pseudo code of Algorithm 2.2. The resulting multi-block structure is depicted in Figure 2.6(a). Consecutive squares of the same branch form curved block structures (transparent); curved block structures of the same parent vessel are interconnected by branch connection blocks (red); and each parent vessel block structure (of order $i > 0$) is connected to a branch connection block of a parent vessel of the previous order $(i-1)$ by means of a vessel connection block (cyan). Most of the generated vessel connection blocks were axially refined, as their axial length was significantly increased by the corrective square removals of Section 2.1.2.
Finally, an axial smoothing is preformed to correct for intersecting squares, Figure 2.6(b), as the centerline at the in- and outlets is not per definition tangent to normal vector of the corresponding cross sectional planes. Note that the squares at the in- and outlets, and the branch connection blocks were fixed during this smoothing procedure, and that too much smoothing affects the curvature of the curved block structures.



(a)

(b)

Figure 2.6: (a) Multi-block structure of the vascular tree of Figure 2.2 with transparent *curved block structures*, *branch connection blocks* in red and (axially refined) *vessel connection blocks* in cyan. (b) The final multi-block structure after an axial smoothing procedure.

2.2 Unstructured hexahedral mesh generation

In this section, a refined multi-block structure (Section 2.2.1) and a radially compressed surface mesh of the vascular tree are provided as input to the well-known grid-based method [34, 61]. The result is an unstructured set of hexahedral volume elements for the core of the fluid domain with cells aligned with the centerline of the arterial network (Section 2.2.2). This unstructured hex-core resembles the volume of the original geometry well and allows the creation of a high-quality boundary layer grid between the interface of the hex-core and the original surface mesh (Section 2.2.3). This completes the generation of a high-quality unstructured hexahedral mesh for the fluid domain.

2.2.1 Multi-block structure refinement

The multi-block structure from Section 2.1 is refined by introducing a grid in every block. The resulting grids are the outcome of a refinement procedure which is based on two parameters: the *axial edge refinement factor*, $f_{edge,a}$ ($\in \mathbb{N}$), and the *transversal edge refinement factor*, $f_{edge,t}$ ($\in \mathbb{N}$). The axial edge refinement factor sets the number of edge subdivisions for the edges in the direction of the centerline. The transversal edge refinement factor defines the refinement of the edges in a plane perpendicular to the centerline and at all the edges of the branch connection blocks (where the axial and transversal directions are not unambiguously defined). The transversal edge refinement factor is kept constant throughout the model to avoid a non-conformal mesh at the interface regions between two refined, consecutive blocks of the multi-block structure. The axial edge refinement factor, however, may vary throughout the model (e.g. see Section 2.4).

Example 2.2.1. (Refined MBS of a complex vascular tree)
Every block in the multi-block structure of Figure 2.6(b) is refined, as illustrated in Figure 2.7. The branch connection blocks are refined by splitting every edge in 14 $(= f_{edge,t})$ uniform parts. As such, they match the transversal refinement of the curved block structures for which each transversal edge is also split in 14 uniform parts. The default factor for the axial refinement is set to 1, however, at the outlet branches with the smallest cross sections $f_{edge,a}$ is adapted to 2.

2.2.2 The multi-block grid-based method

In the grid-based method, an initial Cartesian grid is overlaid on the input surface geometry and the grid cells falling outside the computational domain are removed. To obtain a body-fitted grid, the stair-step surface mesh is projected onto the bounding surface of the input geometry [34, 61]. A mesh resulting from the grid-based method is unstructured and consists of hexahedral volume elements with

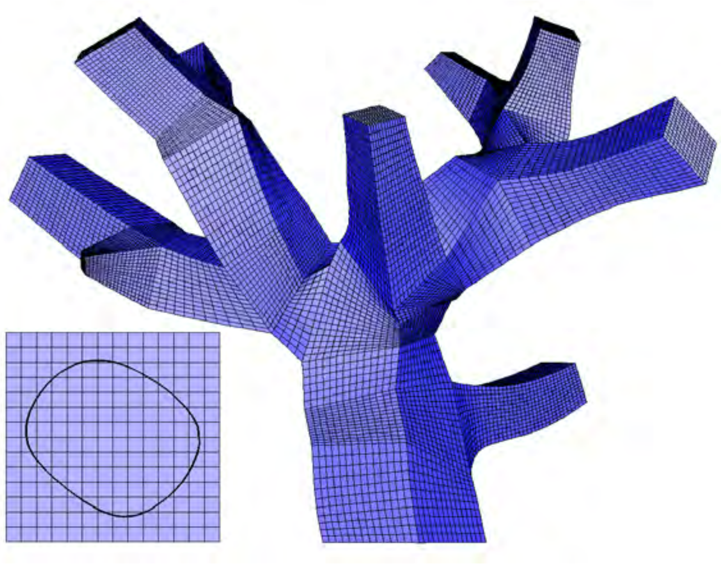


Figure 2.7: Refined multi-block structure, of the vascular tree (Section 2.2.1). A two-dimensional view of the inlet is presented in the lower left corner.

an excellent cell quality in the object interior and near the boundaries that are parallel to one of the coordinate planes of the initial grid. A drawback of this method is the generation of highly skewed cells near the boundaries which are not parallel to one of the coordinate planes. The quality of these cells largely depends on the orientation of the initial grid which is used as input for the grid-based approach. Therefore, the grid-based method is applied onto a refined multi-block structure and not onto a Cartesian grid. This new approach is further denoted by the *multi-block grid-based* (MBGB) method. When a multi-block structure comprises curved block structures which are aligned with the centerline and take into account the local vessel radius, this results in:

- Stretching the cells towards the boundary rather than skewing them.
- Alignment of the grid with the predominant direction of the flow.

To allow a proper projection, the volume of the remaining cells – further denoted as the *hex-core* – is corrected by removing all hexahedral volumes with non-manifold edges (i.e. edges sharing more than two faces of the border mesh, see Figure 2.8). Then, the stair-step surface mesh of the hex-core is smoothed and projected onto the input surface geometry to obtain a body-fitted hex-core.

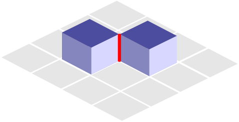
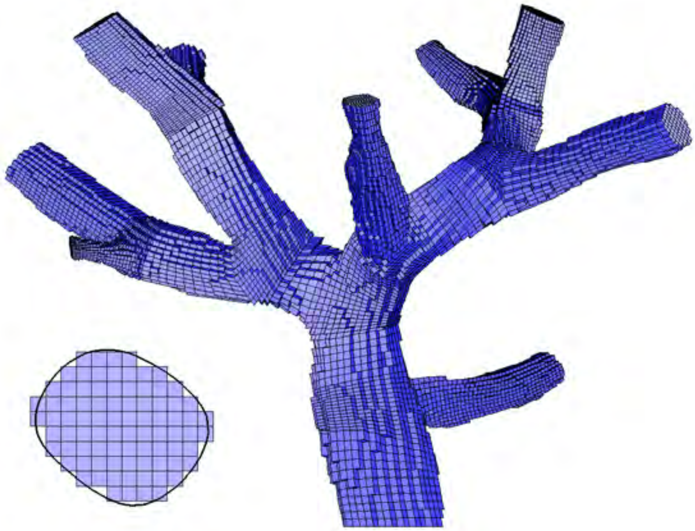


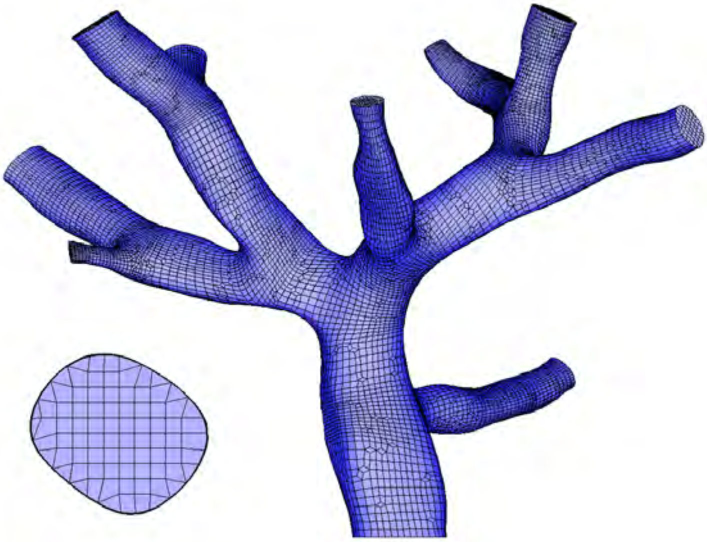
Figure 2.8: The blue cells share a non-manifold edge (red) and are removed from the hex-core to avoid self-intersection of the body-fitted quadrilateral interface mesh.

Note 2.2.1. Special care is taken for the contours at the in- and outlets as they require a projection onto the characteristic edges of the input surface geometry.

Example 2.2.2. (Hex-core of a complex vascular tree)
Using the multi-block grid-based approach, a body-fitted grid is generated for the example geometry of the vascular structure. Therefore, the refined multi-block structure of Figure 2.7 and a (radially compressed, see Section 2.2.3) surface geometry of the vascular structure are used as input for the grid-based method. By removing grid cells, for which the centroids lay outside the provided surface geometry, a hexahedral core for the fluid domain is generated as depicted in Figure 2.9(a). The subfigure in the lower left corner shows the inlet contour of the surface mesh and the cells of the initial grid which are lying inside. The body-fitted mesh is generated by fitting the crenelated surface mesh of the hex-core onto



(a)



(b)

Figure 2.9: (a) Unstructured hex-core generated during the grid-based method. (b) Unstructured hex-core after the body-fitting procedure of the grid-based method (Section 2.2.2). A two-dimensional view of the inlet is presented in the lower left corner of both subfigures.

the input surface geometry, Figure 2.9(b). A preliminary smoothing operation of the crenelated surface mesh was required to obtain a valid (non-self-intersecting) quadrilateral surface mesh.

2.2.3 Boundary layer grid generation

The cell quality at the boundary can be further improved by inserting buffer layers [38, 62, 73]. In this work a similar effect is obtained by performing the grid-based method onto an input surface geometry of which the interface mesh (i.e. the surface mesh excluding the faces at the in- and outlets) is a radially compressed isomorphism of the original (lumen-wall) interface mesh. This allows the creation of a boundary layer grid with a controllable thickness, e.g. by setting the distance between the original and the compressed interface as a function of the local distance-to-centerline.

Construction of the radially compressed surface geometry

To obtain a radially compressed input surface for the multi-block grid-based method, a radial translation of the (original) surface mesh nodes is performed by averaging the inward face normals of the adjacent faces and scaling the normalized vector by a length parameter l^i, Equation (2.5), which indicates the amount of radial displacement at node i.

$$l^i = \begin{cases} R^i\ (1 - f_R) & , \quad R^i \leq R_{max} \\ R_{max}\ (1 - f_R) & , \quad R^i > R_{max} \end{cases} \tag{2.5}$$

In Equation (2.5), R^i represents the distance between node i and the centerline before compression and f_R is the constant ratio $\frac{r^i}{R^i}$, with r^i the nodal distance to the centerline after compression. The piecewise function and the threshold R_{max} are introduced to limit the maximum thickness of the boundary layer region; this can be of use, for instance in aneurysm regions (Section 2.3). Note that the contours at the in- and outlets should stay in their original or characteristic cross sectional planes after translation of the nodal coordinates of the surface mesh. This can be realized by an iterative process in which every iteration performs a part of the total translation, followed by the projection of the contours onto the characteristic planes.

Example 2.2.3. a. (Radial compression of a complex vascular tree)
The triangulated surface mesh of the complex vascular tree in Figure 2.2 is radially compressed with a compression ratio f_R of 0.8. The compressed interface is shown in Figure 2.10, together with the transparent interface of the original geometry colored by the local distance-to-centerline to denote the thickness variation of the

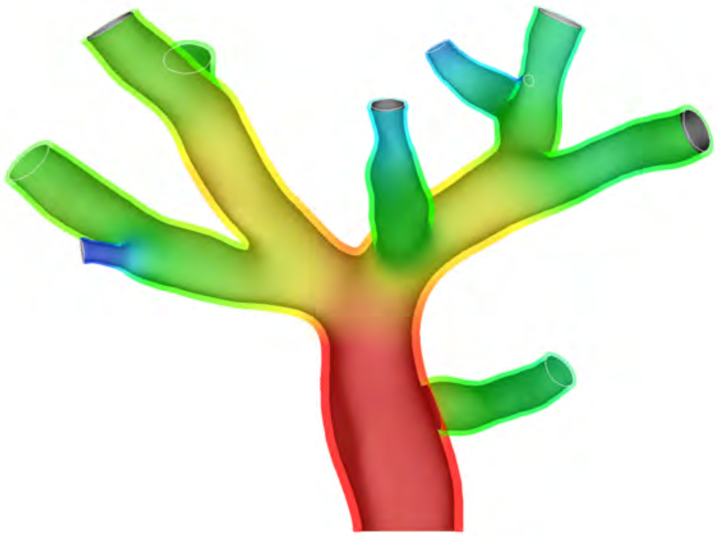


Figure 2.10: Radially compressed interface and boundary layer grid region (colored by thickness).

boundary layer grid. Regarding accuracy, the originally provided interface mesh consists of regular triangles with an increase in mesh density towards the outlets where the vessel diameter reduces. This resulted in a proper shrinkage, even at the high-curvature regions of the bifurcations. However, when the original interface mesh is not uniform with highly non-regular triangles, an additional Laplacian smoothing before every shrink iteration avoids self-intersection and the construction of degenerated surface elements.

Construction of the boundary layer grid

The body-fitting process of the grid-based method projects the interface nodes of the hex-core onto the compressed interface mesh. Isoparametric coordinates for each of these nodes are computed with respect to the face where they are projected onto. As the compressed interface mesh holds – by construction – the same mesh topology as the original interface mesh, isoparametric coordinate transformations map the hex-core interface onto this original interface mesh. The mapped nodes are then connected to the corresponding nodes of the isomorphic hex-core interface, generating one layer of hexahedral cells which can be further refined radially ($f_{BL} \in \mathbb{N}$) to obtain multiple layers at the boundary layer. While keeping the border mesh in place, a final Laplacian smoothing is performed to improve the cell quality at the transition region between the hex-core and the boundary layer grid.

Note 2.2.2. When the faces of the interface meshes are bounded by linear triangles, *e.g.* when the surface mesh is provide in the STL-file format, the mapping is simplified to barycentric coordinate transformations.

Example 2.2.3. b. (Hexahedral fluid mesh of a complex vascular tree)
The hex-core of the complex vascular tree, Figure 2.9(a), is completed with a boundary layer grid in the region between the original interface and the radially compressed interface as shown in Figure 2.11. The subfigure at the lower left corner demonstrates that a Laplacian smoothing of the volume mesh improves the cell quality at the transition region between the hex-core and the boundary layer grid.

2.2.4 Hexahedral grid for the arterial wall

The multi-block grid-based method was originally developed to mesh the fluid domain of complex vascular structures. However, once the quadrilateral interface mesh is generated, a hexahedral mesh for the arterial wall can be constructed by radially expanding this quadrilateral interface mesh into a hexahedral mesh with

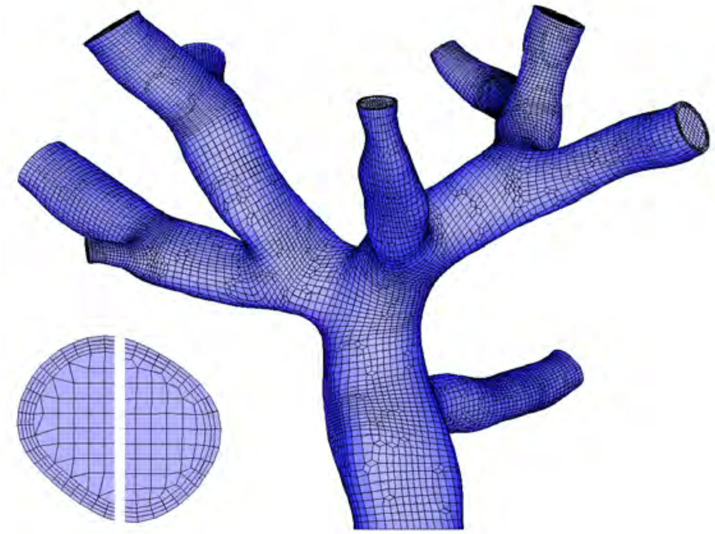


Figure 2.11: Final fluid mesh of the complex vascular tree with a two-dimensional view of the inlet before (left) and after (right) smoothing.

multiple layers using a similar procedure as the one explained in Section 2.2.3. To define the local wall thickness, f_R is set > 1 in Equation (2.5) to denote an expansion ratio instead: a negative translation l^i (< 0) with respect to the inward face normals. Moreover, when a non-conformal mesh is allowed at the fluid-structure interface, the quality of the solid mesh can be further improved. In this regard, the quadrilateral interface is searched iteratively for faces in which two opposing vertices have exactly three adjacent nodes of which two are common. These faces are removed from the element list and the two opposing vertices, are replaced by one new vertex halfway their interconnection, Figure 2.12. After every iteration, the element connectivity list of the mesh is updated.

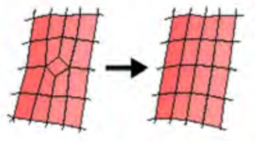


Figure 2.12: Quadrilateral surface element removal at the interface mesh to improve the element quality of the mesh for the arterial wall.

Example 2.2.4. (Hexahedral solid mesh of a complex vascular tree)
The interface mesh of the fluid domain, see Figure 2.11, was cleared from all diamond shaped surface elements – 'diamond shaped' with respect to the overall cell alignment – generating an interface mesh for the solid domain. This high-quality quadrilateral interface mesh was radially expanded as a function of the local distance-to-centerline with an (outer-to-inner wall) diameter ratio $f_R = 1.1$ to obtain a wall thickness to inner wall diameter ratio of 10 %. Three layers were included by a radial refinement. The resulting mesh for the arterial wall is shown in Figure 2.13, together with the fluid mesh.
Note again that the nodes at the in- and outlet boundaries require a projection onto the characteristic planes of the corresponding cross sections to assure a proper mesh. Remark that this implies that the node sets at the in- and outlet boundaries and the normals of the corresponding characteristic planes are known, and can be stored for later preprocessing purposes (*e.g.* to define boundary conditions which only allow radial displacement at the nodes of the in- and outlet boundaries, with respect to the local cylindrical coordinate system).

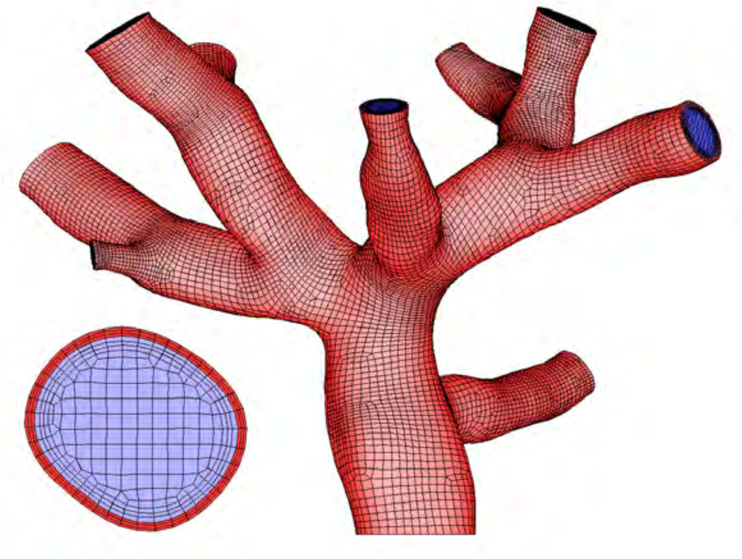


Figure 2.13: Mesh for both the fluid domain and the solid domain of a complex vascular tree. A two-dimensional view of the inlet is depicted in the lower left subfigure.

2.3 Multi-block structure extension

Complex vascular structures have the tendency to contain high-curvature regions. These high-curvature regions are often the regions of interest and require an adequate mesh. Existing hexahedral meshing strategies which combine a refined multi-block structure with a mapping approach include a (non-automated) corrective step to adjust the nodal positions of the (auto-) generated multi-block structure [18, 54, 55].

These nodal corrections lower the geodesic distance between the multi-block structure and the surface geometry on the level of blocks, what improves the subsequent mapping procedure and results in a better alignment of the gridlines with the predominant direction of the flow. However, depending on the degree of curvature, this corrective step deteriorates the quality of the individually adjusted hexahedral blocks, what may lead to low-quality cells in high-curvature regions. In mild stenoses and fusiform aneurysms, the multi-block structure can be generated as a function of the maximum inscribed sphere, but in severe stenoses and saccular aneurysms, the block adjustment can be a time-consuming task for the operator. Moreover, while the block scaling has a positive effect in stenoses by increasing the cell density, the opposite is true for aneurysms where a block scaling will lower the cell density. Compared to obtaining a body-fitted grid with the mapping approach, the MBGB method reduces the change in geodesic distance *i*) on the level of the individual cells by constructing a hex-core and *ii*) without manual interventions.

Therefore, the presented meshing strategy has to be extended with the possibility of introducing additional block-structures to enclose complex regions. Extending the multi-block structure is not limited to aneurysm regions only. In the aneurismatic mouse aorta of Figure 2.14(a) for example, the extension also includes the bifurcation of the coeliac artery and the trifurcation of the right renal artery and the mesenteric artery. In this figure, the multi-block structure is constructed after adapting the by Algorithm 2.1 generated data structure $\boldsymbol{D}$, Section 2.1.1, as suggested in Figure 2.15. Then, based on the bounding box of a separately provided surface mesh of the aneurysm and trifurcation region, an *additional connection block structure*, $\boldsymbol{C}_A^{0,0,0,0} \in \mathbb{R}^{N_A \times 8 \times 3}$ with N_A the number of blocks in this additional connection block structure, is generated to replace the branch with label $D^{0,0,0,1} = 1$ and the corresponding branch connection blocks $\boldsymbol{C}_B^{0,0,0,0}$ and $\boldsymbol{C}_B^{0,0,0,1}$, see Figure 2.14(b). The refined multi-block structure, the resulting hex-core and the final fluid mesh are shown in Figure 2.17. The equiangle skewness

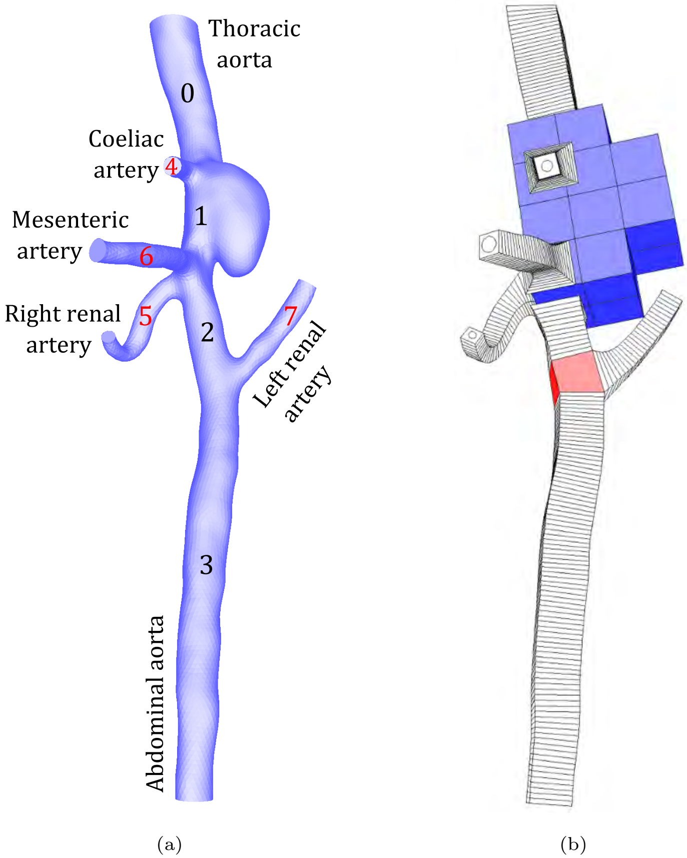


Figure 2.14: (a) Geometry of an abdominal aorta of a mouse, with aneurysm. (b) Multi-block structure with an adapted *branch connection block* (blue) comprising the aneurysm and trifurcation region.

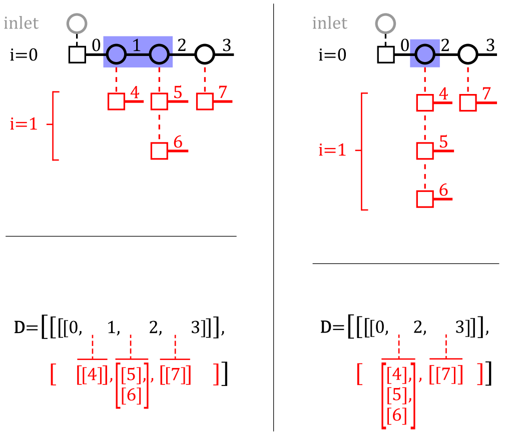


Figure 2.15: (a) Schematic representation of the branching topology and the corresponding datastructure ***D***, before (left) and after (right) adaptation.

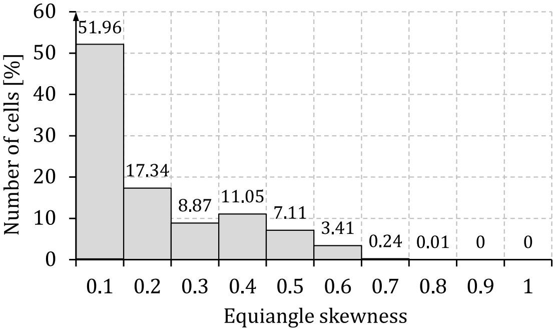
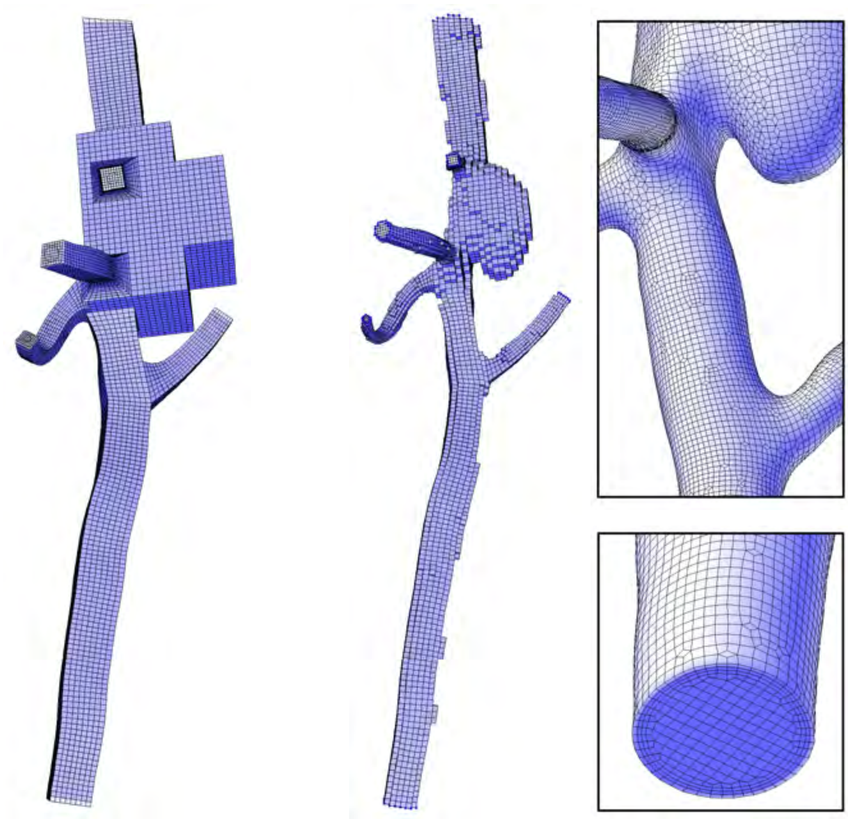


Figure 2.16: The equiangle skewness distribution for the fluid mesh of the aneurysmatic abdominal mouse aorta 2.17(c).



(a) (b) (c)

Figure 2.17: (a) Refined multi-block structure and (b) hex-core of the aneurysmatic abdominal mouse aorta. (c) Detailed views of the generated fluid mesh.

histogram for the fluid mesh – counting $117\,k$ cells – is provided in Figure 2.16.

Note 2.3.1. The nodes available for the construction of a vessel connection block, between an additional connection block structure (instead of a branch connection block) and the first square of a surrounding curved block structure, are not limited to the border mesh nodes of this additional block structure. By defining the transversal edge refinement factor ($f_{edge,t}$), the border mesh of the additional block structure can be refined, what introduces new grid points. These can be used to connect the surrounding curved block structures to, creating high-quality vessel connection blocks which fully enclose the geometry of the side branches.

2.4 Mesh density control and automation

2.4.1 Global mesh density parameters

In theory, the only parameters required to generate a fluid mesh for a vascular tree – using the MBGB approach – are those which define the mesh density. As there can only be one constant value for the transversal edge refinement factor $f_{edge,t}$ it has a global impact on the average number of cells in the cross section of the core as the edge lengths of the squares are set relative to the local diameter of the original surface geometry, with a scaling factor $f_s > f_R$. To fully define the mesh density at a cross section, a list of parameters $\boldsymbol{f}_{BL}$ has to be provided to define the (non-uniform) radial subdivision in the boundary layer region for which the thickness is defined by the radial compression ratio f_R, Equation (2.5). Towards automation of the meshing procedure, it is recommended to set a constant distance between the centerline nodes, d_0, as it is required at an early stage of the procedure to construct the consecutive squares. In a later refinement step, a parameter list $\boldsymbol{f}_{edge,a}$ can be provided to introduce local axial refinements along the branches and to obtain cell aspect ratios close to one. A default factor, $f^i_{edge,a}$, can be defined for every block i of the multi-block structure as a function of $f_{edge,t}$ and the corresponding block size, Equation (2.6).

$$f^i_{edge,a} = round\left(\frac{\overline{l^i_{edge,a}}}{\overline{l^i_{edge,t}} \Big/ f_{edge,t}} \right) \tag{2.6}$$

In Equation (2.6), $\overline{l^i_{edge,a}}$ and $\overline{l^i_{edge,t}}$ are the mean axial and transversal edge lengths of block i, with $i \in [0, N-1]$, and the round function rounds to an integer value. The mean value of the edge lengths is used as the blocks enclosing the vessel geometry are only in the limit square prisms. This limits the characteristic cell aspect ratios for block i, defined by Equation (2.7), to the interval $[0.5, 1.5[$.

$$AR^i = \frac{\overline{l^i_{edge,a}} \Big/ f^i_{edge,a}}{\overline{l^i_{edge,t}} \Big/ f_{edge,t}} \tag{2.7}$$

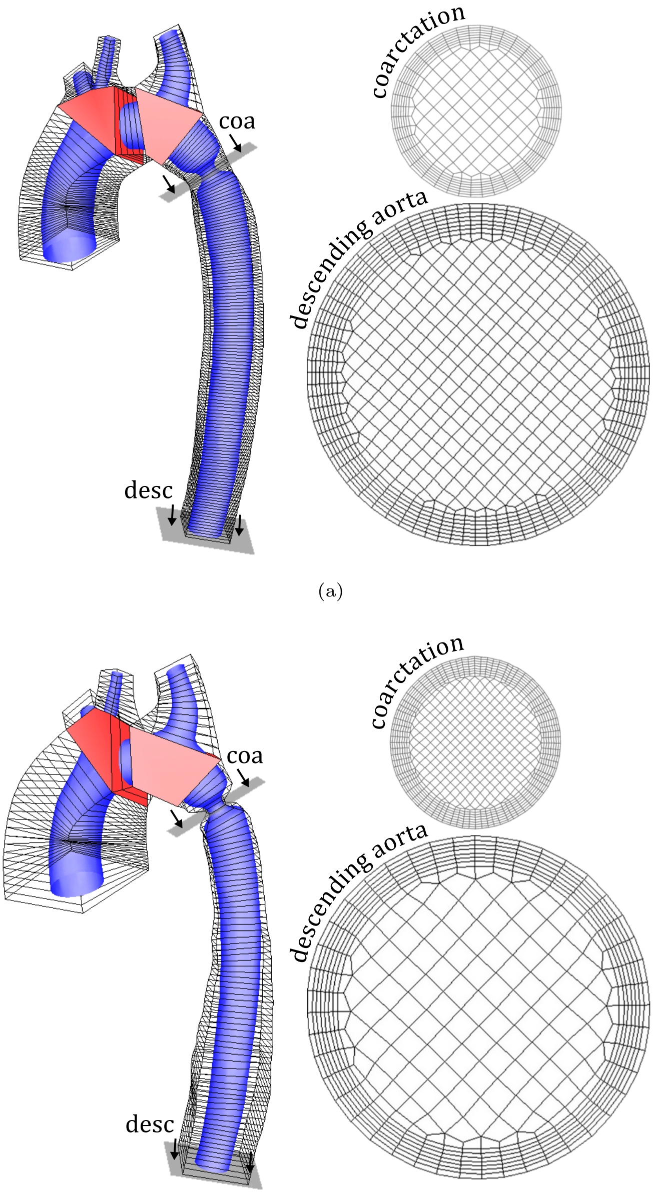
A constant transversal edge refinement factor can be predefined, Equation (2.8), by rewriting Equation (2.7) such that an aspect ratio close to 1 is obtained at the block with the maximum mean transversal edge length ($\overline{l_{edge,t,max}}$) without axial refinement ($f_{edge,a,max} = 1$). Where $\overline{l_{edge,a,max}}$ is approximated by the distance between the centerline nodes d_0 and where $\overline{l_{edge,t,max}}$ is approximated by the maximum diameter of the maximum inscribed spheres D_M, scaled by f_s.

$$f_{edge,t} = round\left(\frac{f_s D_M}{d_0}\right) \tag{2.8}$$

With default values of 1.1 for the square scaling factor f_s and 0.8 for the radial compression ratio f_R, only two parameters, d_0 and f_{BL}, are required to autogenerate a mesh for the fluid domain.

2.4.2 Local refinement

In practice, an efficient mesh of a complex geometry is realized by adaptive refinements. Apart from adapting the default factors of the local axial mesh refinement, $f^i_{edge,a}$, as was done at the outlets of Figure 2.7, it is also possible to control the local mesh density at cross sections by modifying the square scaling factor f_s on a local base. Increasing the overall square scaling factor and lowering its value in the region of interest, locally increases the cross sectional cell density of the hexcore and the circumferential cell density at the boundary layer grid. In contrast to Figure 2.18(a), this has been performed in Figure 2.18(b) at the stenosis region of an aortic coarctation. In Chapter 4, a numerical example is provided that shows that this adaptive refinement results in a more efficient computational grid.



(a)

(b)

Figure 2.18: The cross sectional grids of the fluid mesh at the coarcation (coa) and the descending aorta (desc), which result from a multi-block structure without (a) and with (b) local grid refinements.

2.5 Conclusion

This chapter proposes a novel strategy to generate unstructured hexahedral grids for the fluid domain of complex treelike structures, based on only six parameters of which four are mandatory: the distance between the centerline nodes (d_0), the square scaling factor (f_s), the radial compression factor (f_R) and the number of boundary layers (f_{BL}), and two are optional: the transversal edge refinement factor ($f_{edge,t}$) and the axial edge refinement factor ($f_{edge,a}$).

In summary, for a given surface mesh of a vascular tree geometry, the centerline is computed together with the radii of the maximum inscribed spheres. Each set of centerline nodes is further referred to as a 'branch' and corresponds to a vessel segment between two consecutive branching points or between a branching point and an inlet or outlet. The connectivity between the branches is extracted from the centerline data and used to unambiguously define the complete branching topology in terms of 'parent vessels'. For each parent vessel, squares are constructed of which the centroids are located at the centerline nodes, of which the normals are tangent to centerline and of which one side is oriented parallel to the local transversal direction vector, which gets calculated such that a smooth transition of the squares is realized in every branch and across the branching points. Relative to the local diameters of the maximum inscribed sphere, the squares are scaled to radially enclose the surface geometry and interconnected to form curved block structures for the different parent vessels. Based on the branching topology, these parent vessel block structures are auto-connected to obtain a multi-block structure of the vascular tree. The multi-block structure is refined by introducing a grid in every block. The resulting grid is the outcome of a refinement procedure in which the edge refinement parameters can be modified from their default values to control the global mesh density. The refined multi-block structure and a radially compressed surface mesh of the vascular tree are provided as input to the well-known grid-based method. As the multi-block structure comprises curved block structures which are aligned with the centerline and take into account the local vessel radius, the result is an unstructured set of (centerline aligned) hexahedral volume elements for the core of the fluid domain. This unstructured hex-core resembles the volume of the original geometry well and allows the creation of a high-quality boundary layer grid between the interface of the hex-core and the original surface mesh, even in regions of high curvature. Finally, the methodology allows for local refinements in both the axial direction and the cross sections, and is able to handle aneurysms by extending the multi-block structure with additional blocks.

Chapter 3

Validation of the extended TreeMesh method based on the results of a CFD variability study[1]

3.1 Introduction

Numerous studies have been performed in which image-based CFD predictions are validated against careful in vitro measurements [23, 56, 79] and *in vivo* measured data [7, 57, 70], but only few studies in the biomedical field report the sensitivity of CFD predictions to different solvers and solution strategies [52, 69]. We participated in a study, led by Steinman and Loth, which sought to assess the variability in the CFD prediction of the pressure and the flow in an image-based giant carotid aneurysm with a proximal stenosis by a wide variety of research groups [68]. The study was stimulated by a controversy regarding pressure drops predicted in the original study of Cebral et al. [13]. The so-called SBC2012 CFD challenge consisted of two phases. In phase I, the lumen geometry was specified together with realistic flow rates and fluid properties, leaving each research group to choose their solver, discretization, and solution strategy. While setting up the numerical models for phase I, an in vitro model of the same geometry was constructed for phase II, from which pressure and flow rates were measured. Groups repeated

[1] This chapter is based on the co-authored work *"Variability of Computational Fluid Dynamics Solutions for Pressure and Flow in a Giant Aneurysm: The ASME 2012 Summer Bioengineering Conference CFD Challenge."*, as published in the *Journal of Biomechanical Engineering* 135(02): 021015, 2013 [68]

their simulations using a geometry reconstructed from a micro-computed tomography (CT) scan of the physical model together with the measured flow rates and fluid properties. The variability among the solutions was assessed for both phases by having each group interpolate their results onto a standardized mesh and centerline.

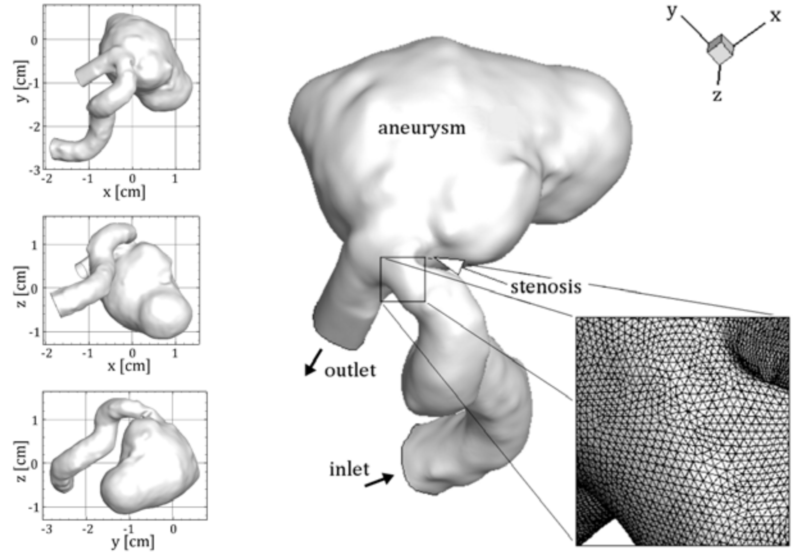
Twenty-five different research groups located over the whole world (12 European, 9 North American, 3 Asian and 1 South American) participated in the first phase of the CFD variability study. They contributed twenty-seven complete phase I solutions of which seventeen used commercial solvers (11 Fluent, 2 Star-CCM+, 2 CFX, 1 FIDAP and 1 Star-CD), five were based on open-source platforms (2 Nεκταr, 1 caffa3D.MB, 1 FEnICS and 1 OpenFOAM), and five used in-house solvers. Where five solutions were based on the finite element method, two on the spectral element method, and one on the finite difference method, the majority of solutions was based on the finite volume method (19). Although the geometry and the inlet flow rates were prescribed, see Section 3.2.1, participants were free in applying flow extensions and in their choice of boundary conditions at the in- and outlet. With a standard traction free outlet condition in most of the cases, twelve groups reported solutions based on a plug velocity profile and the rest reported fully developed velocities profiles (7 Poiseuille and 8 Womersley). Regarding the grid type, the majority (19) employed tetrahedra-dominant meshes, followed by five hexahedra-dominant meshes (including our body-fitted full hexahedral grid, see Section 3.3.2), two polyhedral meshes, and two uniformly spaced Cartesian grids with immersed boundaries. For phase II, twenty-four complete solutions were received.

As mentioned above, the primary objective of this CFD Challenge was to determine the variability of CFD solutions in a case where all participants were provided with the same lumen geometry and flow rate boundary conditions, but were given no guidance regarding solution strategy, discretization, etc. Regarding the provided data, the required simulations and the requested data sets for central processing, a detailed overview of the study is provided in Section 3.2. A secondary objective was to confirm or refute the pressure drops predicted in the paper by Cebral et al. [13]. In this chapter however, the aim is to describe our solution strategy in detail (Section 3.3) and to compare our simulation results with the solutions contributed by the other participating groups (Section 3.4). As our solution strategy is distinguished from the others by the use of a full hexahedral, body-fitted and flow aligned grid (Chapter 2), this allows us to validate the use of these grids together with our computational fluid dynamics models for vascular applications, and to justify our CFD modelling approach in Chapters 4 and 8.

3.2 Details of the CFD variability study

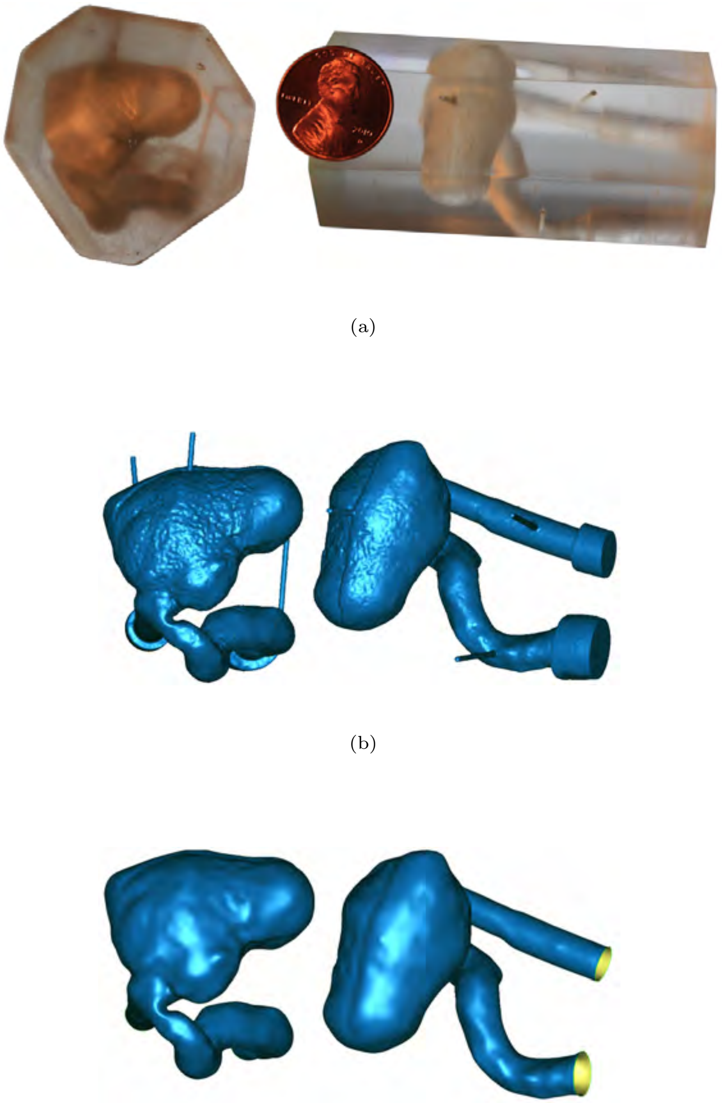
3.2.1 Lumen geometry and flow conditions

A giant cerebral aneurysm with a stenosis proximal to the aneurysm ostium was selected as case study for the CFD challenge and provided by Cebral et al. [13]. The phase I geometry of the lumen was provided as a surface mesh to the CFD challenge participants in (ASCII) STL format, after refining it up to 325000 triangles to allow higher resolutions than in the original case. The geometry and surface mesh are presented in Figure 3.1(a), note the complexity of the geometry featuring (from inlet to outlet) a tortuous vessel with a narrowing proximal to a giant saccular aneurysm with an irregular shape. The minimum cross-sectional area at the stenosis is reduced by 69 %, relative to the inlet area of 24.45 mm^2. Towards phase II, a physical model of the lumen – based on the phase I geometry – was constructed for in vitro measurements, Figure 3.2(a). The surface recon-



(a)

Figure 3.1: The phase I geometry. Views along each of the coordinate axes, indicating the size (left). Isometric view with inlet, outlet, stenosis and aneurysm annotations (center). Detailed view of the triangulated surface mesh (right).



(a)

(b)

(c)

Figure 3.2: The phase II geometry. (a) Physical flow model in polyester resin, cut to size for micro CT scanning. (b) 3D surface reconstruction of the lumen in the micro CT scanned physical model. (c) Phase II surface geometry provided to the CFD challenge participants. Figures were taken from [68].

struction of the micro-CT scanned model, shown in Figure 3.2(b), was provided for phase II after pressure port removal, surface smoothing and end port clipping, Figure 3.2(c). For both phases, the pulsatile inflow was available by means of a Fourier decomposition of the flow rate waveform. This allowed the CFD challenge participants to prescribe consistent flow rate boundary conditions at any temporal resolution. Newtonian blood viscosity was assumed and provided together with the blood density (Table 3.1). Note that no recommendations or constraints were provided regarding the choice of volume elements, mesh densities, temporal resolution or inflow and outflow boundary conditions.

	Phase I	Phase II
Density $\left[\frac{kg}{m^3}\right]$	1000	1113
Dynamic viscosity $[Pa\,s]$	0.004	0.00401
Period of cardiac cycle$[s]$	0.99	1.164
Moment of peak-systole $[s]$	0.275	0.052

Table 3.1: Prescribed fluid properties and flow parameters, for both phase I and phase II

3.2.2 Overview of the simulations

Multiple simulations were performed to assess the variability in predicted pressures among different solution strategies. In phase I, the fluid properties and flow rates were based on the article by Cebral et al. [13]. To allow comparison with an in vitro model, the fluid properties, the flow rates and the pressures (at the inlet, outlet and aneurysm) were measured for phase II. For each phase, two pulsatile flow rates – having a similar waveform, but shifted in magnitude – were specified to quantify to which extent the choice of flow rate might affect the pressure drop. The simulation with the lower (higher) inflow will be further denoted by pulsatile 1 (pulsatile 2). AV1 (AV2) and PK1 (PK2) will be used to refer to either the corresponding cycle-averaged or the peak systolic values. To test the quasi-steady assumption for the pressure drop and to assess the corresponding variability in the predicted pressures among different solution strategies, separate simulations (SS1, SS2, SS3, and SS4) were performed under steady flow rates (corresponding to AV1, AV2, PK1, and PK2 conditions), see Table 3.2 for an overview of the twelve simulations performed.

3.2.3 Postprocessing details

After preliminary spatial and temporal refinement studies, see Section 3.3.2.3, six simulations were performed for each phase. Variability was assessed by having

		Flow rates $\left[\frac{ml}{s}\right]$	
	Cases	Phase I	Phase II
Unsteady	AV1 (PK1)	5.13 (9.14)	4.40 (9.34)
	AV2 (PK2)	6.41 (11.42)	6.16 (11.25)
Steady state	$\mathrm{SS1}_{\mathrm{AV1}}$	5.13	5.10
	$\mathrm{SS2}_{\mathrm{AV2}}$	6.41	6.36
	$\mathrm{SS3}_{\mathrm{PK1}}$	9.14	9.17
	$\mathrm{SS4}_{\mathrm{PK2}}$	11.42	11.36

Table 3.2: All phase I and phase II simulations characterized by their prescribed flow rate $\left[\frac{ml}{s}\right]$. For the unsteady simulations, both the cycle averaged (AV) and the peak systolic (PK) values are mentioned. For the steady state simulations, the corresponding pulsatile case is denoted in the subscript.

each group interpolate their results onto a predefined centerline, reference volume and surface mesh. As such, the following data was requested:

- Pressure and velocity magnitude at about 250 points along a specified centerline. (For quantification of the variability of the pressure drop and the velocity along the centerline in the cycle-averaged, the peak systolic and the steady state solution; and for the evaluation of the quasi-static assumption.)
- Pressure and velocity magnitude versus time (for one cardiac cycle) at five selected points: inlet, stenosis, aneurysm neck, aneurysm sac and outlet. (For quantification of the variability in the pressure and the velocity waveforms. For the comparison with the experimentally-measured pressure drops, the reader is referred to the discussion section of the original article [68].)
- Surface pressures interpolated on a reference surface mesh. (For visualization of the cycle-averaged and the peak systolic pressures of both pulsatile flow cases.)
- Velocity magnitudes interpolated on a reference volume mesh. (For visualization of the cycle-averaged and the peak systolic flow patterns of both pulsatile flow cases.)

The results regarding these data sets are presented in Section 3.4.

3.3 Solution strategy

3.3.1 Solver settings and boundary conditions

All simulations have been performed with the commercial finite volume CFD solver Fluent 12.1 (Ansys). The pressure and velocity fields were solved using the SIMPLE algorithm with second-order upwind discretization for the momentum equation and standard pressure interpolation (i.e. according to the momentum equations coefficients). For the transient solutions, an implicit time integration scheme was used with first order accurate temporal discretization. Convergence was obtained when the scaled residuals of continuity and momentum decreased below a value of $10\,e$-6. At the inlet of the fluid domain a plug flow with specified mass flow rate was applied, and a constant static pressure of $0\,Pa$ was imposed at the outlet.

3.3.2 Grid generation

The use of a full hexahedral body-fitted grid differentiates our contribution to the CFD variability study from those of other participants. Two different hexahedral meshing strategies were applied: a mapping approach and a multi-block grid based approach. The generated grids feature grid alignment with the predominant flow direction at the parent vessel and an increase in cell density at the stenosis.

3.3.2.1 Accuracy of flow aligned grids

The so-called transport equation for property ϕ is presented in Equation (3.1), with ρ the density, $\boldsymbol{v}$ the velocity vector and Γ the diffusion coefficient. It contains the various transport processes such as the rate of change term and the convective term on the left hand side and the diffusive term and the source term on the right hand side.

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla.(\rho\boldsymbol{v}\phi) = \nabla.(\Gamma\nabla\phi) + S_\phi \tag{3.1}$$

To assess the efficacy of flow aligned grids on the accuracy of the CFD solution, we focus on the two-dimensional steady-state advective transport of a scalar ϕ in an incompressible flow, Equation (3.2), through a rectangular domain with $\boldsymbol{v} = (\,u \quad v\,)^T$ and $u, v \geq 0$.

$$u\frac{\partial\phi}{\partial x} + v\frac{\partial\phi}{\partial y} = 0 \tag{3.2}$$

Consider a uniform grid, which subdivides the domain into a finite number of non-overlapping cells and of which the cell centroids store the discrete values of ϕ. Using the finite volume method, the governing differential equation, $\nabla.(\rho\boldsymbol{v}\phi) = 0$, is integrated over each three-dimensional control volume (CV), or its corresponding control surface (CS) after applying the divergence theorem, Equation (3.3), with $\boldsymbol{n}$ the outward pointing unit normal vector at the bounding surface.

$$\int_{CV} \nabla.(\rho\boldsymbol{v}\phi)\,dV = \oint_{CS} (\rho\boldsymbol{v}\phi).\boldsymbol{n}\,dA \tag{3.3}$$

For a cell with centroid P, neighbouring cell centroids N, E, S, W and cell face

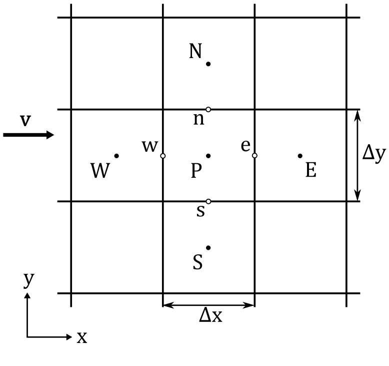


Figure 3.3: CartesianGrid

centroids n, e, s, w, as shown in Figure 3.3, with Δx and Δy the constant edge lengths of the cells in the x and the y direction respectively, integration of the convective term over the closed surface results in

$$\big((\rho u\phi)_e - (\rho u\phi)_w\big)\Delta y + \big((\rho v\phi)_n - (\rho v\phi)_s\big)\Delta x = 0 \tag{3.4}$$

When evaluating Equation (3.4) by a first-order upwind discretization, the values of ϕ at the cell faces are approximated by the values of ϕ in the upstream cells.

$$\begin{cases} \phi_n = \phi_e = \phi_P \\ \phi_s = \phi_S \\ \phi_w = \phi_W \end{cases} \tag{3.5}$$

Substituting these assumptions in Equation (3.4) and dividing by $\Delta x\,\Delta y$, yields the following discretized equation

$$u\frac{\phi_P - \phi_W}{\Delta x} + v\frac{\phi_P - \phi_S}{\Delta y} = 0 \tag{3.6}$$

Note that Equation (3.6) is no longer exactly modeling the continuous Equation (3.2) by assuming a piecewise constant variation of ϕ from cell to cell, Equation (3.5). Therefore, let us consider the Taylor series truncation error associated with the upwind discretization scheme. Using a Taylor series expansion of the functions $\phi(x - \Delta x) = \phi_W$ and $\phi(y - \Delta y) = \phi_S$ about the point $P(x, y)$, we may write

$$\phi_W = \phi_P - \Delta x\left(\frac{\partial\phi}{\partial x}\right)_P + \frac{\Delta x^2}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_P - \quad ... \tag{3.7a}$$

$$\phi_S = \phi_P - \Delta y\left(\frac{\partial\phi}{\partial y}\right)_P + \frac{\Delta y^2}{2}\left(\frac{\partial^2\phi}{\partial y^2}\right)_P - \quad ... \tag{3.7b}$$

Rearranging above equations results in,

$$\frac{\phi_P - \phi_W}{\Delta x} = \left(\frac{\partial\phi}{\partial x}\right)_P - \frac{\Delta x}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_P + \quad ... \tag{3.8a}$$

$$\frac{\phi_P - \phi_S}{\Delta y} = \left(\frac{\partial\phi}{\partial y}\right)_P - \frac{\Delta y}{2}\left(\frac{\partial^2\phi}{\partial y^2}\right)_P + \quad ... \tag{3.8b}$$

After substituting the above expressions into the discrete Equation (3.6), the original differential Equation (3.2) can be rearranged to the left hand side, revealing the truncation error of the two-dimensional first-order advection equation at the right hand side:

$$u\left(\frac{\partial\phi}{\partial x}\right)_P + v\left(\frac{\partial\phi}{\partial y}\right)_P = \frac{u\Delta x}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_P + \frac{v\Delta y}{2}\left(\frac{\partial^2\phi}{\partial y^2}\right)_P + ... \tag{3.9}$$

The truncation error is of order $\mathcal{O}(\max\{\Delta x, \Delta y\})$, as expected for a first-order scheme. This error term looks similar to the diffusive term in the convection-diffusion Equation (3.1). Although a pure convection problem is being solved, the upwind scheme actually brings some diffusion into the model as can be seen by comparing the continuous Equation (3.2) with Equation (3.9), which is derived from the discrete Equation (3.6). The components of this ‘false diffusion’ – or ‘numerical viscosity’ in the context of the momentum equations (with $\phi \in \{u, v\}$) – are proportional to the coefficients $\frac{u\Delta x}{2}$ and $\frac{v\Delta y}{2}$ and thus with the grid size which is defined by Δx and Δy. As such, the dissipation across the streamlines can be reduced by refining the grid. However, this entails an increase in the number of degrees of freedom in the system, what is reflected in an increase in the overall wall-clock time of the simulation.

Another approach to reduce the false diffusion exists in aligning the grid lines with the predominant direction of the flow. If the flow is aligned with the grid, either u or v is equal to zero. Say $u > 0$ and $v = 0$, the corresponding component of the leading order error term in Equation (3.9) disappears as $\frac{v\Delta y}{2} = 0$. The other component of the leading order error term disappears when the variation of the property ϕ along the streamlines can be neglected. In this case e.g. $\left(\frac{\partial^2 \phi}{\partial x^2}\right)_P = 0$ for a fully developed laminar flow in a straight channel aligned with the x-direction. The significant reduction in the amount of dissipation across the streamlines implies that effort in an adequate grid alignment procedure is rewarded with an improved accuracy of the upwind scheme [47, 81, 85]. Despite the efficacy of this approach without increasing the computation time, it is less frequently used in practice than increasing the number of cells or the order of the spatial discretization; especially when complex geometries are involved.

3.3.2.2 The mapping approach

A hexahedral grid for the fluid domain of phase I was generated by a mapping approach using an in-house pyFormex code, developed by De Santis et al. [15, 18].

1. First, a curved block structure is created inside the parent vessel by concatenating single blocks, colored white in Figure 3.5(a).
2. As if it were a side branch, the aneurysmal cavity gets filled by creating another curved block structure, colored blue in Figure 3.5(a). Instead of appending single blocks, sets of two adjacent blocks are concatenated to limit the decrease in mesh density inside the aneurysm after mapping and to improve the final mesh quality.
3. Then, an external multi-block structure is created by scaling the internal multi-block structure and by manually optimizing the location of the individual nodes.

4. A first refinement factor 'r_1' is set to subdivide the volume of each block of the internal multi-block structure into $r_1 \times r_1 \times r_1$ elements and to split the faces of each block of the external multi-block structure into $r_1 \times r_1$ faces. This is shown for the internal multi-block structure in Figure 3.5(b) for $r_1 = 4$.
5. The refined surface mesh of the external block structure is then mapped onto the surface geometry of the lumen.
6. A volume mesh is created by connecting the (mapped) quadrilateral surface mesh to the refined border mesh of the internal multi-block structure, which holds the same topology. This connection is refined in radial direction with the same refinement factor r_1, however, with a non-uniform refinement towards the vessel wall when $r_1 > 3$ to introduce a boundary layer grid.
7. Finally, to reach the desired resolution, all elements of the hexahedral mesh are isotropically refined with a factor 'r_2', Figure 3.5.

For further details about this methodology, the reader is referred to the Treemesh chapter in [15]. Using this procedure, three coarse grids were constructed ($r_1 = 3$ and $r_2 = 3$ to 5), as well as five finer grids including a boundary layer grid ($r_1 = 5$ to 8 and $r_2 = 4, 5$). This resulted in meshes with approximately $93k$, $221k$, $432k$, $1024k$, $2000k$, $3456k$, $5488k$ and $8912k$ hexahedral cells, in meshes with a higher mesh density at the stenosis region, and in grid lines which are aligned with the predominant direction of the flow in the parent vessel.

3.3.2.3 Spatial and temporal resolution

Minimum and maximum edge lengths of the studied meshes are listed as $L_{edge,min}$ and $L_{edge,mean}$ in Table 3.3, together with the results of a mesh refinement study. As the focus was put on the pressure drop, E_{mean} and E_{max} represent the mean and the maximum error of the converged pressures values along the given centerline nodes, relative to pressure values obtained with the reference mesh (i.e. the finest grid with almost 9 million cells). This was done for the steady state simulation SS2 of phase I in which the highest averaged flow rate (6.41 $\frac{ml}{s}$) is applied. The grid counting $1024k$ cells was selected for further simulations and showed a mean error on the pressure of 0.052 % and a maximum error of only 0.2 %, with respect to the finest grid. This corresponds to a mean spatial resolution of 0.176 mm; compared to the other CFD challenge grids, this is close to the median value of 0.18 mm with an interquartile range (IQR) of 0.15 mm to 0.26 mm, in which half of the solutions are defined. Note that the coarsest grid with only $92k$ cells performs good as well and results in a maximum error below 0.5 %.

To assess the temporal resolution for the unsteady simulations, multiple phase I pulsatile 2 simulations – characterized by AV2(PK2) – were set up for different time steps ($\Delta T = 0.055\,s$, $0.0275\,s$, $0.011\,s$, $0.0055\,s$, $0.00275\,s$, $0.001375\,s$, $0.0006875\,s$), all with a time point at peak-systole ($t_{systole} = 0.275\,s$). The mean and maximum error of the pressure values along the centerline, relative to the pressure values of the simulation with the smallest time step, were compared for different ΔT's at the moment of peak-systole. From this, it was concluded that 360 time steps per period ($T = 0.99\,s$) would yield a sufficiently accurate solution for an acceptable computation time. This corresponds to a time step of $0.00275\,s$, what lies in the CFD challenge interquartile range of $0.17\,ms$ to $4\,ms$. Using a similar strategy, it was found that three cycles are sufficient to obtain a periodic regime. All calculations were performed in parallel on 12 cores of a Dell PowerEdge R610 server with 2 six-core Intel Xeon X5680 ($3.33\,GHz$) processors and $96\,GB$ of RAM. For the $1024k$ cells mesh, one period of the unsteady simulation with a time step size of $0.00275\,s$ took about 60 hours to finish.

This fairly long computation time was a result of slow convergence of the continuity equation, with the largest residuals of the mass imbalance located in regions where highly skewed cells were grouped together. Further investigation showed that the mean computation time per time step was highly dependent on the under-relaxation factor for the pressure, ω_p, and that values lower than the default value of 0.3 significantly decreased the computation time, with an optimum for $\omega_p = 0.261$, (Figure 3.4).

Cells	r_1	r_2	$L_{edge,min}\,[m]$	$L_{edge,mean}\,[m]$	$E_{mean}[\%]$	$E_{max}\,[\%]$
$93k$	3	3	$3.5\,e$-5	$3.9\,e$-4	0.073	0.41
$221k$	3	4	$2.5\,e$-5	$2.9\,e$-4	0.058	0.29
$432k$	3	5	$1.9\,e$-5	$2.31\,e$-4	0.06	0.26
$1024k$	5	4	$0.8\,e$-5	$1.76\,e$-4	0.052	0.20
$2000k$	5	5	$0.65\,e$-5	$1.39\,e$-4	0.045	0.21
$3456k$	6	5	$0.6\,e$-5	$1.16\,e$-4	0.017	0.082
$5488k$	7	5	$0.475\,e$-5	$0.99\,e$-4	0.010	0.054
$8912k$	8	5	$0.47\,e$-5	$0.86\,e$-4	Reference grid	

Table 3.3: Spatial resolution and corresponding errors of the grid refinement study

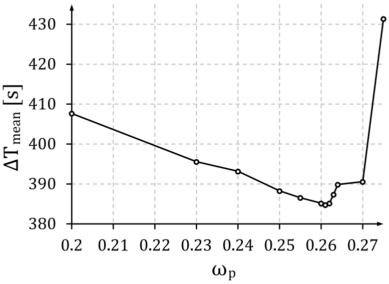
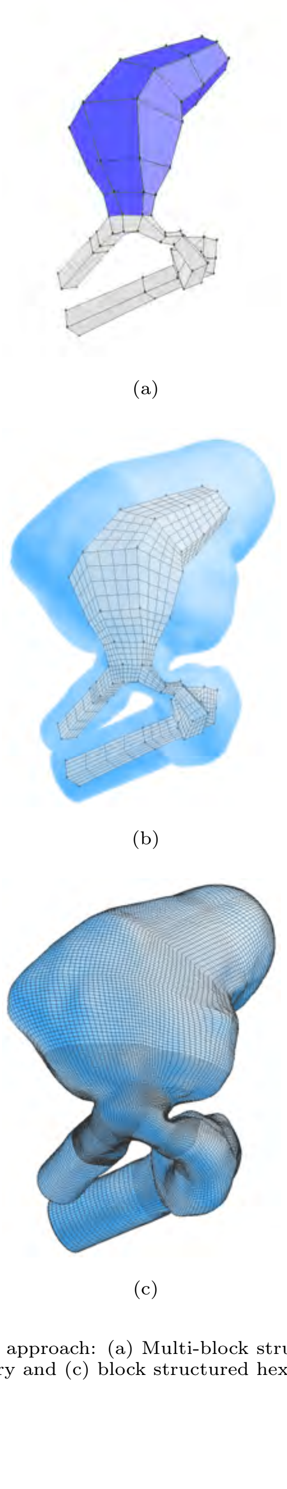


Figure 3.4: Mean computation time per time step [s] as a function of the under-relaxation factor for the pressure. Simulations were performed on an Intel Xeon X5355 (8 cores of 2.66 GHz) with first-order upwind discretization for the momentum equations.

3.3.2.4 The multi-block grid-based approach

The mapping approach, as discussed in Section 3.3.2.2, offers great flexibility regarding the applicability to complex geometries. However, a large geodesic distance between the internal block structure and the surface geometry is observed at high curvature regions – such as aneurysm necks, severe stenoses or bifurcations – with a negative effect on the surface mapping. At the expense of operator time and hex-core quality, the nodes of the multi-block structure can be adjusted, but – as pointed out in Section 3.3.2.3 – this is no assurance that the final grid is purged from all zones of low quality cells and that acceptable computation times are obtained. When performing a study that considers multiple large models with a similar branching topology and geometry, the multi-block structure corrections can be automated, as we did in Trachet et al. [75]. But even then, significant variation of the geodesic distance between the internal block structure and the surface geometry may exist, what impedes the generation of a high-quality boundary layer grid and results in low quality cells in regions with high curvatures, which are likely to be the regions of interest [12].

In an attempt to overcome these issues of high operator times and high computation times when addressing complex geometries, a robust methodology was developed in Chapter 2 to auto-generate hex-cores which resemble the surface geometry more closely in order to improve the mapping procedure and the final grid quality. The development of this multi-block grid-based (MBGB) approach – literally forming the core of the extended Treemesh method – was motivated by the idea that high-quality hexahedral blocks should be added to the multi-block structure, instead of being manipulated (and distorted) to resemble the surface geometry properly. In comparison with the multi-block structured hex-core of the

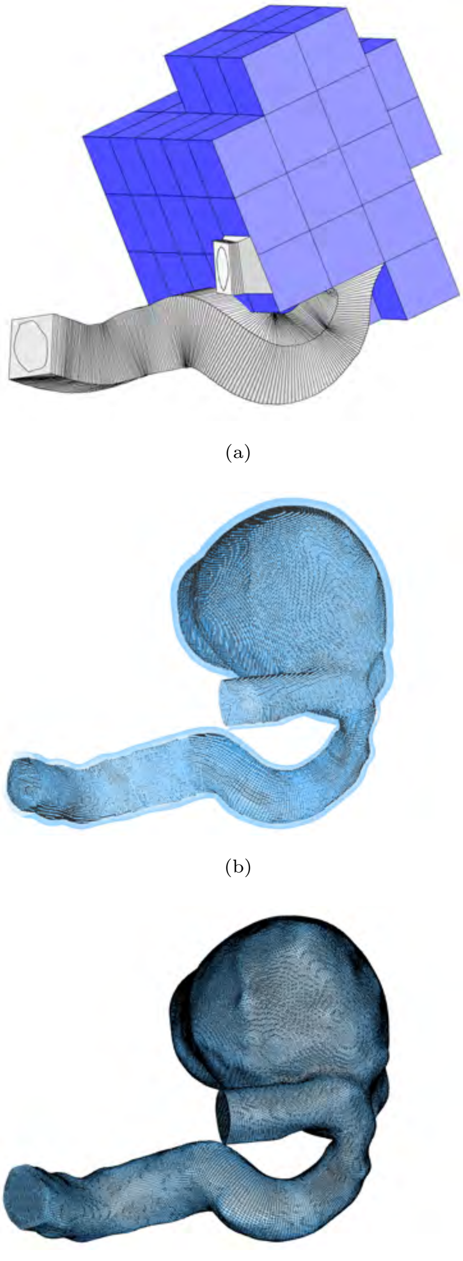


(a)

(b)

(c)

Figure 3.5: The mapping approach: (a) Multi-block structure, (b) block structured hex-core inside the geometry and (c) block structured hexahedral grid.



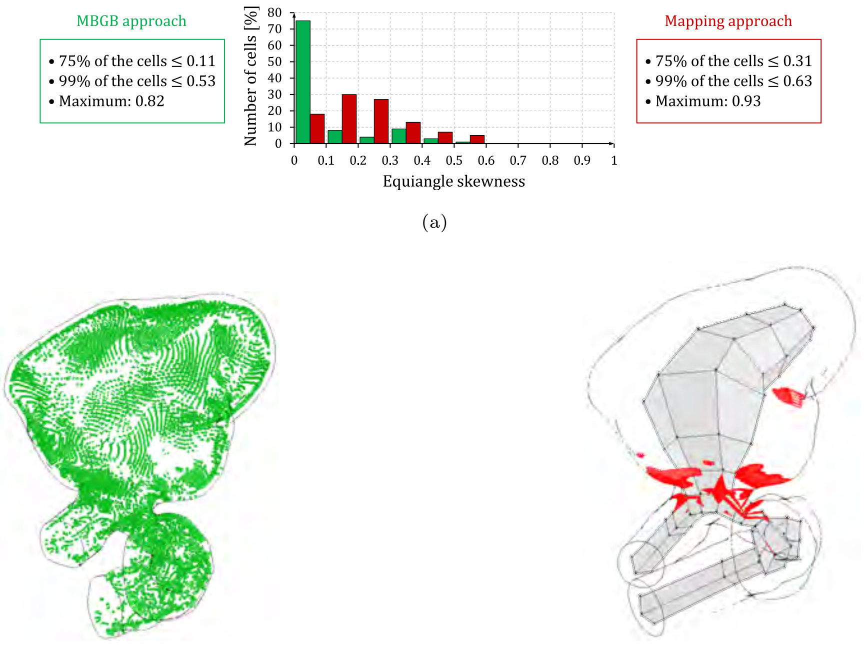
(a)

(b)

(c)

Figure 3.6: The MBGB approach: (a) Multi-block structure, (b) unstructured hex-core inside the geometry and (c) unstructured hexahedral grid.

mapping approach, the multi-block grid-based method returns an unstructured hex-core with much smaller variations in the geodesic distance between the hex-core and the surface geometry, see Figure 3.6(b). This procedure improves the quality of the body-fitted grid in general and the boundary layer grid in particular, especially at high-curvature regions. Figure 3.6 compares the multi-block grid based approach with the mapping approach of Figure 3.5 for the CFD challenge geometry. Remark that generation of the multi-block structure of Figure 3.6(a) is fully automated, without any nodal repositioning, and that the hex-core in Figure 3.6(b) is unstructured instead of multi-block structured. As the hex-core of the MBGB mesh resembles the surface geometry at the level of individual cells, a remarkable improvement of the cell quality is observed in the distribution of the equiangle skewess, Figure 3.7(a). Moreover, note that 1% of the skewest cells in the MBGB mesh are – compared to the mapping approach – located at the hex-core border, all over the fluid domain and not only at the boundary layer grid of high-curvature regions, Figure 3.7(b) versus Figure 3.7(c). To obtain a similar grid density in the parent vessel as for the mapping approach, the number of cells is increased to account for the higher grid density at the aneurysm and to account for the flow extension of the outlet in the phase II geometry. As such, the MBGB meshes consist of $1397k$ and $1841k$ cells for the phase I geometry and the phase II geometry respectively. Regarding computation time, only 6 hours and 36 minutes are required to finish one period of the unsteady phase I pulsatile 2 simulation with a time step size of $0.00275\,s$ and a default pressure under-relaxation factor of 0.3. This is significantly less than the 60 hours of computation time on the same Dell PowerEdge R610 server for the mesh with the block structured hex-core, based on the mapping mapping approach (Section 3.3.2.3).



(a)

(b)

(c)

Figure 3.7: (a) Histogram of the equiangle skewness distribution, including some facts. Spatial distribution of 1 % of the skewest cells in the meshes based on (b) the MBGB approach (with 1397k cells) and (c) the mapping approach (with 1024k cells).

3.4 Results

In this section, the simulation results for the pressure and the velocity are compared with the results obtained by the other participating groups.

3.4.1 Pressure drop and velocity along the centerline

Pressure drop along the centerline. To quantify the variability in the pressure drop of the cycle-averaged, the peak systolic and the steady state solutions, the centerline pressures are presented in Figure 3.8, and this for both the phase I and the phase II cases (Figures 3.8 (a, b) with twenty-seven contributing solutions at 207 data points and Figures 3.8 (c, d) with twenty-four contributing solutions at 261 data points respectively). The resulting pressure drops for the steady state simulations (Figures 3.8 (a, c)) are colored by flow rate: SS1 (magenta), SS2 (cyan), SS3 (red) and SS4 (green). The pressure drops for the corresponding flow rates (see Table 3.2) of the unsteady simulations (Figures 3.8 (b, d)) share the same colors: AV1 (magenta), AV2 (cyan), PK1 (red) and PK2 (green). Our simulation results regarding the solution strategy as described in Section 3.3 and the MBGB mesh are superimposed by the curve in black. In general, the various solvers predicted similar trends: a major pressure drop is found around the point of maximum stenosis, followed by a small pressure recovery distal to this stenosis and then a further, more mild, pressure drop at the aneurysm neck. The corresponding pressure drop predictions, for which the statistics are presented in Table 3.4, were remarkably consistent and typically differing by less than 10 %.

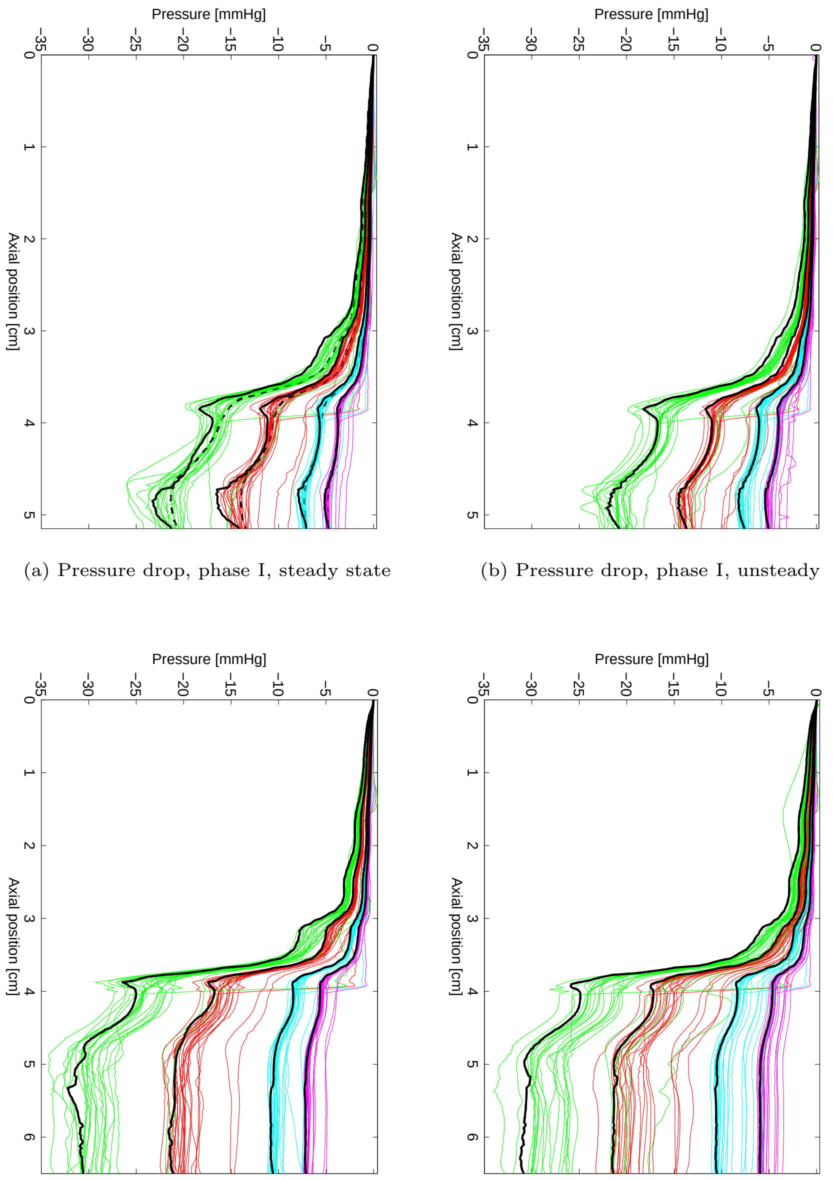
		Phase I			Phase II		
	Case	Median	(IQR)	XTM	Median	(IQR)	XTM
Unsteady	AV1	5.0	(4.8–5.1)	5.0	5.6	(5.2–5.7)	5.9
	AV2	7.6	(7.2–7.8)	7.6	10.0	(9.4–10.2)	10.5
	PK1	13.7	(13.0–13.9)	13.7	19.3	(17.7–20.3)	21.4
	PK2	20.5	(19.4–21.0)	20.7	27.8	(25.6–28.8)	30.9
Steady state	SS1	4.8	(4.7–4.9)	4.7	6.7	(6.5–6.9)	7.1
	SS2	7.2	(7.0–7.3)	7.1	10.0	(9.7–10.1)	10.6
	SS3	13.9	(13.4–14.4)	14.2	19.5	(18.6–20.3)	21.1
	SS4	20.9	(20.2–21.8)	21.3	29.1	(27.7–30.0)	30.6

Table 3.4: Descriptive statistics for the inlet-outlet pressure drops (in $[mmHg]$), comparing the median and interquartile range (IQR) with our solution, based on the extended Treemesh method (XTM).

For the phase I results, our solutions are very close to the median solution: within 1 % for the unsteady cases which show an IQR of 6 to 8 % of the median value, and within 2 % for the steady cases which show an interquartile range (IQR) of 4 to 7 % of the median value. Despite the similar trend for pressure drop along the centerline, the degree of pressure recovery distal to the stenosis was variable, with many solutions predicting no pressure recovery at all. The latter might be affected by a lack of accuracy in the pressure drop solution as we did not find a pressure recovery either when discretizing the momentum equations with a first-order upwind scheme instead, as shown by the dashed curves in Figure 3.8(a).

For the phase II results a similar trend is found as for the phase I simulations, but with a total pressure drop which is larger. These phase II results were obtained for similar flow rates, but for a slightly different geometry (including a smaller cross sectional area at the stenosis) and a larger density of the fluid. The variability in the phase II pressure drops is larger than for the phase I pressure drops, with our solution differing less than 11% from the median value. Note again that not all phase II solutions predict the pressure recovery distal to the stenosis. Remark that the difference between the cycle-averaged and peak systolic pressure drops and the corresponding pressure drops for the steady state simulations is negligible for both the phase I and phase II simulations, suggesting that a quasi-steady assumption would be reasonable.

Velocity magnitude along the centerline. To quantify the variability in the velocity of the cycle-averaged, the peak systolic and the steady state solutions, the centerline velocity magnitudes are presented in Figure 3.9, and this again for both the phase I and the phase II cases (Figures 3.9 (a, b) and Figures 3.9 (c, d) respectively). To represent the different cases, an identical coloring by flow rate is applied as for the pressure drops in Figure 3.8, with SS1 (magenta), SS2 (cyan), SS3 (red) and SS4 (green) for the steady state simulations (Figures 3.9 (a, c)) and AV1 (magenta), AV2 (cyan), PK1 (red) and PK2 (green) for unsteady simulations (Figures 3.9 (b, d)). In the different cases, a similar trend is observed, however, with a fairly large variation in maximum velocity at the throat section of the stenosis (axial position = $3.8\,cm$) and near the in- and outlet. At the inlet (axial position = $0\,cm$) the effect of the choice of inlet boundary condition is clearly visible as distinct solution groups can be distinguished. For the plug profile inlet boundary condition for example, the initially coincident solutions are shown in the inset figure of Figure 3.9(a).



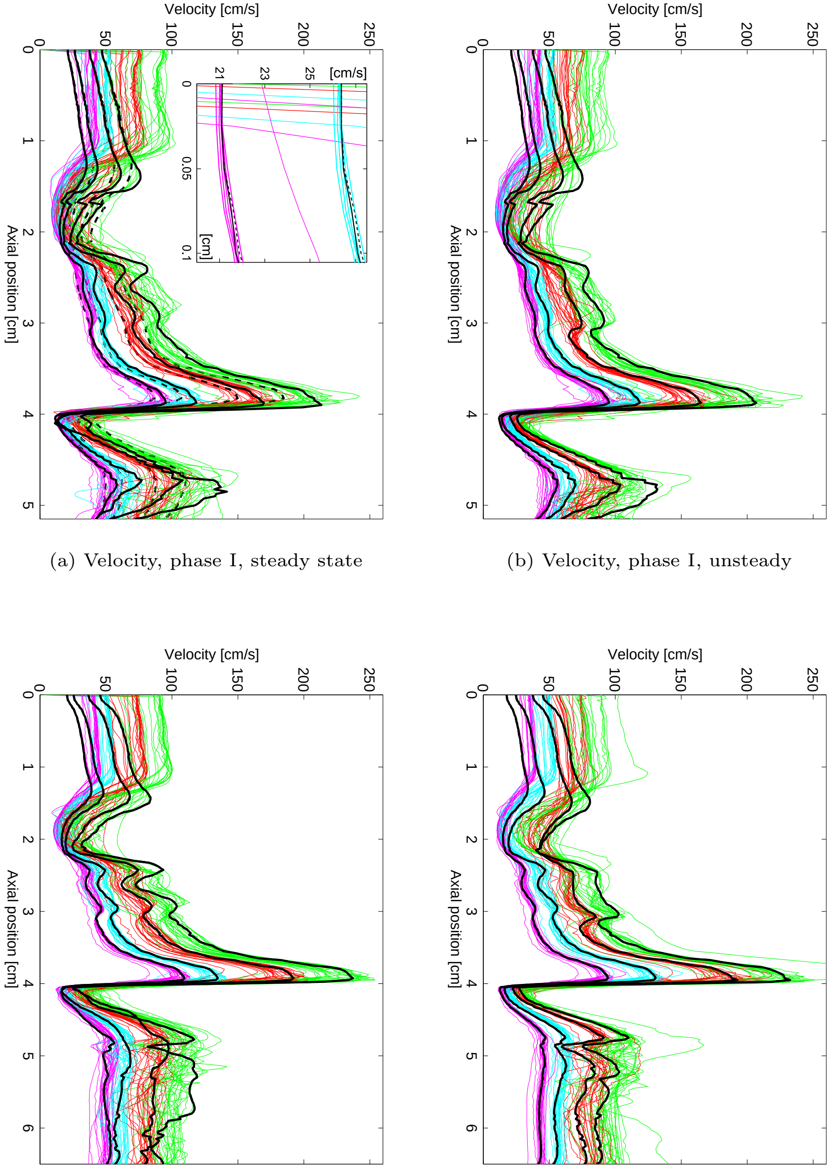
(a) Pressure drop, phase I, steady state

(b) Pressure drop, phase I, unsteady

(c) Pressure drop, phase II, steady state

(d) Pressure drop, phase II, unsteady

Figure 3.8: Benchmark results for the pressure drop along the centerline, with our solution superimposed in black. (a, b) phase I and (c, d) phase II results for both the steady state simulations (a, c) and unsteady simulations (b, d). The curves are colored by flow rate (see Table 3.2): AV1, SS1 (magenta); AV2, SS2 (cyan); PK1, SS3 (red); PK2, SS4 (green).



(a) Velocity, phase I, steady state

(b) Velocity, phase I, unsteady

(c) Velocity, phase II, steady state

(d) Velocity, phase II, unsteady

Figure 3.9: Benchmark results for the velocity magnitude along the centerline, with our solution superimposed in black. (a, b) phase I and (c, d) phase II results for both the steady state simulations (a, c) and unsteady simulations (b, d). The curves are colored by flow rate (see Table 3.2): AV1, SS1 (magenta); AV2, SS2 (cyan); PK1, SS3 (red); PK2, SS4 (green).

3.4.2 Pressure and velocity magnitude versus time

For quantification of the variability in the solutions regarding pressure versus time and velocity magnitude versus time, the pressure and velocity waveforms are presented in Figure 3.10 and Figure 3.11 respectively, for one cardiac cycle by means of boxplots. Each boxplot presents the data at different locations in the fluid domain. The pressure data is presented at four points as a pressure drop relative to the outlet pressure: inlet (red), stenosis (green), aneurysm neck (cyan) and aneurysm sac (magenta); whereas the velocity data is presented for the outlet (blue) as well. Our solution is resampled and presented by a solid line, which is superimposed on the boxplots. These boxplots show *i*) the median values as black dots, *ii*) the 25^{th} to 75^{th} percentile range or IQR by a vertical bar, and *iii*) the range up to the maximum and minimum value by vertical lines. These lines are defined to be at maximum 1.5 times the length of the IQR-bar; any values exceeding this limit are defined as outliers which are represented by circular symbols.

Although the variability is found to be larger in the velocity data than in the pressure data, both Figure 3.10 and Figure 3.11 show an increase in variability when the pressure and velocity magnitude increases. This can be seen in every boxplot by comparing pressures and velocities at different locations (e.g. aneurysm sac pressure versus inlet pressure, aneurysm sac velocity versus stenosis velocity) and by comparing corresponding curves among the different boxplots (pulsatile 1 versus pulsatile 2 and phase I versus phase II). The boxplots of the inlet pressures (red) and the velocities at the stenosis (green) reveal that our unsteady phase I solutions are very close to the median solutions throughout the complete cardiac cycle. Our unsteady phase II solutions on the other hand, predict pressures and velocity magnitudes within the IQR, except for the velocity magnitudes at the inlet (the red curves in Figure 3.11) where our solution clearly underestimates the global results. This can be explained by the fact that the inlet boundary condition strongly influences the inlet velocity, and that the applied plug profile results in a lower inlet velocity magnitude at the center of the inlet when compared to fully developed inlet profiles. As a result, our solution is not the only one predicting this fairly low inlet velocity, as can be seen from the inset figure in Figure 3.9(a) and from the fact that the corresponding IQR-bars are relatively large. Finally, note that our simulated total pressure drops are very close to the median predicted values for both phase I and phase II, they are represented by the inlet pressure curves (relative to an outlet pressure of zero $mmHg$).

For the comparison with the experimentally-measured pressure drops, the reader is referred to the discussion section of the co-authored article by Steinman et al. [68].

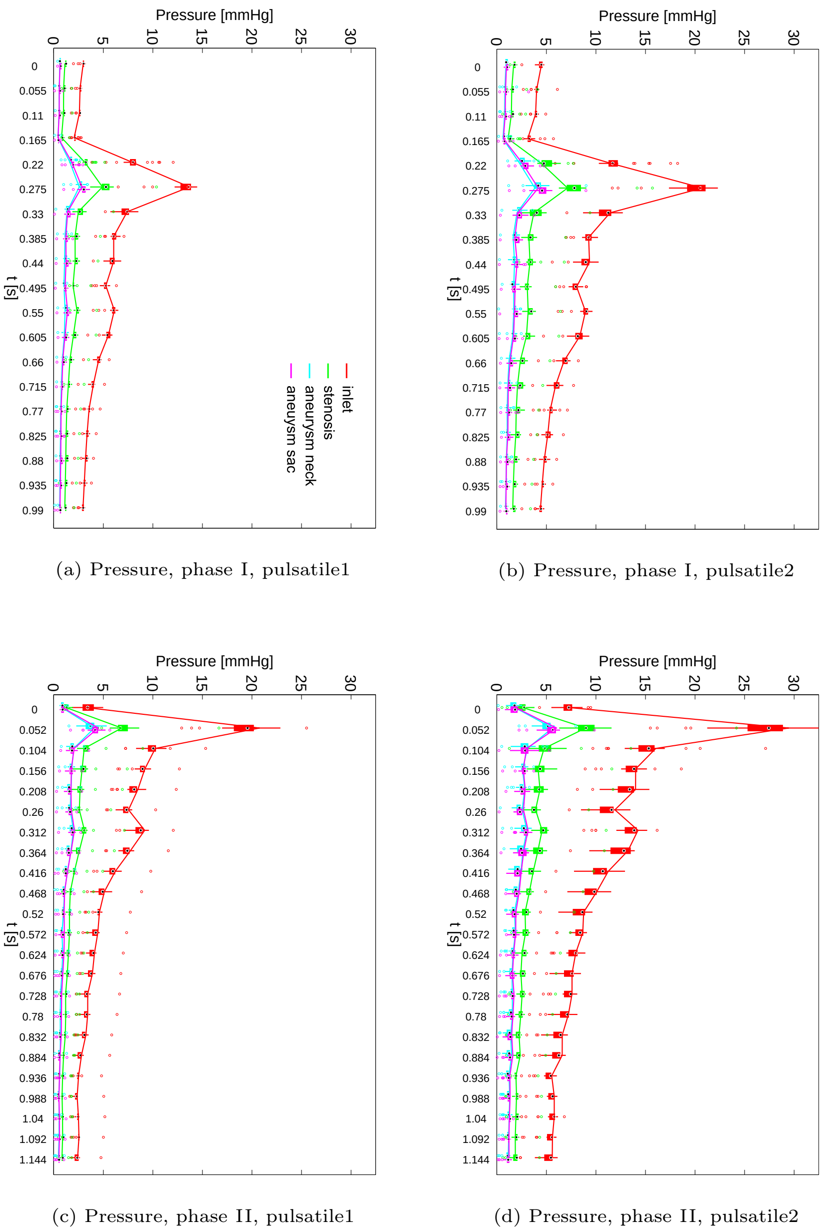
3.4.3 Variability in the surface pressure

For visualization of the cycle-averaged and the peak systolic pressures of both the phase I and the phase II solutions, the surface pressures are interpolated on a predefined reference surface mesh. To ensure intersection with all (differently discretized) CFD models, the surface nodes were translated along their inward pointing normals by $0.05\,mm$ with respect to the original surface mesh. As most variability among the pressures was found for the pulsatile 2 cases (see Figure 3.10), AV2 and PK2 surface pressures are visualized for both phase I and phase II. The cycle-averaged surface pressures (AV2) are shown in Figure 3.12 (top: phase I, bottom: phase II) for an inlet pressure set to $90\,mmHg$. The peak systolic surface pressures (PK2) are shown in Figure 3.13 (top: phase I, bottom: phase II) for an inlet pressure set at $120\,mmHg$.

Despite some outliers, the pressure distributions shows consistent results with an overall low variability for all cases. Our solution (l) is denote by the red box in each case and follows the majority of the solutions with respect to the magnitude of the pressure and its spatial distribution. Regardless of the absolute pressure levels, the cycle-averaged (AV2) and the peak systolic (PK2) results show similar spatial patters for the surface pressure. Compared to the phase I results, the phase II results show (again) somewhat more variability across the solutions, with lower aneurysm sac pressures confirming the stronger pressure drop across the stenosis as predicted by Figure 3.8. It is worth noting that the outliers in phase I are not necessarily the outliers in phase II, what might indicate issues in the post processing phase which is however an inextricably part of the CFD analysis.

3.4.4 Variability in the velocity field

For visualization of the cycle-averaged and the peak systolic flow patterns, the phase I velocity magnitudes are interpolated on a predefined reference volume mesh and presented by means of a isosurfaces. As most variability among the velocities was found for the pulsatile 2 cases (see Figure 3.11), phase I isosurfaces are visualized for both AV2 and PK2. The cycle-averaged (AV2) isosurfaces are shown in Figure 3.14 (top) for a velocity magnitude of $30\,\frac{cm}{s}$. The peak systolic (PK2) isosurfaces are shown in Figure 3.14 (bottom) for a velocity magnitude of $50\,\frac{cm}{s}$.



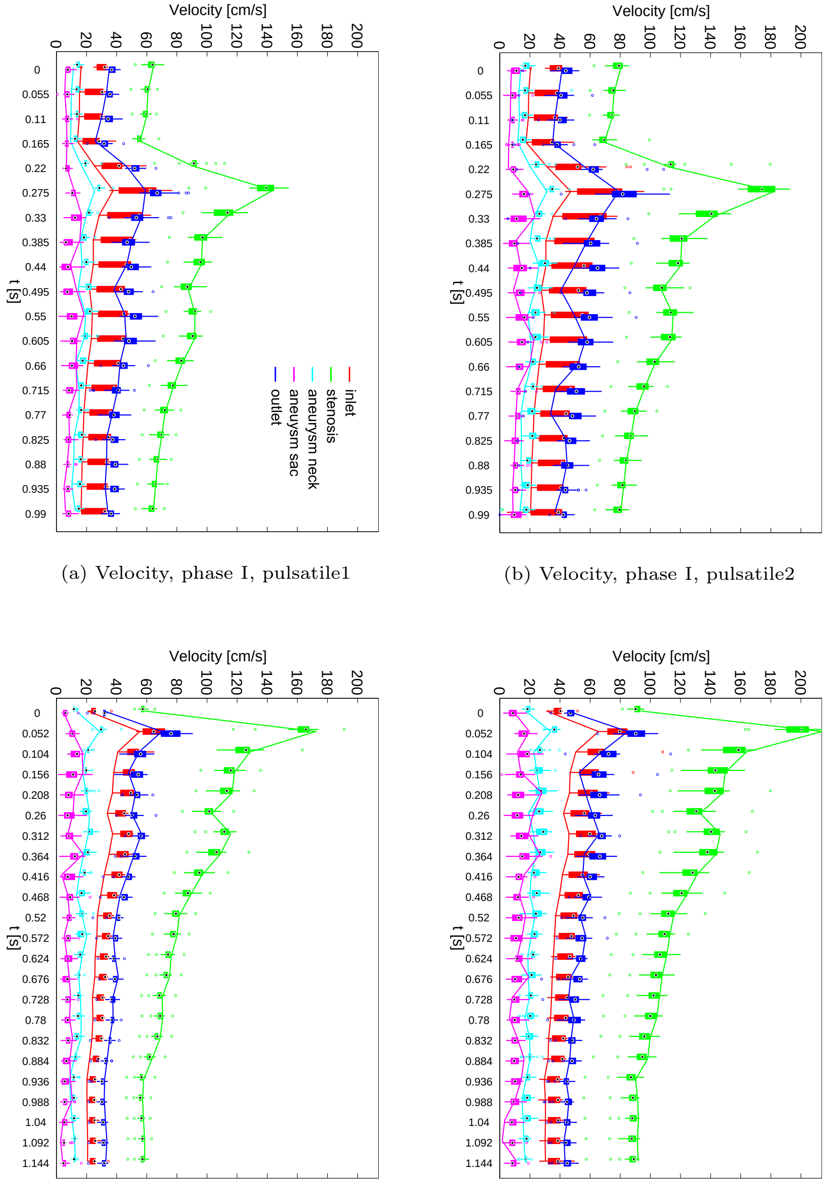
(a) Pressure, phase I, pulsatile1

(b) Pressure, phase I, pulsatile2

(c) Pressure, phase II, pulsatile1

(d) Pressure, phase II, pulsatile2

Figure 3.10: Benchmark boxplots (with the outliers represented by dots) for the pressure as a function of time at four fixed locations, relative to the outlet pressure and with our solution superimposed. (a, b) phase I and (c, d) phase II results for both the pulsatile 1 cases (a, c) and the pulsatile 2 (b, d) cases. The boxplots (minimum, 25th percentile, median, 75th percentile, maximum and outliers) are colored by location: inlet (red), stenosis (green), aneurysm neck (cyan), aneuysm sac (magenta).



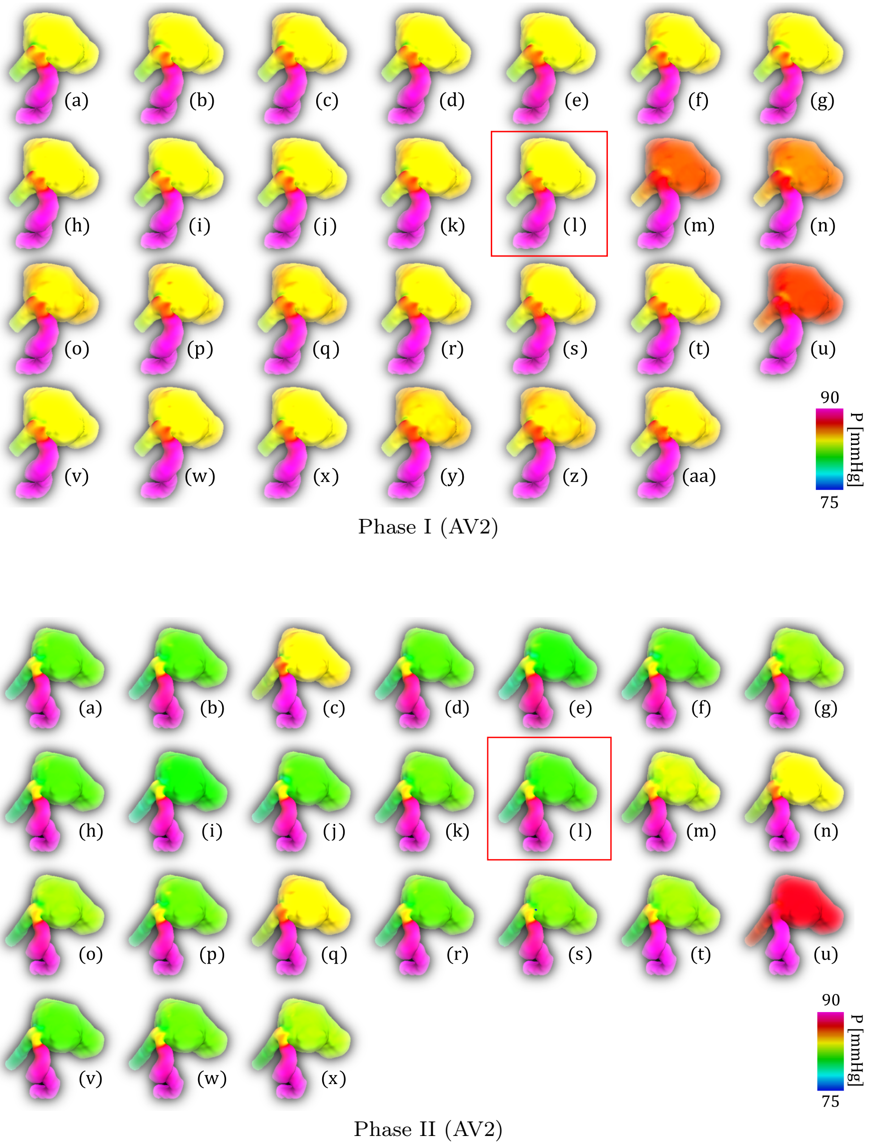
(a) Velocity, phase I, pulsatile1

(b) Velocity, phase I, pulsatile2

(c) Velocity, phase II, pulsatile1

(d) Velocity, phase II, pulsatile2

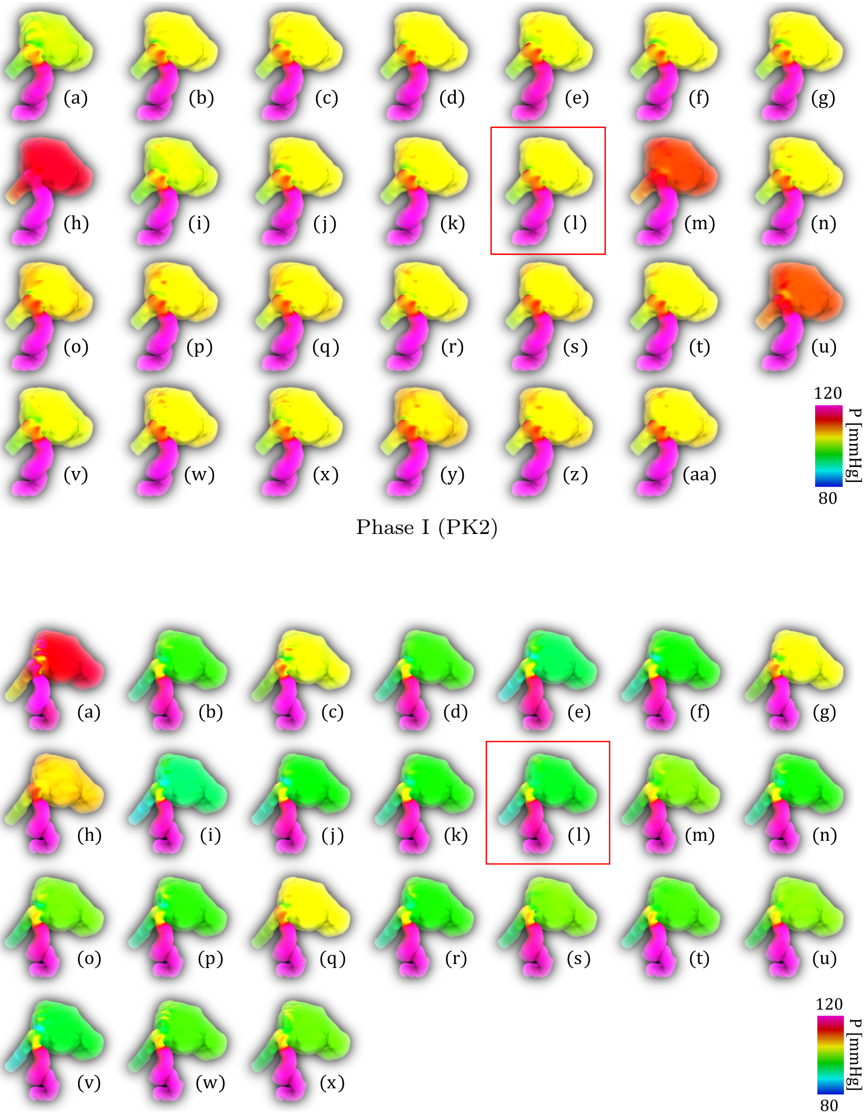
Figure 3.11: Benchmark boxplots (with the outliers represented by dots) for the velocity as a function of time at five fixed locations with our solution superimposed. (a, b) phase I and (c, d) phase II results for both the pulsatile 1 cases (a, c) and the pulsatile 2 (b, d) cases. The boxplots (minimum, 25^{th} percentile, median, 75^{th} percentile, maximum and outliers) are colored by location: inlet (red), stenosis (green), aneurysm neck (cyan), aneuysm sac (magenta), outlet (blue).



Phase I (AV2)

Phase II (AV2)

Figure 3.12: Cycle-averaged surface pressures for the pulsatile 2 case (AV2), for both phase I (top) and phase II (bottom). Inlet pressure is set to 90 $mmHg$. Figure modified from [68]. Our solution (l) is denoted by the red box.

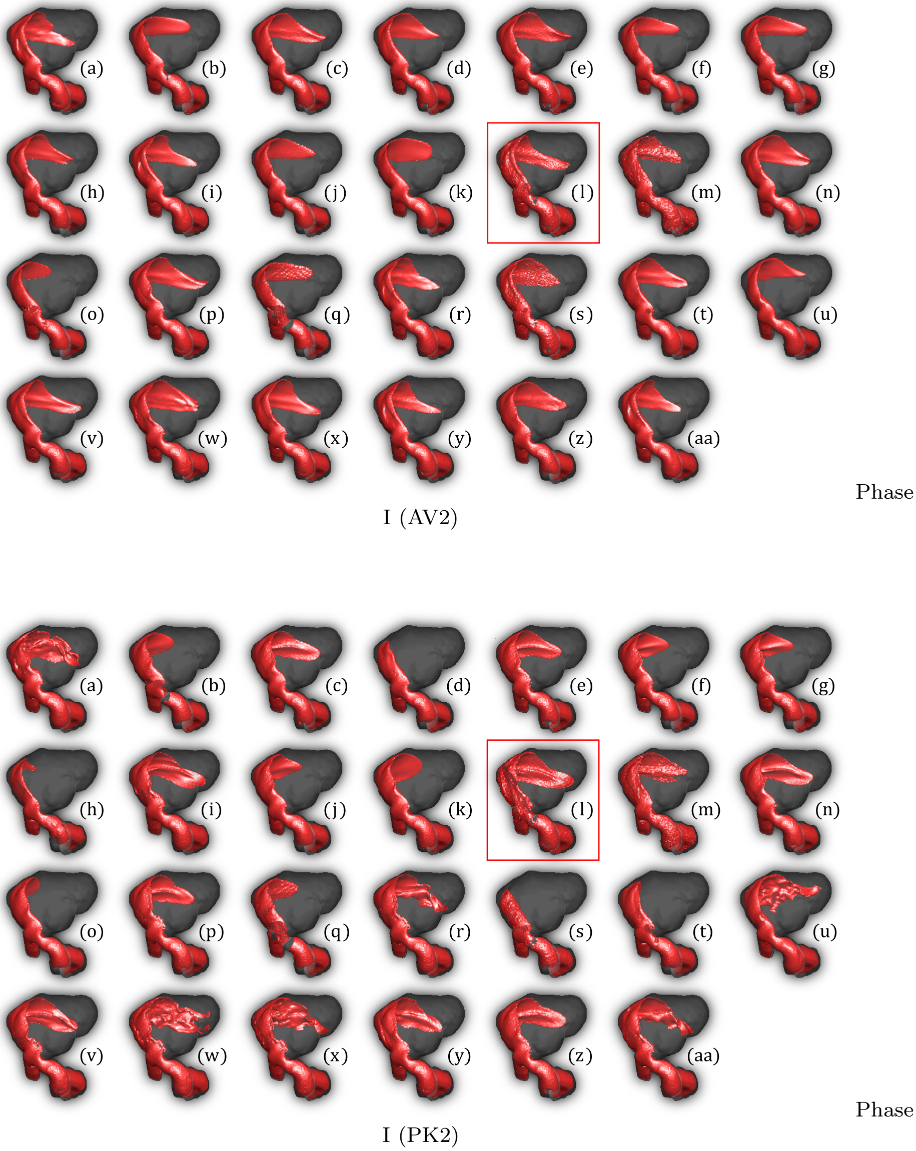


Phase I (PK2)

Phase II (PK2)

Figure 3.13: Peak systolic surface pressures for the pulsatile 2 case (PK2), for both phase I (top) and phase II (bottom). Inlet pressure is set to 120 $mmHg$. Figure modified from [68]. Our solution (l) is denoted by the red box.

Compared to the spatial pressure distributions, the isosurfaces of the velocity show more variability among the solutions. This variability is most clear upon inspection of the inflow pattern into the aneurysm sac, with some of the solutions predicting an inflow which is in agreement with our findings (l), indicated by the red box, and other solutions predicting a lower inflow. Note that a comparison of the overall PK2 solutions to the AV2 solutions reveals that a high mass flow rate does not necessarily induce a high inflow into the aneurysm sac.



Phase

I (AV2)

Phase

I (PK2)

Figure 3.14: Isosurfaces of the phase I velocity field. (a) Cycle-averaged (AV2) isosurfaces for a velocity magnitude of 30 $\frac{cm}{s}$. (b) Peak systolic (PK2) isosurfaces for a velocity magnitude of 50 $\frac{cm}{s}$. Figure modified from [68]. Our solution (l) is denoted by the red box.

3.5 Conclusion

By participating in a blinded CFD variability study, we were able to compare our solutions to those obtained by other experts in this field. This allowed us to validate our solution strategy as a whole and our meshing procedure in particular, without having prior knowledge of the final outcome of the study influencing our approach.

Regarding the meshing strategy, the extended Treemesh method of Chapter 2 was applied to spatially discretize the lumen of a giant carotid aneurysm. Apart from a local increase in the grid density at the stenosis, the generated grid features hexahedral cells which are aligned with the predominant parent vessel flow. As shown in Section 3.3.2.1, this improves the accuracy of the solution by reducing the truncation error which results from discretizing the convective term. Compared to a mapping approach, it was shown that a multi-block grid-based mesh is able to significantly improve the cell quality, especially in high-curvature regions near the wall, and this without any additional manual – and operator time consuming – interventions. As such, the multi-block grid-based approach made the simulations less dependent of the pressure under-relaxation factor and resulted in a speed-up by a factor of 10.

Regarding validation, the pressure and velocity along the centerline (Section 3.4.1), the pressure and velocity in discrete centerline points over time (Section 3.4.2), the surface pressure (Section 3.4.3) and the velocity field (Section 3.4.4) were analyzed and following conclusions were drawn:

- *Regarding the pressure.* Although a wide variety of solvers and solution strategies were employed, remarkably consistent solutions were found for the total pressure drops, with a quasi-steady assumption having a negligible impact. Our predictions for the pressure drops were very close to the median values, especially for cases with an overall low variability (e.g. within 2 % for the phase I results). This variability was higher for the phase II results which showed larger pressure drops (and higher velocities) at the stenosis region. Further inspection of the pressure drop along the centerline revealed different degrees of pressure recovery distal to the stenosis, with many solutions predicting no pressure recovery at all. This might be the consequence of reduced accuracy just distal to the stenosis, as pressure recovery was abscent in our solution too when modifying our solution strategy by applying a less accurate first-order upwind discretization scheme for the momentum equations. Besides some outliers, low variability was found regarding the pressure evolution over time and the variability regarding the CFD-predicted pressure patterns, with our solutions predicting pressures close to the median results.

- *Regarding the velocity.* The variability in the velocities along the centerline was found to be substantially higher than the variability in the pressures, especially at the inlet where the effect of the inlet boundary conditions was clearly visible. At the stenosis, where the velocity reaches its maximum, our phase I solutions are very close to the median predictions, whereas our results for the phase II velocities slightly overestimate the median solutions. High variability was found among the flow patters too, with multiple groups predicting an inflow into the aneurysm sac which was comparable to our solution.

Based on these results, further use of the presented solution strategy and meshing procedure is justified for the vascular CFD models in Chapters 4 and 8.

Chapter 4

Application of the extended Treemesh method to an FSI-model of aortic coarctation [1]

4.1 Introduction

In previous chapters, the robustness of the meshing approach was tested on common complex geometries, such as treelike structures including trifurcations, aneurysms and a mild stenosis. To complete the picture with a severe stenosis, this chapter presents a grid sensitivity analysis of a fluid-structure interaction simulation performed in a model of aortic coarctation. In this application, the true potential of the extended Treemesh method is shown by obtaining anisotropically refined grids which lower the computation time while maintaining the same level of accuracy. This sensitivity analysis was carried out as part of a more general study performed by Taelman et al., which sought to assess insights into the effect of local stiffening and narrowing on the central hemodynamics seen in repaired aortic coarctation [71]. In this example the pressure drop along an aorta with aortic coarctation is analyzed. Aortic coarctation (CoA) is a congenital disease characterized by a narrowing of the upper descending aorta. The pressure difference between the ascending and descending aorta is the most important indication of

[1] This chapter is based on two articles *i*) *"Unstructured hexahedral mesh generation of complex vascular trees using a multi-block grid-based approach."*, as submitted to the Journal *Computer Methods in Biomechanics and Biomedical engineering* and *ii*) *"Insights into the effect of a local stiffness and narrowing on central hemodynamics in repaired aortic coarctation."*, as submitted by Taelman et al. to the *Annals of Biomedical Engineering.*

aortic coarctation. As it is difficult to assess in a non-invasive way, the simulation of this pressure drop using computational techniques is of high clinical relevance. Moreover, the narrowing leads to an impedance mismatch and is a source of wave reflections. As simulations with rigid walls fail to capture some physiological patterns (such as wave reflection), the fluid-structure interaction (FSI) between the blood flow and the deformation of the arterial wall was taken into account. As FSI simulations are computationally expensive and time consuming, it is important to use a computational grid for which accurate results are obtained with a limited number of cells.

4.2 Materials and methods

The geometrical model was obtained by MRI images of a healthy 39 year old male volunteer. The functional impact of aortic coarctation was modeled by including a stenosed region, indicated by the red zone in Figure 4.3, with a length of 1 *cm* (L) and a coarctation index (CI) of 0.5. The coarctation index is defined by the ratio of the diameter at the coarctation zone ($\mathrm{D_{coa}}$) to the diameter of the descending aorta ($\mathrm{D_{desc}}$).

Unstructured hexahedral grids – with and without local grid refinements – were generated using the extended Treemesh method of Chapter 2. Multiple grids were generated for a constant square scaling factor f_s of 1.1 and a radial compression ratio f_R of 0.8 by gradually decreasing the distance between the centerline nodes d_0 and increasing the amount of cells throughout the boundary layer, defined by f_{BL}. The auto-generated multi-block structure for the finest grid is shown in Figure 4.1(a). In order to include a local refinement at the coarcation zone and distal to the stenosis, another multi-block structure was constructed by locally adapting the square scaling factors $\boldsymbol{f}_s$, as illustrated in Figure 4.1(b), and by modifying the axial edge refinement factors $\boldsymbol{f}_{edge,a}$ on a local base, as presented in Section 2.4.2. The multi-block structures were used as input for the multi-block grid-based approach of Section 2.2.2. As the FSI coupling code allows a non-conformal mesh at the fluid-structure interface, the interface mesh was optimized according to the procedure in Section 2.2.4 and Figure 2.12, by iteratively removing all ‘diamond shaped’ faces as depicted in Figure 4.2. This improves the quality of the mesh for the arterial wall which was generated by radially expanding the quadrilateral interface into a hexahedral mesh with multiple layers and a diameter to thickness ratio of 10 %.

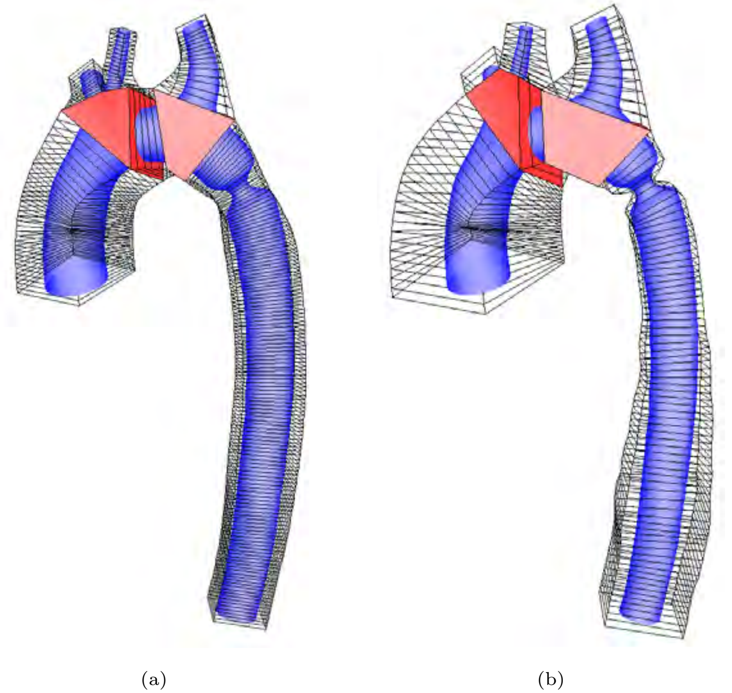


Figure 4.1: Multi-block structures enclosing the radially compressed surface geometry for construction of (a) the uniform grid R4 and (b) the locally refined grid R5, see Table 4.1.

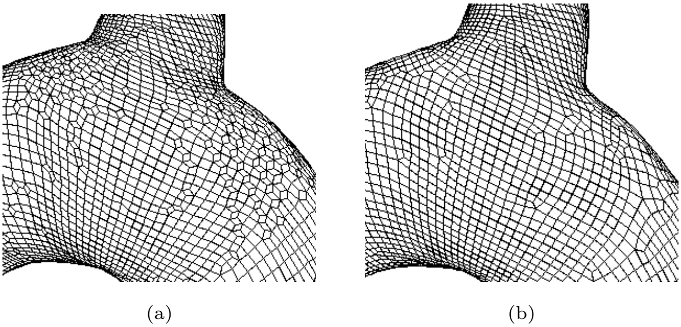


Figure 4.2: Interface mesh at the bifurcation of the left subclavian artery before (a) and after (b) improving the cell quality by removing redundant 'diamond shaped' faces.

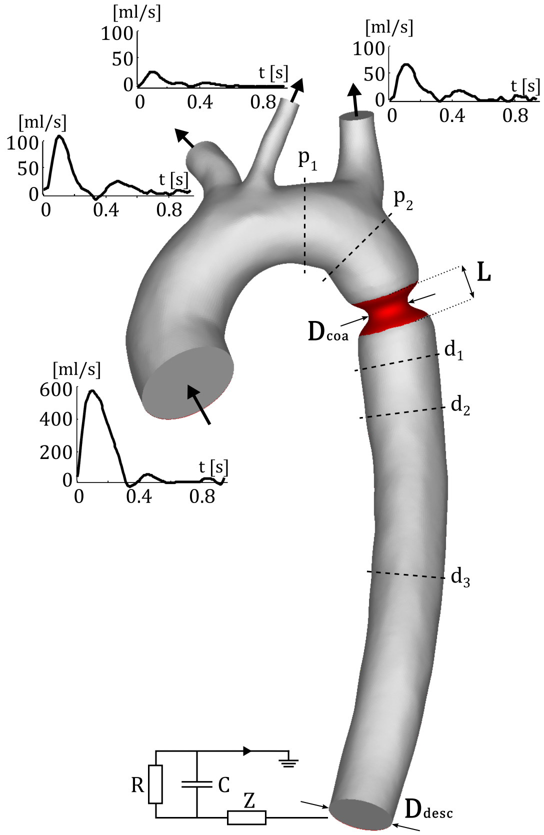


Figure 4.3: Geometrical model of the aortic arch with aortic coarctation (red zone), including the boundary conditions for the fluid domain and the cross sectional regions (p1, p2, coa, d1, d2, d3, desc) in which the convergence was analyzed numerically (see Table 4.1).

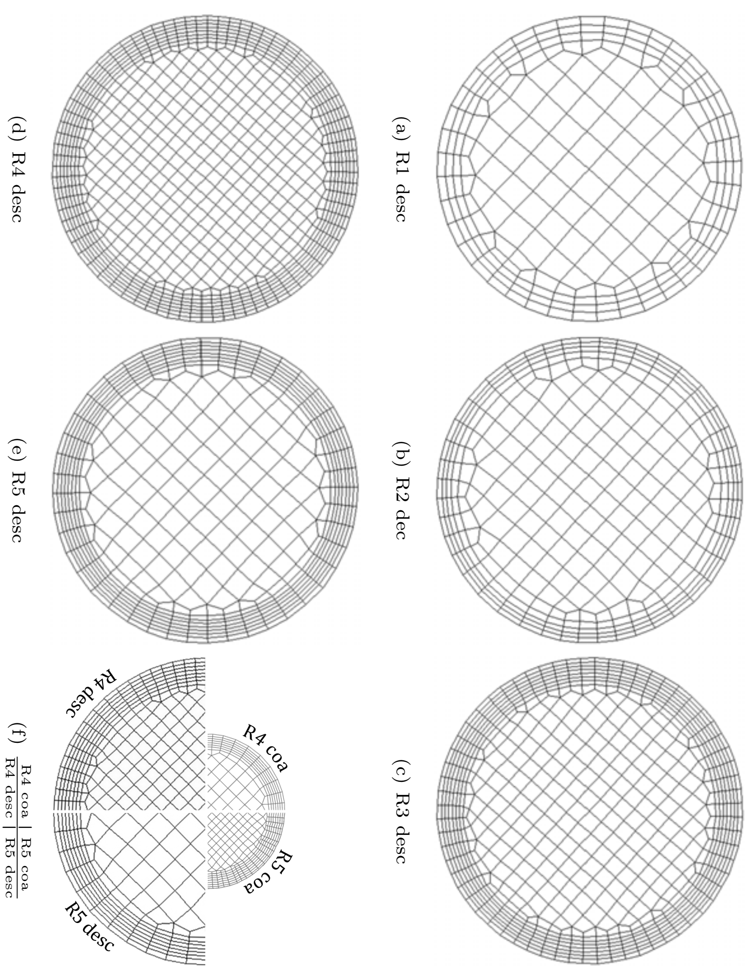
Flow rates were measured with phase-contrast MRI at the ascending aorta and the three side branches (the brachiocephalic artery, the left common carotid artery and the left subclavian artery) and imposed as physiological boundary conditions, see Figure 4.3. At the descending aorta, a three-element windkessel model was implemented ($Z = 0.08\,\frac{mmHg}{ml/s}$, $R = 1.024\,\frac{mmHg}{ml/s}$, $C = 2.0\,\frac{ml}{mmHg}$), in which the parameters of the model were defined such that physiological pressure variations were retrieved. Blood was modeled as a Newtonian fluid (viscosity $3\,mPas$, density $1050\,\frac{kg}{m^3}$). At the in- and outlet boundaries of the solid domain, only radial displacement was allowed and the material behavior of the aortic tissue was described using a polynomial hyperelastic model (hyperelastic constants: $C_{10} = 18.9\,kPa$, $C_{01} = 2.75\,kPa$, $C_{20} = 400\,kPa$, $C_{11} = 847.2\,kPa$).

To obtain an accurate calculation of the stress on the fluid-structure interface, the flow equations are solved in the Arbitrary Lagrangian-Eulerian (ALE) formulation on a deforming mesh, using Fluent (Ansys). The structural equations are solved in a Lagrangian frame, using Abaqus/Standard (Simulia). The solvers were strongly coupled with the IQN-ILS technique [19].

4.3 Results

Four different meshes were constructed with an increasing cell density: R1, R2, R3 and R4. The grids were generated in about 30 minutes by using our in-house developed code with dedicated (fifteen-step) software wizard. The first 15 minutes were required to visually inspect the consecutive steps and generate the first mesh, and 5 more minutes were required to adapt the refinement parameters and generate each of the other three grids. The cross sectional views of these grids at the location of the descending aorta are presented in Figures 4.4 (a) to (d). The corresponding number of cells are depicted in Table 4.1, together with the calculation time required to compute one cardiac cycle on two 10-core Intel Xeon E5-2680v2 processors.

As FSI simulations are computationally expensive and time consuming, it is important to use a computational grid for which accurate results are obtained with a limited number of cells. To illustrate how the proposed meshing strategy is able to provide an adaptively refined grid, a fifth mesh is constructed (R5) of which the cross sectional view at the location of the descending aorta is presented Figure 4.4(e). By locally adapting the square scaling factors, the mesh density is modifiable both radially and circumferentially. By adapting the axial edge refinement factors on a local level, the mesh density is modifiable in axial direction.


(a) R1 desc

(b) R2 dec

(c) R3 desc

(d) R4 desc

(e) R5 desc

(f) $\frac{\text{R4 coa} \mid \text{R5 coa}}{\text{R4 desc} \mid \text{R5 desc}}$

Figure 4.4: (a)-(e) Cross sectional view of the grids of the fluid domain (R1 to R5) at the descending aorta and (f) comparison with the cross sectional views of the R4 and R5 grids at the coarctation (CI = $\frac{\mathrm{D_{coa}}}{\mathrm{D_{desc}}}$ = 0.5).

Grid	# Fluid cells	# Solid elem.	Calc. time per cycle	Mean error [%]							
				asc	p_1	p_2	coa	d_1	d_2	d_3	desc
R1	42k	21k	12 h 29 min	1.44	1.42	1.42	1.72	1.61	1.56	1.33	0.77
R2	105k	37k	17 h 29 min	1.01	0.98	0.98	1.26	1.54	1.88	1.26	0.71
R3	281k	74k	30 h 55 min	0.38	0.34	0.34	0.75	1.22	1.07	0.94	0.64
R4	408k	102k	40 h 38 min				Reference grid				
R5	216k	51k	23 h 38 min	0.5	0.47	0.47	0.86	1	1.15	0.98	0.6

Table 4.1: Grid refinement study of the pressure in an FSI model of aortic coarctation. Simulations were performed on a Dell PowerEdge R620 server (2× Intel Xeon E5-2680v2 CPUs at 2.8Ghz) with one core used by the coupling code and eighteen cores used by the flow solver and the structural solver.

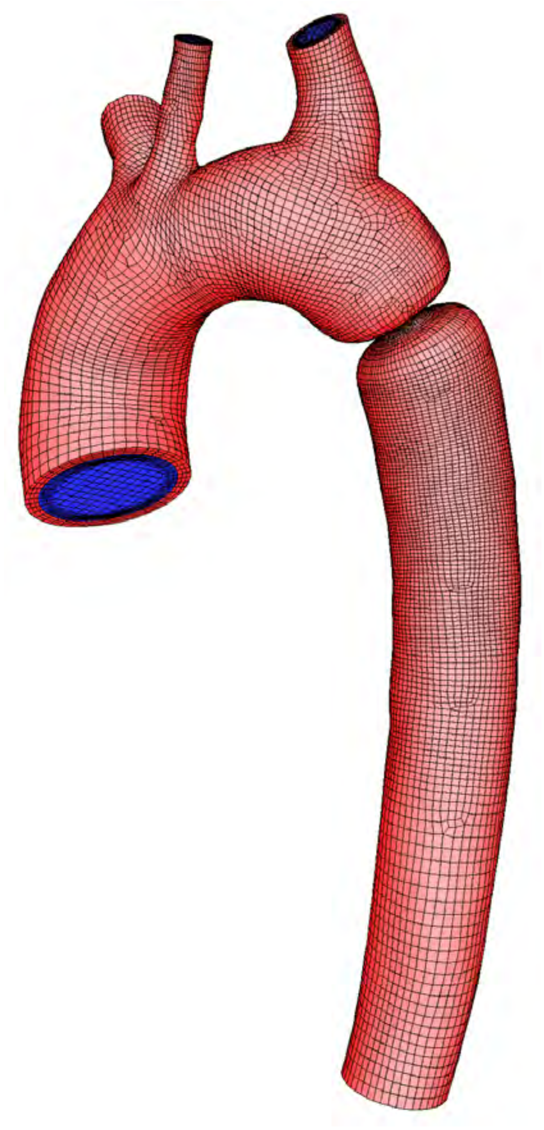


Figure 4.5: Mesh for the fluid domain (blue) and mesh for the arterial wall (red) of an aortic arch with aortic coarctation.

As such, the coarctation region was refined and the ascending and descending aorta were coarsened with a gradual transition between the stenosis and the distal descending aorta. The resulting mesh is presented in Figure 4.5 and counts $216k$ cells in the fluid domain and $51k$ cells in the solid domain. Compared to the finest mesh (R4), the locally refined grid (R5) features a higher mesh density in the coarctation zone, but a coarser grid proximal to the stenosis and in the lower part of the descending aorta, see Figure 4.4(f).

The results of the grid refinement study are shown in Figure 4.6, depicting the pressure evolution along the centerline of the aorta at peak systole, and in Table 4.1, showing the mean error of the pressure evolution in different cross sections along the aorta. These errors are defined with respect to the reference grid R4 and relative to the pressure amplitude in the corresponding cross section. From the results in Table 4.1, it can be seen that even for the meshes with a low cell density, the mean errors proximal, halfway and distal to the coarctation zone remain low ($< 2\,\%$). By increasing the cell density, this error is further reduced. When comparing the locally refined grid R5 with the uniformly refined grids R3 and R4, an important reduction in computation time is gained (23 h 38 min per cardiac cycle versus 30 h 33 min and 40 h 38 min) without a loss in accuracy. The mean error obtained with the mesh R5 stays below 1.15 % and comparable errors are found as for the mesh R3.

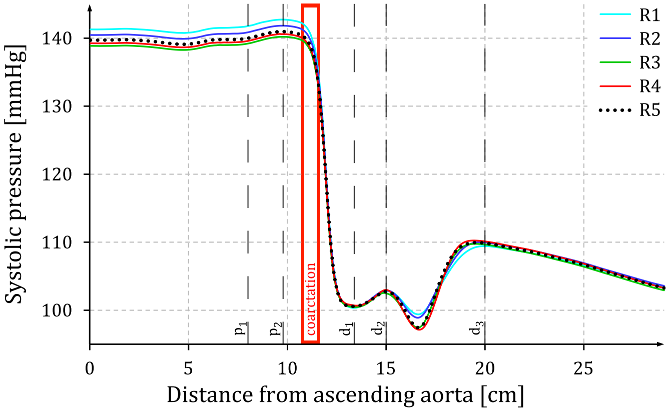


Figure 4.6: Pressure along the centerline at peak systole for increasing mesh densities (R1 to R4) and a grid with a local refinement at the coarctation region and a gradual coarsening towards the descending aorta (R5).

4.4 Conclusion

An aortic arch with three side branches was obtained from medical images and further processed by including a severe stenosis in the upper descending aorta, distal to the left subclavian artery. Four uniform grids for both the fluid domain and the arterial wall were generated using the extended Treemesh method in a limited amount of operator time. The fluid meshes – ranging from $42k$ to $408k$ cells – feature high-quality hexahedral cells and grid alignment with the predominant direction of the flow. The solid meshes – ranging from $21k$ to $102k$ cells – feature an additional cell quality improvement by enhancing the fluid-structure interface prior to its radial extension. The computational grids were used in an FSI model with physiological boundary conditions, and a consistent (inlet to outlet) pressure drop of $36\,mmHg$ was obtained. As FSI simulations are computationally expensive and time consuming, it is important to use a computational grid for which accurate results are obtained with a limited number of cells. In this regard, it was demonstrated that the extended Treemesh method allowed for axial, radial as well as circumferential refinement in the region of the coarctation with a gradual decrease in mesh density towards the distal descending aorta. As such, the highly disturbed flow distal to the stenosis was captured with sufficient accuracy and without having to attain the same mesh density in other regions of the fluid domain, what resulted in a reduction of the computation time.

Two

Inverse modelling of image-based patient-specific blood vessels:

The Backward Displacement Method

Chapter 5

Introduction to the prestress problem in arteries and limitations of existing methodologies

5.1 Introduction

In the last decade computational methods, medical imaging techniques and computational power have improved tremendously, leading to an increasing interest in numerical simulations related to the cardiovascular system. Motivations to perform these numerical analyses and – more importantly – to improve their solution methods are (i) to give medical researchers a better comprehension and insight into the pathology of cardiovascular diseases, (ii) to offer them a computational environment allowing to test new and evaluate existing medical procedures and (iii) the general belief that patient-specific computational models will eventually be used in clinical practice.

Nowadays non-invasive imaging techniques such as X-ray computed tomography (CT) and magnetic resonance imaging (MRI) allow for accurate in vivo visualization of 3D patient-specific geometries that are part of the cardiovascular system. The obtained geometries can be completed by flow, wall thickness and wall motion measurements gathered by MRI and/or ultrasound to provide data for patient-specific boundary conditions for the numerical simulations. Together with pressure measurements there is sufficient data available to include patient-specific boundary conditions in computational fluid dynamics (CFD), computational structural dynamics (CSD) and even fluid-structure interaction (FSI) simulations. In CFD models the flow-related measurements can be imposed directly (after correcting for the flow mismatch between in- and outlets) [75] or incorporated into windkessel and/or 1D models [36] at the inlet and outlets. In CSD models the in vivo data can be used to optimize the constitutive law for the tissue material [21] originally obtained by fitting ex vivo measurements. Both models can be coupled to take into account the interaction between the blood flow and the arterial wall in an FSI simulation.

When focusing on the modelling procedure of the arterial system and the arterial wall in particular, it is important to keep in mind that a physiological pressure load is present at the moment of image acquisition and therefore a stress and strain field is present in the in vivo obtained patient-specific geometry. Several authors have shown that neglecting the presence of this physiological pressure load inside blood vessels in general and inside cerebral or aortic aneurysms in particular results in an inaccurate rating of the stresses and deformations [14, 27, 43, 44, 64]. As (i) the stress distribution throughout the arterial wall cannot be measured and (ii) it is impossible to measure in vivo the zero-pressure geometry of a blood vessel, an inverse problem has to be defined to solve for this geometry or stress field when the in vivo measured geometry and the corresponding internal pressure at the moment of medical imaging are known.

This inverse problem and its solution methods are not only applicable in biomechanical models. There are many engineering applications in which a load free configuration or an initial stress distribution may be useful. For example in production and design processes it can be used to calculate the yet to be manufactured shape of gaskets or rubber seals and even the geometry of the manufacturing tools themselves (e.g. the elastomer puncher in a rubber pad forming process for metal sheets), such that the desired shape is reached under loading conditions [29, 30]. Another example can be found in the design process of a turbine blade where the geometrical outcome of a CFD optimization study has to be attained when the blade is subjected to pressure and centrifugal forces [22].

Note that this dissertation only deals with recovering the stress field resulting from an internal pressure load and not with recovering the residual stress field which is still present in an arterial ring segment (after excision). Indeed, when such a ring segment is cut radially, it opens up – characterized by an opening angle – and the corresponding residual strains are released, recovering a zero-stress state. As such, this work only recovers the zero-pressure geometry.

5.2 Mathematical problem description

Before defining the inverse problem, a general forward problem is formulated. Therefore, we define a stress free reference configuration by

$$\Omega(\boldsymbol{X}, \mathbf{0}) \tag{5.1}$$

in which $\boldsymbol{X}$ denotes the material coordinates of the undeformed reference geometry, and where the second argument of the configuration Ω refers to the zero stress state that corresponds to this unloaded reference configuration. Then, a forward analysis can be defined as the calculation of the equilibrium configuration

$$\Omega(\boldsymbol{x}, \boldsymbol{\sigma}) \tag{5.2}$$

with $\boldsymbol{x}$ the coordinates of the deformed geometry and $\boldsymbol{\sigma}$ the second-order stress tensor. As shown in Figure 5.1, this deformed configuration results from a pressure load p, applied at the inner surface of the undeformed blood vessel wall,

$$p = -\boldsymbol{T}.\boldsymbol{n} = -(\boldsymbol{\sigma}.\boldsymbol{n}).\boldsymbol{n}$$

with $\boldsymbol{T}$ the traction vector at the inner wall and $\boldsymbol{n}$ the outward unit normal vector. A zero traction vector is applied at the outer wall of the undeformed reference state (5.1), if external tissue support is not considered. Furthermore, the nodes at the ending cross sections of the model in Figure 5.1 are only allowed to move in radial direction with respect to the local centerline

$$\begin{cases} U_\theta = 0 \\ U_z = 0 \end{cases} \tag{5.3}$$

However, if the axial residual stress has to be taken into account, one can set $U_z \neq 0$ or define an additional axial pressure load.

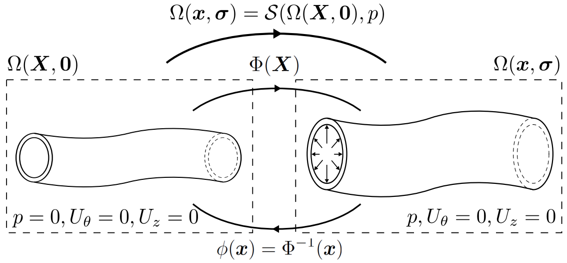

$\Omega(\boldsymbol{x},\boldsymbol{\sigma}) = \mathcal{S}(\Omega(\boldsymbol{X},\mathbf{0}),p)$
$\Omega(\boldsymbol{X},\mathbf{0})$
$\Phi(\boldsymbol{X})$
$\Omega(\boldsymbol{x},\boldsymbol{\sigma})$
$p = 0, U_\theta = 0, U_z = 0$
$p, U_\theta = 0, U_z = 0$
$\phi(\boldsymbol{x}) = \Phi^{-1}(\boldsymbol{x})$

Figure 5.1: Schematic representation and notations of the forward problem. The boundary conditions are represented in the rectangles, with a pressure load (p) at the inner wall surface, a zero traction vector at the outer wall surface, a zero circumferential and axial displacement at the ending cross sections ($U_\theta, U_z = 0$).

To be in equilibrium, the equilibrium conditions, the compatibility requirements and the presence of an appropriate material model, which sets the relation between the stress field and the strain field, are satisfied. The equilibrium configuration (5.2) can be computed by a structural solver, which we denote by $\mathcal{S}$. Using these definitions, we define the forward analysis by

$$\Omega(\boldsymbol{x},\boldsymbol{\sigma}) = \mathcal{S}(\Omega(\boldsymbol{X},\mathbf{0}),p) \tag{5.4}$$

The deformation can be defined by the forward mapping $\Phi : \boldsymbol{X} \mapsto \boldsymbol{x}$ and the deformation gradient tensor $\boldsymbol{F}$

$$\boldsymbol{x} = \Phi(\boldsymbol{X}) \tag{5.5a}$$

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \frac{\partial \Phi(\boldsymbol{X})}{\partial \boldsymbol{X}} \tag{5.5b}$$

The inverse or backward problem calculates the undeformed reference geometry that corresponds to a given geometry, which is deformed due to a pressure load, see Figure 5.2. Therefore, in contrast to Figure 5.1, we now assume

$$\begin{cases} \boldsymbol{X} = \boldsymbol{X}^* \\ \boldsymbol{\sigma} = \boldsymbol{\sigma}^* \end{cases}, \qquad \begin{cases} \boldsymbol{x} = \boldsymbol{x}_m \\ p = p_m \end{cases}$$

where $\boldsymbol{X}^*$ and $\boldsymbol{\sigma}^*$ are the zero-pressure geometry present in the undeformed reference configuration and the stress state present in the in vivo configuration, *i.e.* the unknown variables of this inverse problem. The in vivo geometry $\boldsymbol{x}_m$ and the internal pressure load p_m are the known input parameters for the inverse problem, where the subscript m refers to (in vivo) measurements. Then, the backward problem can be formulated as follows:
Find the in vivo configuration

$$\Omega(\boldsymbol{x}_m,\boldsymbol{\sigma}^*) \tag{5.6}$$

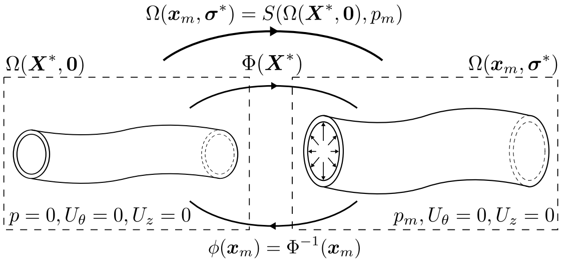

$\Omega(\boldsymbol{x}_m,\boldsymbol{\sigma}^*) = S(\Omega(\boldsymbol{X}^*,\mathbf{0}),p_m)$
$\Omega(\boldsymbol{X}^*,\mathbf{0})$
$\Phi(\boldsymbol{X}^*)$
$\Omega(\boldsymbol{x}_m,\boldsymbol{\sigma}^*)$
$p=0, U_\theta=0, U_z=0$
$p_m, U_\theta=0, U_z=0$
$\phi(\boldsymbol{x}_m) = \Phi^{-1}(\boldsymbol{x}_m)$

Figure 5.2: Schematic representation and notations for the inverse problem. The boundary conditions are represented in the rectangles, with a pressure load (p) at the inner wall surface, a zero traction vector at the outer wall surface, a zero circumferential and axial displacement at the ending cross sections ($U_\theta, U_z = 0$).

which is yet unknown as only $\boldsymbol{x}_m$ is known and $\boldsymbol{\sigma}^*$ is not, and which is in equilibrium with the measured internal pressure load p_m, the zero traction at the outside and the kinematic Dirichlet boundary conditions (5.3). Therefore, find the corresponding undeformed reference configuration

$$\Omega(\boldsymbol{X}^*, \mathbf{0}) \tag{5.7}$$

so that the in vivo equilibrium configuration (5.6) can be found by imposing the in vivo measured pressure p_m onto the zero-pressure reference configuration (5.7) in a forward analysis (5.4)

$$\Omega(\boldsymbol{x}_m, \boldsymbol{\sigma}^*) = \mathcal{S}(\Omega(\boldsymbol{X}^*, \mathbf{0}), p_m) \tag{5.8}$$

Equation (5.8) results in the stress tensor field $\boldsymbol{\sigma}^*$ which is defined as the *prestress* introduced by the forward analysis or the *in vivo stress* accompanying the in vivo image-based geometry. The unloaded reference geometry can be written as

$$\boldsymbol{X}^* = \phi(\boldsymbol{x}_m) = \Phi^{-1}(\boldsymbol{x}_m)$$

in which $\boldsymbol{\phi} : \boldsymbol{x} \mapsto \boldsymbol{X}$ denotes the inverse deformation mapping. This allows to obtain the original in vivo geometry at the moment of imaging $\boldsymbol{x}_m$ using the in (5.5a) proposed forward deformation of the zero-pressure geometry $\boldsymbol{X}^*$

$$\Phi(\boldsymbol{X}^*) = \Phi(\phi(\boldsymbol{x}_m)) = \Phi(\Phi^{-1}(\boldsymbol{x}_m)) = \boldsymbol{x}_m$$

5.3 Literature overview of existing methodologies

Besides the backward displacement method, proposed in Chapter 6, there exist other techniques to incorporate in vivo stress into computational models of the cardiovascular system, given the in vivo image-based geometry $\boldsymbol{x}_m$ and the internal pressure load at the moment of image acquisition p_m.

The idea to take the zero-pressure geometry into account in a non-invasive way in the numerical modelling process of a blood vessel was first implemented by Raghavan et al. [53]. Their method can be described as an optimization procedure for finding the parameter k that approximates the coordinates of the unknown zero-pressure reference geometry $\boldsymbol{X}^*$ by $(\boldsymbol{x}_m - k\boldsymbol{U})$. Here, the nodal displacements $\boldsymbol{U}$ result from a single forward calculation where the lumen pressure load is applied onto the in vivo measured reference geometry $\boldsymbol{x}_m$.
The inverse elastostatic method as described by Govindjee and Mihalic [29] and later extended by the same authors for nearly incompressible materials [30] was introduced to the field of cardiovascular modelling by Lu et al. as another way

to calculate the zero-pressure state [43]. However, the implementation involves access to the finite element code, which can be seen as a drawback.

Gee et al. implemented the previous strategy as the Inverse Design (ID) method, and compared this prestressing technique with another method, the so called Modified Updated Lagrangian Formulation (MULF) [27, 28]. The methodology used, is similar to the Backward Incremental (BI) method introduced by de Putter et al. [14]. In contrast to the backward displacement method of Chapter 6, the zero-pressure geometry is not calculated directly but the equilibrium configuration $\Omega(\boldsymbol{x}_m, \boldsymbol{\sigma}^*)$ is computed instead. By incrementally increasing the pressure load towards the full in vivo pressure p_m while discarding the corresponding deformations $(\boldsymbol{x}^i = \boldsymbol{x}_m)$ a prestressed $(\boldsymbol{\sigma}^*)$ and prestrained configuration is generated. The procedure calculates the new stress tensor field $\boldsymbol{\sigma}^i$ that will be used at the next increment by loading the i^{th} non-equilibrium configuration $\Omega(\boldsymbol{x}_m, \boldsymbol{\sigma}^{i-1})$ with the incrementally increased internal pressure $(p^i = p^{i-1} + \delta p^i)$.

As such, the stress tensor gets updated towards the in vivo stress tensor and the strain tensor gets implicitly updated by a multiplicative split of the deformation gradient tensor $\boldsymbol{F}_{0,i} = \boldsymbol{F}_{0,i-1}.\boldsymbol{F}_{i-1,i}$ [1, 64]. Afterwards, the zero-pressure geometry can be constructed by reducing the luminal pressure to 0 Pa [65]. The accuracy depends on the incremental application of the pre-existing load. According to [14] the last pressure increment has to be chosen extremely small to return a final equilibrium configuration and enough increments should be chosen. The method allows the use of a black box structural solver if the finite element code is able to update the initial stress tensor field with each iteration [65]. A similar approach of prestressing was used earlier by Pinsky et al. to include the internal stress state in the cornea under the presence of the full intraocular pressure load through a fixed point iteration instead of increasing the pressure incrementally [48].

Although the above methods only involve a structural model of the arterial system under consideration, their resulting zero-pressure geometry or in vivo stress state can also be used in FSI models, which take into account the interaction between the blood flow and the arterial wall, as the effect of viscous forces would be small. However, Bazilevs et al. report that the effect is not negligible [8] and account for the viscous traction caused by the fluid when solving the balance of linear momentum for the solid. They obtain the fluid traction vector from a separate steady flow CFD simulation with rigid walls. The prestress component $\boldsymbol{S}_0$ of the additive decomposition $(\boldsymbol{S} + \boldsymbol{S}_0)$ of the second Piola-Kirchhoff stress tensor is then calculated iteratively and used afterwards as initial stress in their in vivo geometry based FSI model [35].

In this work, the backward displacement method (BDM) is presented to solve for the zero-pressure geometry by iteratively updating the nodal coordinates towards the unknown unloaded configuration. Applying the arterial pressure, present at the moment of imaging, in a forward analysis fully recovers the in vivo measured geometry and restores its stress state. The main concern was to come with an algorithm which allows for an easy and straightforward implementation and a fast convergence. This was explicitly confirmed by other authors [59, 84].

Chapter 6

The backward displacement method [1]

To solve the backward problem of section 5.2, this Chapter proposes the backward displacement method (BDM). The method computes the zero-pressure geometry of an *in vivo* measured structure iteratively by subtracting nodal displacements from the corresponding coordinates of the *in vivo* measured geometry. These nodal displacements are updated every iteration; they are the result of a forward structural simulation in which the approximation of the zero-pressure geometry is inflated by the *in vivo* measured pressure. The *in vivo* stress present in the *in vivo* measured structure can be calculated by inflating the computed zero-pressure geometry to the *in vivo* measured pressure. In this Chapter, the BDM is solved using two numerical methods. The first method proposed in Section 6.2 solves the backward displacement problem using fixed point (FP) iterations. The second method presented in Section 6.3 accelerates the backward displacement method using a quasi-Newton (QN-ILS) algorithm. Both algorithms make use of a forward structural analysis to update the approximate zero-pressure geometry,

[1] The fixed point approach, proposed in this chapter was first presented at the *Fifth international conference on advanced computational methods in engineering (ACOMEN 2011)* [11] and further based on the article *"A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels."*, as published in the *Journal of Computational and Applied Mathematics*, 2013, and rewarded by a Mimics Innovation Award (MIA) in the same year. The quasi-Newton approach is based on the article *"Inverse modelling of image-based patient-specific blood vessels: zero-pressure geometry and in vivo stress incorporation."*, as published in *ESAIM: Mathematical Modelling and Numerical Analysis*, 2013.

evaluating the residual vector as the distance that is still present between the image-based geometry and the geometry resulting from this forward problem. When convergence is reached (i) a zero-pressure geometry is found and (ii) the resulting *in vivo* measured geometry is recovered and in equilibrium with an *in vivo* stress field and the *in vivo* load. Furthermore, only the nodal coordinates of the mesh need to be updated before every iteration, allowing for a straightforward implementation in combination with existing structural solvers, even if the solver is a black box and no access is granted to the source code (as is the case with most commercial packages).

6.1 Definitions

The backward displacement function $\mathcal{B}$ is defined as

$$\begin{aligned}\mathcal{B}(\boldsymbol{U}) &= \boldsymbol{x}_m - \boldsymbol{U}\\ &= \tilde{\boldsymbol{X}}\end{aligned} \tag{6.1}$$

where the output coordinates $\tilde{\boldsymbol{X}} \in \mathbb{R}^{3N\times 1}$ are found by subtracting a displacement variable $\boldsymbol{U} \in \mathbb{R}^{3N\times 1}$ from the *in vivo* measured geometry $\boldsymbol{x}_m \in \mathbb{R}^{3N\times 1}$. With N the total number of nodes in the three dimensional model, the variables are represented by column vectors as this notation is required for the quasi-Newton formulation of Section 6.3.

The structural solver was introduced in Equation (5.4) by the function $\mathcal{S}(\Omega, p)$ to emphasize that the solver can be treated as a black box. When one is interested in the displacement resulting from the forward analysis (5.4) in which all input variables except for the coordinates of the reference geometry $\boldsymbol{X} \in \mathbb{R}^{3N\times 1}$ are known, we can define the displacement output of the structural solver by the function $\mathcal{S}_U$

$$\begin{aligned}\mathcal{S}_U(\boldsymbol{X}) = \boldsymbol{x} - \boldsymbol{X} &= \mathcal{S}(\Omega(\boldsymbol{X}, \boldsymbol{0}), p_m) - \boldsymbol{X}\\ &= \boldsymbol{U}\end{aligned} \tag{6.2}$$

The by $\boldsymbol{x} \in \mathbb{R}^{3N\times 1}$ represented deformed geometry, and thus the nodal displacement output $\boldsymbol{U} \in \mathbb{R}^{3N\times 1}$ of the function $\mathcal{S}_U$ can be found by solving the forward structural problem (5.4).

Using these definitions, the inverse problem can be described by

$$\tilde{\boldsymbol{X}} = \mathcal{B}(\mathcal{S}_U(\boldsymbol{X})) \tag{6.3}$$

where a tilde was introduced to distinguish between the output variable of $\mathcal{B}(\mathcal{S}_U)$ and its input variable, as both are in general not the same. However, when the backward problem of Section 5.2 has been solved, Equation (5.8) has to be satisfied, *i.e.* a forward analysis of the zero-pressure reference configuration $\Omega(\boldsymbol{X}^*, \mathbf{0})$ has to result in the *in vivo* configuration $\Omega(\boldsymbol{x}_m, \boldsymbol{\sigma}^*)$ when applying the internal pressure load p_m present during medical imaging. When this constraint is met, Equation (6.3) yields the fixed point problem

$$\tilde{\boldsymbol{X}} = \boldsymbol{X} = \boldsymbol{X}^* \tag{6.4}$$

with $\boldsymbol{X}^* \in \mathbb{R}^{3N\times 1}$ the final zero-pressure geometry. The prestress tensor $\boldsymbol{\sigma}^*$ results from the same forward analysis and can be used as an initial stress tensor in consecutive simulations. The corresponding root finding formulation can be described by

$$\mathcal{R}_X(\boldsymbol{X}) = \mathcal{B}(\mathcal{S}_U(\boldsymbol{X})) - \boldsymbol{X} = \tilde{\boldsymbol{X}} - \boldsymbol{X} = \mathbf{0} \tag{6.5}$$

with $\mathcal{R}_X$ the residual operator in function of the unloaded material coordinates.

6.2 Fixed point strategy

Equations (6.3) and (6.4) suggest that a backward displacement method based on fixed point iterations can be used to solve for the zero-pressure geometry. The fixed point algorithm to recover this zero-pressure geometry and the *in vivo* stress tensor field is shown in Algorithm 6.1. It starts by initializing an approximation for the zero-pressure geometry $\boldsymbol{X}^{i=0}$. As initial guess, the original image-based geometry $\boldsymbol{x}_m$ is chosen. Then, a fixed point based iterative procedure is performed until convergence is reached. First, the structural solver calculates an equilibrium configuration $\Omega(\boldsymbol{x}^i, \boldsymbol{\sigma}^i)$ from the intermediate reference configuration $\Omega(\boldsymbol{X}^i, \mathbf{0})$ loaded with the full *in vivo* pressure load p_m. The displacements of the material points in the forward analysis are denoted by $\boldsymbol{U}^i$. Afterwards, the approximation of the zero-pressure geometry ($\boldsymbol{X}^i$) is updated by subtracting the nodal displacements $\boldsymbol{U}^i$ from the *original* image-based coordinates $\boldsymbol{x}_m$. Finally, this procedure leads to an update of the mesh $\boldsymbol{X}^{i+1} = \tilde{\boldsymbol{X}}^i$ used in the next iteration (step 13 in Algorithm 6.1) or, if convergence is reached, to the zero-pressure geometry $\boldsymbol{X}^*$. Furthermore, the forward analysis calculates the stress state $\boldsymbol{\sigma}^i$ (step 9 in Algorithm 6.1) which is left unused throughout the algorithm but represents the *in vivo* stress tensor $\boldsymbol{\sigma}^*$ present in the *in vivo* measured geometry upon convergence. During the iterative process, a residual vector $\boldsymbol{r}^i \in \mathbb{R}^{3N\times 1}$ is calculated

$$\boldsymbol{r}^i = \tilde{\boldsymbol{X}}^i - \boldsymbol{X}^i = (\boldsymbol{x}_m - \boldsymbol{U}^i) - \boldsymbol{X}^i \tag{6.6a}$$

$$= \boldsymbol{x}_m - (\boldsymbol{x}^i - \boldsymbol{X}^i) - \boldsymbol{X}^i \tag{6.6b}$$

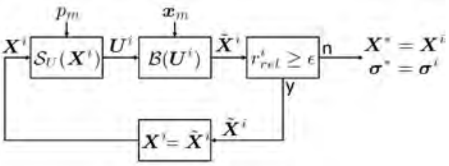
$$= \boldsymbol{x}_m - \boldsymbol{x}^i \tag{6.6c}$$

This residual vector is a measure for the distances that are still present between the image-based geometry and the deformed geometry resulting from the i^{th} forward analysis. Convergence is reached when the relative residual

$$r^i_{rel} = \frac{\|\boldsymbol{r}^i\|_2}{\|\boldsymbol{r}^0\|_2} < \epsilon \tag{6.7}$$

is smaller than the convergence criterion ϵ. Where $\|\boldsymbol{r}^i\|_2$ stands for the L^2-norm of the residual vector in the i^{th} iteration.

Algorithm 6.1 The backward displacement method using a Fixed point algorithm to recover the zero-pressure geometry and the *in vivo* stress tensor field. (BDM FP)


p_m
$\boldsymbol{x}_m$
$\boldsymbol{X}^i$
$\mathcal{S}_U(\boldsymbol{X}^i)$
$\boldsymbol{U}^i$
$\mathcal{B}(\boldsymbol{U}^i)$
$\tilde{\boldsymbol{X}}^i$
$r^i_{rel} \geq \epsilon$
n
$\boldsymbol{X}^* = \boldsymbol{X}^i$
$\boldsymbol{\sigma}^* = \boldsymbol{\sigma}^i$
y
$\tilde{\boldsymbol{X}}^i$
$\boldsymbol{X}^i = \tilde{\boldsymbol{X}}^i$

```
$i = 0$
Initialize $\boldsymbol{X}^0 = \boldsymbol{x}_m$
$\boldsymbol{U}^0 = \mathcal{S}_U(\boldsymbol{X}^0)$
$\tilde{\boldsymbol{X}}^0 = \mathcal{B}(\boldsymbol{U}^0)$
$\boldsymbol{r}^0 = \tilde{\boldsymbol{X}}^0 - \boldsymbol{X}^0$
$\boldsymbol{X}^1 = \boldsymbol{X}^0 + \boldsymbol{r}^0 = \tilde{\boldsymbol{X}}^0$
while $r^i_{rel} \geq \epsilon$ do
    $i = i + 1$
    $\Omega(\boldsymbol{x}^i, \boldsymbol{\sigma}^i) = \mathcal{S}(\Omega(\boldsymbol{X}^i, \boldsymbol{0}), p_m)$
    $\boldsymbol{U}^i = \mathcal{S}_U(\boldsymbol{X}^i) = \boldsymbol{x}^i - \boldsymbol{X}^i$
    $\tilde{\boldsymbol{X}}^i = \mathcal{B}(\boldsymbol{U}^i) = \boldsymbol{x}_m - \boldsymbol{U}^i$
    $\boldsymbol{r}^i = \tilde{\boldsymbol{X}}^i - \boldsymbol{X}^i$
    $\boldsymbol{X}^{i+1} = \boldsymbol{X}^i + \boldsymbol{r}^i = \tilde{\boldsymbol{X}}^i$
end while
Zero-pressure reference geometry $\boldsymbol{X}^* = \boldsymbol{X}^i$
in vivo stress tensor for the in vivo measured geometry $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}^i$
```

6.3 Quasi-Newton strategy with an approximation of the Jacobian's inverse from a least-squares model

To accelerate the convergence of the backward displacement method, a quasi-Newton iterative method with an approximation for the inverse of the Jacobian from a least-squares model (QN-ILS) can be used. This method uses similar principles as the QN-LS method in which the Jacobian is approximated from a least-squares model and then inverted. The QN-LS method was analyzed for linear problems by Haelterman et al. [32]. The interface quasi-Newton method with an approximation of the inverse of the Jacobian from a least-squares model (IQN-ILS) was introduced by Degroote et al. [19] to couple the interface variables of the flow solver and the structural solver in a partitioned fluid-structure interaction simulation. This method was also analyzed for linear problems by Haelterman et al. [33] in a more general framework, where the method was called QN-ILS. In this work we also call the method QN-ILS as the position is computed for all material points and not only for those lying on the interface. The implementation for a nonlinear problem with black box solvers and with the least-squares QR-factorization is as described in [19] and has proven to need less (coupling) iterations compared with other schemes such as Aitken relaxation and Interface-GMRES(R) [20].

The root finding problem described by Equation (6.5) can be solved using Newton-Raphson iterations. By writing the Jacobian matrix of the residual operator $\mathcal{R}_X$ as

$$\frac{d\mathcal{R}_X}{d\boldsymbol{X}} = \boldsymbol{\mathcal{J}}_X$$

the equations that have to be solved are given by

$$\boldsymbol{\mathcal{J}}_X|_{\boldsymbol{X}^i}\,\Delta\boldsymbol{X}^i = -\boldsymbol{r}^i \tag{6.8a}$$

$$\boldsymbol{X}^{i+1} = \boldsymbol{X}^i + \Delta\boldsymbol{X}^i \tag{6.8b}$$

with the i^{th} residual vector, as shown in Equation (6.6), calculated by

$$\boldsymbol{r}^i = \mathcal{R}_X(\boldsymbol{X}^i) = \mathcal{B}(\mathcal{S}_U(\boldsymbol{X}^i)) - \boldsymbol{X}^i = \tilde{\boldsymbol{X}}^i - \boldsymbol{X}^i \tag{6.9}$$

Convergence of the Newton-Raphson iterations is reached when the relative residual satisfies Equation (6.7).

Unlike what happens using the fixed point iterative procedure, the values for $\mathcal{B}(\mathcal{S}_U(\boldsymbol{X}^i))$ are not passed on to the next backward displacement iteration. $\boldsymbol{X}^{i+1}$ gets computed instead using Equation (6.8a)

$$\Delta\boldsymbol{X}^i = \widehat{\boldsymbol{\mathcal{J}}_X{}^{-1}}\Big|_{\boldsymbol{X}^i}\,(-\boldsymbol{r}^i) \tag{6.10}$$

in which the hat indicates an approximation. However, the exact Jacobian of $\mathcal{R}_X$ is unknown as we do not know the Jacobian of $\mathcal{S}$. We will now go into more detail on how the product of this approximation for the inverse of the Jacobian with the vector $-\boldsymbol{r}^i$ can be calculated, making use of information obtained during previous backward displacement iterations.

$\Delta\boldsymbol{r}^i = \boldsymbol{0} - \boldsymbol{r}^i$ represents the difference between the desired residual vector and the current one. $\Delta\boldsymbol{r}^i$ is approximated by a linear combination of the known $\delta\boldsymbol{r}^j \in \mathbb{R}^{3N\times 1}$ (for $j = 0, ..., i-1$)

$$\Delta\boldsymbol{r}^i \approx \boldsymbol{V}^i\boldsymbol{c}^i \tag{6.11}$$

with $\boldsymbol{c}^i \in \mathbb{R}^{i\times 1}$ the coefficients of the decomposition in iteration i and a matrix $\boldsymbol{V}^i \in \mathbb{R}^{3N\times i}$

$$\boldsymbol{V}^i = [\delta\boldsymbol{r}^{i-1}\ \delta\boldsymbol{r}^{i-2}\ \ldots\ \delta\boldsymbol{r}^1\ \delta\boldsymbol{r}^0] \tag{6.12a}$$

$$\delta\boldsymbol{r}^j = \boldsymbol{r}^{j+1} - \boldsymbol{r}^j \qquad (j = 0, \cdots, i-1) \tag{6.12b}$$

Then, the assumption is made that $\Delta\tilde{\boldsymbol{X}}^i$, corresponding to $\Delta\boldsymbol{r}^i$, can be written by the same linear combination of the, in previous iterations, computed $\delta\tilde{\boldsymbol{X}}^j \in \mathbb{R}^{3N\times 1}$ (for $j = 0, ..., i-1$)

$$\Delta\tilde{\boldsymbol{X}}^i = \boldsymbol{W}^i\boldsymbol{c}^i \tag{6.13}$$

with a matrix $\boldsymbol{W}^i \in \mathbb{R}^{3N\times i}$

$$\boldsymbol{W}^i = [\delta\tilde{\boldsymbol{X}}^{i-1}\ \delta\tilde{\boldsymbol{X}}^{i-2}\ \ldots\ \delta\tilde{\boldsymbol{X}}^1\ \delta\tilde{\boldsymbol{X}}^0] \tag{6.14a}$$

$$\delta\tilde{\boldsymbol{X}}^j = \tilde{\boldsymbol{X}}^{j+1} - \tilde{\boldsymbol{X}}^j \quad (j = 0, \cdots, i-1) \tag{6.14b}$$

Both $\boldsymbol{V}^i$ and $\boldsymbol{W}^i$ are updated at each iteration by the difference between the vectors from the current iteration ($\boldsymbol{r}^i$ and $\tilde{\boldsymbol{X}}^i$) and the vectors from the previous iteration ($\boldsymbol{r}^{i-1}$ and $\tilde{\boldsymbol{X}}^{i-1}$). Remark that if the tildes would be omitted in Equations (6.13) and (6.14), $\Delta\boldsymbol{X}^i (= \boldsymbol{W}^i\boldsymbol{c}^i)$ would become a linear combination of the previously calculated $\delta\boldsymbol{X}^j (j = 0, \cdots, i-1)$, with as result that the rank of $\boldsymbol{W}^{i+1}$ will be the same as the rank of $\boldsymbol{W}^i$. This means that the rank of the matrices $\boldsymbol{W}^i$ would always be equal to one, what would prohibit the convergence of the method since then the rank of the approximation of the inverse of the Jacobian will also become at maximum one. However, for a linear system it has been proven that the use of Equations (6.13) and (6.14) returns an exact Jacobian – and thus a converged result – in at most n+1 iterations, with n the number of degrees of freedom of the problem [33]. Remark, however, that in practical applications the convergence rate is much higher.

Considering that the number of columns in $\boldsymbol{V}^i$ is generally much smaller than the number of rows, determined by the number of nodes in the mesh times the spatial dimension of the problem, Equation (6.11) is an overdetermined system with respect to the elements of $\boldsymbol{c}^i$. Hence, the residual $\Delta\boldsymbol{r}^i = -\boldsymbol{r}^i$ is decomposed as a linear combination of the set of $\delta \boldsymbol{r}^j = \boldsymbol{r}^{j+1} - \boldsymbol{r}^j$ (for $j = 0, ..., i-1$) by solving the least-squares problem

$$\boldsymbol{c}^i = \arg\min_{\boldsymbol{c}^i} \|\Delta \boldsymbol{r}^i - \boldsymbol{V}^i \boldsymbol{c}^i\|_2 \tag{6.15}$$

The coefficients of the decomposition, $\boldsymbol{c}^i$, are given by the normal equations

$$\boldsymbol{c}^i = (\boldsymbol{V}^{i^T} \boldsymbol{V}^i) \boldsymbol{V}^{i^T} \Delta \boldsymbol{r}^i \tag{6.16}$$

However, to be less sensitive to rounding errors introduced during the computations, the least-squares problem is solved by a so-called economy size QR-decomposition of $\boldsymbol{V}^i$

$$\boldsymbol{V}^i = \boldsymbol{Q}^i \boldsymbol{R}^i$$

with $\boldsymbol{Q}^i \in \mathbb{R}^{3N \times i}$ an orthogonal matrix and $\boldsymbol{R}^i \in \mathbb{R}^{i \times i}$ an upper triangular matrix. Because both matrices $\boldsymbol{V}^i$ and $\boldsymbol{W}^i$ are updated at each quasi-Newton iteration the QR-decomposition has to be recalculated. Nevertheless, the cost of the QR-factorization is small compared to the cost of one forward calculation performed by the structural solver $\mathcal{S}_U(\boldsymbol{X}^i)$.

The coefficients of the decomposition $\boldsymbol{c}^i$ are then calculated by solving the triangular system

$$\boldsymbol{R}^i \boldsymbol{c}^i = \boldsymbol{Q}^{i^T} \Delta \boldsymbol{r}^i \tag{6.17}$$

Using Equation (6.9), $\Delta \boldsymbol{r}^i$ can be written as

$$\Delta \boldsymbol{r}^i = \Delta \tilde{\boldsymbol{X}}^i - \Delta \boldsymbol{X}^i$$

Substitution of $\Delta \tilde{\boldsymbol{X}}^i$ by Equation (6.13) and then $\boldsymbol{c}^i$ by Equation (6.17) results in

$$\Delta \boldsymbol{X}^i = \boldsymbol{W}^i \boldsymbol{c}^i - \Delta \boldsymbol{r}^i \tag{6.18a}$$

$$= \left(\boldsymbol{W}^i (\boldsymbol{R}^i)^{-1} \boldsymbol{Q}^{i^T} - \boldsymbol{I} \right) \Delta \boldsymbol{r}^i \tag{6.18b}$$

with $\boldsymbol{I}$ the unity matrix. Identification of Equation (6.18b) with Equation (6.10) shows that the approximation of the inverse of the Jacobian can be calculated by

$$\left. \widehat{\boldsymbol{\mathcal{J}}_X{}^{-1}} \right|_{\boldsymbol{X}^i} = \boldsymbol{W}^i (\boldsymbol{R}^i)^{-1} \boldsymbol{Q}^{i^T} - \boldsymbol{I} \tag{6.19}$$

However, the matrix $\left. \widehat{\boldsymbol{\mathcal{J}}_X{}^{-1}} \right|_{\boldsymbol{X}^i}$ is never calculated nor stored, but the matrix vector product of $\left. \widehat{\boldsymbol{\mathcal{J}}_X{}^{-1}} \right|_{\boldsymbol{X}^i}$ with $\Delta \boldsymbol{r}^i$ is computed directly from the calculated coefficients $\boldsymbol{c}^i$, the matrix $\boldsymbol{W}^i$ and the residual vector $\boldsymbol{r}^i$

$$\begin{aligned} \boldsymbol{X}^{i+1} &= \boldsymbol{X}^i + \left. \widehat{\boldsymbol{\mathcal{J}}_X{}^{-1}} \right|_{\boldsymbol{X}^i} (-\boldsymbol{r}^i) \\ &= \boldsymbol{X}^i + \boldsymbol{W}^i \boldsymbol{c}^i + \boldsymbol{r}^i \end{aligned}$$

so that the required storage for the method is very limited and the calculation time is negligible.

It can be demonstrated that quasi-Newton iterations are performed for the part of $\Delta\boldsymbol{r}^i$ in the span of the columns of $\boldsymbol{V}^i$ (Proposition 6.3.1) and that fixed point iterations are performed for the part of $\Delta\boldsymbol{r}^i$ orthogonal to the span of the columns of $\boldsymbol{V}^i$ (Proposition 6.3.2). Algorithm 6.2 shows the complete QN-ILS technique to recover the zero-pressure geometry $\boldsymbol{X}^*$ and to find the *in vivo* stress $\boldsymbol{\sigma}^*$.

Proposition 6.3.1. *Quasi-Newton iterations are performed for the part of* $\Delta\boldsymbol{r}^i$ *in the column span of* $\boldsymbol{V}^i$

Proof. Let $\Delta\boldsymbol{r}^i$ be an element of the column span of $\boldsymbol{V}^i = [\delta\boldsymbol{r}^{i-1} \dots\ \delta\boldsymbol{r}^1\ \delta\boldsymbol{r}^0]$

$$\Delta\boldsymbol{r}^i = \sum_{j=0}^{i-1} \alpha_j \delta\boldsymbol{r}^j$$

with α_j the coefficients of the linear combination. Then, the components of the decomposition in iteration i, Equations (6.11) and (6.13), are given by

$$\boldsymbol{c}^i = \arg\min_{\boldsymbol{c}^i} \|\Delta\boldsymbol{r}^i - \boldsymbol{V}^i\boldsymbol{c}^i\|_2 = [\alpha_{i-1}\ \dots\ \alpha_1\ \alpha_0]^T$$

Equation (6.18a) can then be reformulated using Equations (6.12b), (6.14b) and (6.9)

$$\begin{aligned}
\Delta\boldsymbol{X}^i = \widehat{\boldsymbol{\mathcal{J}}^{-1}}\Big|_{\boldsymbol{X}^i} \Delta\boldsymbol{r}^i &= \boldsymbol{W}^i\boldsymbol{c}^i - \Delta\boldsymbol{r}^i \\
&= \sum_{j=0}^{i-1} \delta\tilde{\boldsymbol{X}}^j c_j^i - \sum_{j=0}^{i-1} \alpha_j \delta\boldsymbol{r}^j \\
&= \sum_{j=0}^{i-1} \Big((\tilde{\boldsymbol{X}}^{j+1} - \tilde{\boldsymbol{X}}^j)\, c_j^i - \alpha_j (\boldsymbol{r}^{j+1} - \boldsymbol{r}^j) \Big) \\
&= \sum_{j=0}^{i-1} \bigg((\tilde{\boldsymbol{X}}^{j+1} - \tilde{\boldsymbol{X}}^j)\, \alpha_j - \alpha_j \Big((\tilde{\boldsymbol{X}}^{j+1} - \boldsymbol{X}^{j+1}) - (\tilde{\boldsymbol{X}}^j - \boldsymbol{X}^j) \Big) \bigg) \\
&= \sum_{j=0}^{i-1} \alpha_j (\boldsymbol{X}^{j+1} - \boldsymbol{X}^j) \\
&= \sum_{j=0}^{i-1} \alpha_j \delta\boldsymbol{X}^j
\end{aligned}$$

Thus, iteration i ends up as a quasi-Newton iteration in which the product of the Jacobian's inverse with the vector $\Delta\boldsymbol{r}^i$ can be calculated by a linear combination of known finite differences $\delta\boldsymbol{X}^j$ (with $j = 0, ..., i-1$). □

Proposition 6.3.2. *Fixed point iterations are performed for the part of $\Delta \boldsymbol{r}^i$ orthogonal to the column span of $\boldsymbol{V}^i$*

Proof. Let $\Delta \boldsymbol{r}^i$ be orthogonal to $\boldsymbol{V}^i$. Then, the components of the decomposition in iteration i are given by

$$\boldsymbol{c}^i = \arg\min_{\boldsymbol{c}^i} \|\Delta \boldsymbol{r}^i - \boldsymbol{V}^i \boldsymbol{c}^i\|_2 = [0 \ \dots \ 0]^T$$

$$\begin{aligned}\Delta \boldsymbol{X}^i &= \boldsymbol{W}^i \boldsymbol{c}^i - \Delta \boldsymbol{r}^i \\ &= \boldsymbol{0} - \Delta \boldsymbol{r}^i \\ &= \boldsymbol{r}^i\end{aligned}$$

Here, iteration i ends up as a fixed point iteration (see step 13 in Algorithm 6.1)

$$\boldsymbol{X}^{i+1} = \boldsymbol{X}^i + \boldsymbol{r}^i = \tilde{\boldsymbol{X}}^i$$

□

Remark 6.1. Information from the previous backward displacement iterations is needed to calculate the approximation for the inverse of the Jacobian with the vector $-\boldsymbol{r}^i$. Therefore a relaxation with factor ω is performed in the first backward displacement iteration $i = 0$. As the same initial guess ($\boldsymbol{X}^0 = \boldsymbol{x}_m$) was made before when performing the backward displacement method using simple fixed point iterations, the values for the first two relative residuals, found by both methods, will be the identical when $\omega = 1$.

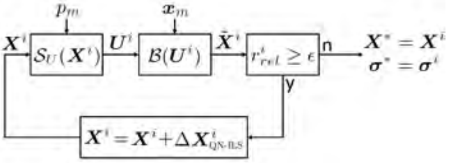
Remark 6.2. When solving for the zero-pressure configuration using Algorithm 6.1 or 6.2 the only variables which get updated explicitly are the material coordinates of the approximate zero-pressure geometry $\boldsymbol{X}^i$. However, the fixed point or root finding problem can perfectly be described in terms of $\boldsymbol{U}$ instead of $\boldsymbol{X}$. Definitions from Section 6.1 and associated solving procedures in Sections 6.2 and 6.3 can be reformulated in order to solve the inverse problem in function of the displacement. Here, a tilde is introduced to distinguish between the output variable of $\mathcal{S}_U(\mathcal{B}(U))$ and its input variable. The fixed point problem can be described by

$$\begin{aligned}\tilde{\boldsymbol{U}} &= \mathcal{S}_U(\mathcal{B}(\boldsymbol{U})) \\ &= \boldsymbol{U} = \boldsymbol{U}^*\end{aligned}$$

while the corresponding root finding formulation becomes

$$\mathcal{R}_U(\boldsymbol{U}) = \mathcal{S}_U \mathcal{B}(\boldsymbol{U})) - \boldsymbol{U} = \tilde{\boldsymbol{U}} - \boldsymbol{U} = \boldsymbol{0}$$

Algorithm 6.2 The backward displacement method using a Quasi-Newton algorithm with an approximation for the inverse of the Jacobian from a least-squares model to recover the zero-pressure geometry and the *in vivo* stress tensor field. (BDM QN-ILS)


p_m
$\boldsymbol{x}_m$
$\boldsymbol{X}^i$
$\mathcal{S}_U(\boldsymbol{X}^i)$
$\boldsymbol{U}^i$
$\mathcal{B}(\boldsymbol{U}^i)$
$\tilde{\boldsymbol{X}}^i$
$r^i_{rel} \geq \epsilon$
n
y
$\boldsymbol{X}^* = \boldsymbol{X}^i$
$\boldsymbol{\sigma}^* = \boldsymbol{\sigma}^i$
$\boldsymbol{X}^i = \boldsymbol{X}^i + \Delta \boldsymbol{X}^i_{\text{QN-ILS}}$

```
$i = 0$
Initialize $\boldsymbol{X}^0 = \boldsymbol{x}_m$
$\boldsymbol{U}^0 = \mathcal{S}_U(\boldsymbol{X}^0)$
$\tilde{\boldsymbol{X}}^0 = \mathcal{B}(\boldsymbol{U}^0)$
$\boldsymbol{r}^0 = \tilde{\boldsymbol{X}}^0 - \boldsymbol{X}^0$
$\boldsymbol{X}^1 = \boldsymbol{X}^0 + \omega \boldsymbol{r}^0$
while $r^i_{rel} \geq \epsilon$ do
    $i = i + 1$
    $\Omega(\boldsymbol{x}^i, \boldsymbol{\sigma}^i) = \mathcal{S}(\Omega(\boldsymbol{X}^i), p_m)$
    $\boldsymbol{U}^i = \mathcal{S}_U(\boldsymbol{X}^i) = \boldsymbol{x}^i - \boldsymbol{X}^i$
    $\tilde{\boldsymbol{X}}^i = \mathcal{B}(\boldsymbol{U}^i) = \boldsymbol{x}_m - \boldsymbol{U}^i$
    $\boldsymbol{r}^i = \tilde{\boldsymbol{X}}^i - \boldsymbol{X}^i$
    Construct $\boldsymbol{V}^i$ and $\boldsymbol{W}^i$ as shown in Equation (6.12a) and (6.14a)
    Calculate QR-decomposition $\boldsymbol{V}^i = \boldsymbol{Q}^i \boldsymbol{R}^i$
    Solve $\boldsymbol{R}^i \boldsymbol{c}^i = -{\boldsymbol{Q}^i}^T \boldsymbol{r}^i$
    $\boldsymbol{X}^{i+1} = \boldsymbol{X}^i + \Delta \boldsymbol{X}^i = \boldsymbol{X}^i + \boldsymbol{W}^i \boldsymbol{c}^i + \boldsymbol{r}^i$
end while
Zero-pressure reference geometry $\boldsymbol{X}^* = \boldsymbol{X}^i$
in vivo stress tensor for the in vivo measured geometry $\boldsymbol{\sigma}^* = \boldsymbol{\sigma}^i$
```

with $\mathcal{R}_U$ the residual operator in function of the nodal displacement.

Moreover, both representations of the inverse problem yield the same theoretical result but with an opposite sign, as shown here for the root finding formulation

$$\mathcal{R}_U(\boldsymbol{U}^i) = \tilde{\boldsymbol{U}}^i - \boldsymbol{U}^i = \mathcal{S}_U(\mathcal{B}(\boldsymbol{U}^i)) - \boldsymbol{U}^i \tag{6.21a}$$

$$= \mathcal{S}_U(\boldsymbol{X}^i) - (\boldsymbol{x}_m - \boldsymbol{X}^i) \tag{6.21b}$$

$$= \boldsymbol{X}^i - \mathcal{B}(\mathcal{S}(\boldsymbol{X}^i)) = -(\tilde{\boldsymbol{X}}^i - \boldsymbol{X}^i) = -\mathcal{R}_X(\boldsymbol{X}^i) \tag{6.21c}$$

where Equations (6.21b) and (6.21c) are found by using the definition of the backward displacement function $\mathcal{B}$.

6.4 Conclusion

This chapter proposes the backward displacement method to solve for the zero-pressure geometry and the *in vivo* stress tensor field starting from the *in vivo* measured geometry of the loaded structure. After introducing the backward displacement function $\mathcal{B}$ and the function $\mathcal{S}$, which represents the structural solver, the inverse problem can be described by a fixed point problem. The approximation of the zero-pressure geometry gets updated by means of fixed point iterations, in which the i^{th} backward displacement function is applied onto the output of the i^{th} forward structural analysis. In order to accelerate the backward displacement method, the corresponding root finding problem can be solved using a quasi-Newton algorithm with an approximation for the inverse of the Jacobian from a least squares model. This technique incorporates information obtained during previous iterations to increase the convergence rate. In both strategies, the mean residual can get evaluated as the mean distance that is still present between the image-based geometry and the geometry resulting from the forward problem. When convergence is reached (i) a zero-pressure geometry is found and (ii) the resulting *in vivo* measured geometry is fully recovered and in equilibrium with an *in vivo* stress field and the *in vivo* load. Furthermore, only the nodal coordinates of the mesh need to be updated, allowing for a straightforward implementation in combination with existing structural solvers, even if the solver is a black box and no access is granted to the source code (as is the case with most commercial packages).

Chapter 7

Numerical validation and application of the backward displacement method

7.1 Introduction

In this chapter, the methods of Chapter 6 are numerically validated and further applied. Two examples are used to demonstrate the backward displacement method. First, a simplified model of a mouse aorta (represented by a straight cylindrical tube) is used for validation purpose and to evaluate the importance of *in vivo* stress incorporation in the *in vivo* measured geometry. The second example concerns a mouse-specific abdominal aorta with four side branches to explore the ability of the backward displacement method to restore a more complex cardiovascular structure at its zero-pressure state. Both examples cover the comparison of the convergence resulting from the fixed point method and the quasi-Newton method with an approximation of the Jacobian's inverse from a least-squares model. Finally, based on the backward displacement method, a material parameter optimization framework was set up to obtain material parameters which mimic the distension of the arterial wall using *in vivo* obtained data. All structural calculations were performed using the commercial finite element analysis software Abaqus/Standard (Simulia). However, as stated earlier, any other structural mechanics solver can be used.

7.2 Numerical validation of the backward displacement method

7.2.1 Structural model of a thick walled cylinder

A small unloaded artery is modelled using a straight cylindrical tube with a length of 10 mm, an inner radius of 0.5 mm and a wall thickness to diameter ratio of 0.15. The boundary conditions only allow a radial displacement at the ending cross sections. The geometrical model is discretized using 64 × 32 × 4 quadratic hexahedral elements with reduced integration and a hybrid formulation. The vessel wall behaviour is modelled by an incompressible isotropic hyperelastic material using the polynomial strain energy density function

$$W = C_{10}(I_1 - 3) + C_{01}(I_2 - 3) + C_{20}(I_1 - 3)^2 + C_{11}(I_1 - 3)(I_2 - 3) \qquad (7.1)$$

where I_1 and I_2 are the first and the second invariant of the left Cauchy-Green deformation tensor, and where C_{ij} are empirically determined material constants for a human artery according to [50].

7.2.2 Recovering the zero-pressure geometry of a thick-walled cylinder

To validate the backward displacement method a zero-pressure geometry is pressurized in a forward analysis by applying a uniformly distributed pressure load of $80\,mmHg$ to the inner surface of the vessel wall. The resulting geometry is taken as the *in vivo* measured geometry at the diastolic phase and serves as a starting point for the backward problem. This inverse problem is then solved using the backward displacement method in order to restore a zero-pressure geometry which is eventually compared to the original zero-pressure geometry. Therefore, the residual vector $\boldsymbol{r}^i \in \mathbb{R}^{3N\times 1}$ is reshaped into the distance vector $\boldsymbol{d}^i \in \mathbb{R}^{N\times 3}$. Then, the error that is still present after iteration i can be defined as the mean distance between the nodes of the original image-based geometry $\boldsymbol{x}_m$ and the corresponding nodes of the i^{th} deformed geometry $\boldsymbol{x}^i$.

$$d^i_{mean} = \frac{1}{N}\sum_{j=1}^{N} \|\boldsymbol{d}^i_j\|_2$$

with $\boldsymbol{d}^i_j$ row j of matrix $\boldsymbol{d}^i$.

The evolution of the relative residual (6.7) throughout the iterative process is shown on a logarithmic scale in Figure 7.1 for both FP and QN-ILS iterations. This allows evaluation of the rate of convergence, when solving for the zero-pressure geometry, using the backward displacement method. For a convergence criterion set at e.g. 10^{-4}, the logarithm of the relative residual decreases almost linearly during consecutive iterations of the fixed point method. The quasi-Newton residuals however, reduce about one and a half times faster. As the additional cost of the QR-factorization is small compared to the cost of one forward calculation performed by the structural solver, the QN-ILS strategy can be used to accelerate the backward displacement method. Upon convergence, the *in vivo* measured geometry is fully recovered, with a mean distance error d^i_{mean} which is more than a factor 10^4 lower than the convergence criterion.

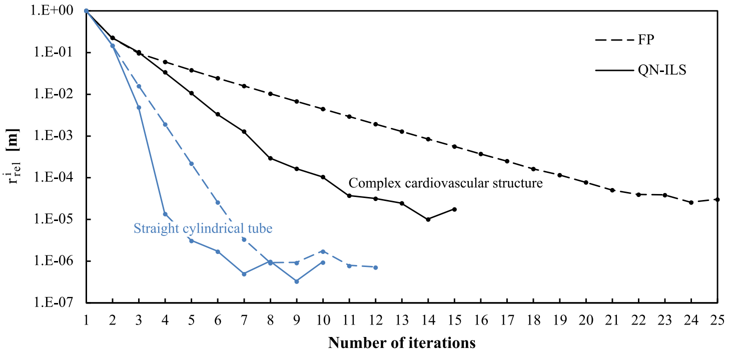


Figure 7.1: Rate of convergence when solving for the zero-pressure geometry of a thick-walled cylinder (example 1) and a mouse-specific abdominal aorta (example 2) using the backward displacement method. Results are shown for both the fixed point (FP) and the quasi-Newton (QN-ILS) approach.

7.2.3 Evaluation of the effect of *in vivo* stress incorporation

To evaluate the effect of *in vivo* stress incorporation, a simulation is performed in which the internal pressure is first set to the end-diastolic pressure ($80\,mmHg$) and subsequently increased to the peak-systolic pressure ($120\,mmHg$). This is done for three different set-ups, with results displayed in Figure 7.2.3:

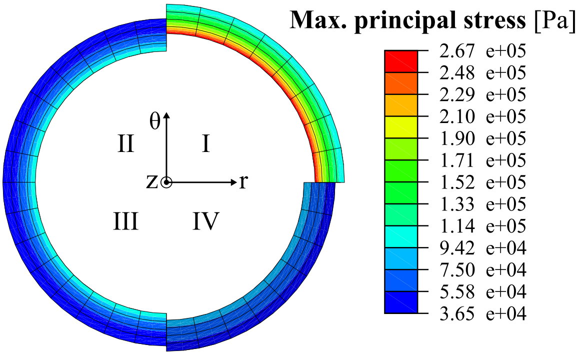
1. The *in vivo* measured geometry is assumed to be the unloaded geometry. In the forward simulation the *in vivo* measured geometry is inflated using the physiological pressure values ($80\,mmHg$ diastolic pressure; $120\,mmHg$ systolic pressure). The results are visualized in quadrant I.
2. The *in vivo* measured geometry is assumed to be the geometry at end-diastole, but neglects the existence of prestress at the diastolic phase. In the forward simulation the *in vivo* measured geometry is only inflated to $40\,mmHg$, the peak-systolic minus end-diastolic pressure difference. To allow for a fair comparison of the calculated stresses with the other two cases, the stress tensor field was corrected by adding an approximation of the stress

field at diastole. The latter resulted from a simulation in which 80 mmHg was applied onto the diastolic geometry. The results are visualized in quadrant IV.

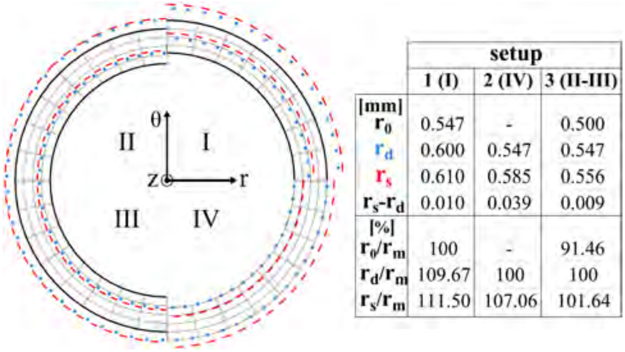
3. The proposed strategy in which the forward simulation towards the physiological pressure values starts from the restored zero-pressure geometry ($p_m = 80\, mmHg$). This results in a prestressed *in vivo* geometry at diastole. The results are visualized in both quadrant II and III.

Figure 7.2(a) presents the maximum principal or circumferential stress at peak-systole ($120\, mmHg$). The contour plots are shown on the corresponding geometry. Figure 7.2(b) visualizes the grid of the undeformed reference geometry, together with inner and outer contours of the zero-pressure (black), the end-diastolic (blue dotted line) and the peak-systolic (red dashed line) geometries. Next to the figure the inner radii, the peak-systolic minus end-diastolic radial displacement, and the ratio of the inner radii to the inner radius of the measured geometry ($r_m = 0.547\, mm$) are tabulated for each of the three setups. By definition, note that the measured geometry equals the zero-pressure geometry in setup 1, and the end-diastolic geometry in setup 2 and 3.

Compared to setup 3, the results in Figure 7.2(a) clearly show the overestimation of the maximum principal stress when the *in vivo* measured geometry is assumed to be the unloaded geometry (setup 1). Furthermore, the outer contours of the cross sectional areas at end-diastole and peak-systole are a better approximation when the *in vivo* geometry is assumed to be the diastolic geometry (setup 2). However, the peak-systolic minus end-diastolic radial displacement in setup 2 is overestimated by a factor of 4.3 due to the nonlinear material law and the absence of prestress/prestrain at the start of the inflation process.



(a)



	setup		
	1 (I)	2 (IV)	3 (II-III)
[mm]			
r_0	0.547	-	0.500
r_d	0.600	0.547	0.547
r_s	0.610	0.585	0.556
r_s-r_d	0.010	0.039	0.009
[%]			
r_0/r_m	100	-	91.46
r_d/r_m	109.67	100	100
r_s/r_m	111.50	107.06	101.64

(b)

Figure 7.2: (a) Contours of the max. principal stress [Pa] at peak-systole. (b) Grid of the undeformed reference geometry, together with inner and outer contours of the zero-pressure (black), the end-diastolic (blue dotted line) and the peak-systolic (red dashed line) geometries. The inner radii of the corresponding contours, the radial end-diastolic to peak-systolic distension and the ratio of the different contours to the measured inner radius ($r_m = 0.547\,mm$) are tabulated. For more information about the different quadrants, the reader is referred to the text in Section 7.2.3.

7.3 Application of the backward displacement method: Abdominal aorta of the mouse

7.3.1 Structural model of an abdominal mouse aorta

In a second example, a more complex cardiovascular structure was created based on contrast-enhanced micro-CT images of the abdominal aorta of an in-house bred male ApoE -/- mouse on a C57BL/6 background (age: 5 months, body weight: 29 g). A mouse-specific 3D geometry of the aortic lumen containing four side branches was obtained *in vivo*, by segmentation of micro-CT (Triumph, Gamma Medica) images in Mimics (Materialise). In order to obtain sufficient contrast during the imaging process the mouse was intravenously injected with Aurovist (Nanoprobes), a contrast agent which provided satisfying results in earlier studies [76]. Using pyFormex, a hehexahedral grid was obtained using the mapping approach according to the method of De Santis et al. [18]. The mesh for the aortic wall consists of 80640 linear elements with 5 elements to represent the wall thickness, 48 elements in the circumferential direction and local refinements in the bifurcation regions, Figure 7.3(b). The wall thickness was assumed to be 10 percent of the local diameter and thus varies throughout the structure. The element type, the free radial displacement boundary condition at the ending cross sections (5.3) and the polynomial hyperelastic material model (7.1) were adopted from the first example (section 7.2.1).

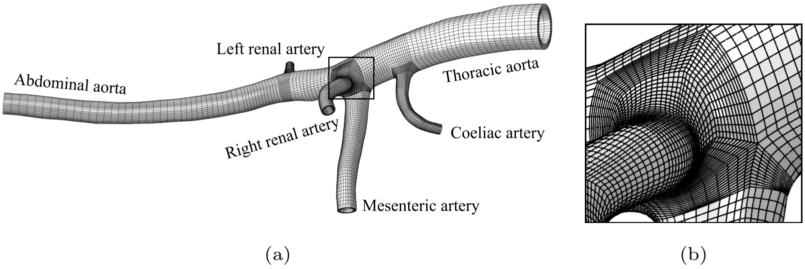


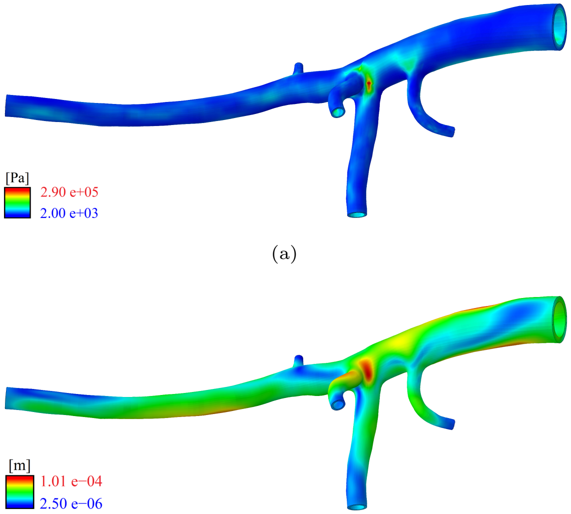
Figure 7.3: (a) Hexaedral mesh for the arterial wall of the abdominal aorta of a mouse and its side branches (obtained using the mapping approach), and (b) a detail of the mesh at the trifurcation region. The length of the sample was about 20 *mm*

7.3.2 In vivo stress incorporation in a mouse specific abdominal aorta

To further evaluate the backward displacement method this more complex geometry was brought to its zero-pressure state, assuming the internal pressure load at the moment of medical imaging to be 80 $mmHg$. Afterwards, *in vivo* stress was computed by reapplying this pressure load in a forward calculation.

The rate of convergence is plotted in Figure 7.1 for backward displacement simulations using both the fixed point and the quasi-Newton approach. In comparison with the fixed point convergence rate of the simplified artery in example 1, the logarithm of the relative residual now follows a linear decline after the second iteration. Although the geometry is much more complicated, a convergence criterion set at e.g. 10^{-4} would slow down the convergence only 3.33 (FP) and a 2.75 (QN-ILS) times. The quasi-Newton approach returns a converged solution twice as fast and thus shows an even more significant acceleration of the backward displacement method when using a complex geometry. Remark that the wall clock time of the overall calculation varies linearly with the number of iterations. The proportionality constant is the time required to perform one forward calculation.

Figure 7.4 depicts the contour plots of the maximum principal stress and the displacement field present when imposing the end-diastolic pressure (80 $mmHg$) onto inner wall of the restored zero-pressure geometry of the more complex cardiovascular structure.



(a)

(b)

Figure 7.4: Contours of (a) the max. principal stress [Pa], and (b) the displacement [m]. Both after applying the internal pressure load, present at the moment of medical imaging, onto the restored zero-pressure geometry.

7.3.3 In vivo based material parameter identification

When observing the simulated end-diastolic to peak-systolic distension d of Figure 7.2, which is defined by

$$d = \frac{D_{sys} - D_{dia}}{D_{dia}} \tag{7.2}$$

with D_{dia} and D_{sys} the inner diameters at end-diastole and peak-systole respectively, it is noticed that the previously assumed material parameters of the polynomial hyperelastic material model result in a mechanic material behaviour which is too stiff. Therefore, we wish to obtain material parameters which allow to mimic the distension of the arterial wall after the inclusion of the *in vivo* stress field, using *in vivo* obtained data only. Given the example of the abdominal mouse aorta, the distension can be measured *in vivo* using ultrasound (m-mode). As this distension is the result of the dynamic pressure load acting on the vessel wall, fitting a proper material model requires knowledge of this internal pressure [67]. To the best of our knowledge, correct and repeatable *in vivo* en non-invasive evaluation of the pressure waveform in the mouse aorta has not been reported yet. However, values for the diastolic blood pressure and the systolic blood pressure can be assessed by means of a tail-cuff system [58, 86].

As such, a new problem is defined: Find the material parameters ψ which result in the measured distension d_m for the corresponding internal pressure loads p_{dia} and p_{sys}, when the image-based geometry x_m and the corresponding internal pressure upon image acquisition p_m are known.

To solve this nonlinear problem, the objective function

$$f(\psi) = \left(\frac{d_s - d_m}{d_m}\right)^2 \tag{7.3}$$

is minimized using a standard minimization function in Matlab (Mathworks), with d_m and d_s the measured and simulated distension respectively. This optimization function utilizes a Sequential Quadratic Programming (SQP) method with a BFGS-update of the Hessian in each iteration.

To evaluate the objective function in iteration i, the backward displacement method is performed first, given the image-based geometry x_m, the internal pressure load present at the moment of imaging p_m, and the new guess for the material parameters ψ^i, which is provided by the nonlinear optimization function in the previous iteration or by an initial guess in the first iteration.

$$\tilde{\boldsymbol{X}}^i = \mathcal{B}(\mathcal{S}_U(\boldsymbol{X}^i)) \tag{7.4}$$

with $\mathcal{B}$ the backward displacement function, Equation (6.1), and $\mathcal{S}_{\boldsymbol{U}}$ the displacement resulting from structural solver, Equation (6.2),

$$\tilde{\boldsymbol{X}}^i = \mathcal{B}(\boldsymbol{U}) = \boldsymbol{X}_m - \boldsymbol{U} \tag{7.5}$$

$$\boldsymbol{U} = \mathcal{S}_U(\boldsymbol{X}) = \mathcal{S}(\Omega(\boldsymbol{X}^i, \boldsymbol{0}), p_m, \psi^i) - \boldsymbol{X}^i \tag{7.6}$$

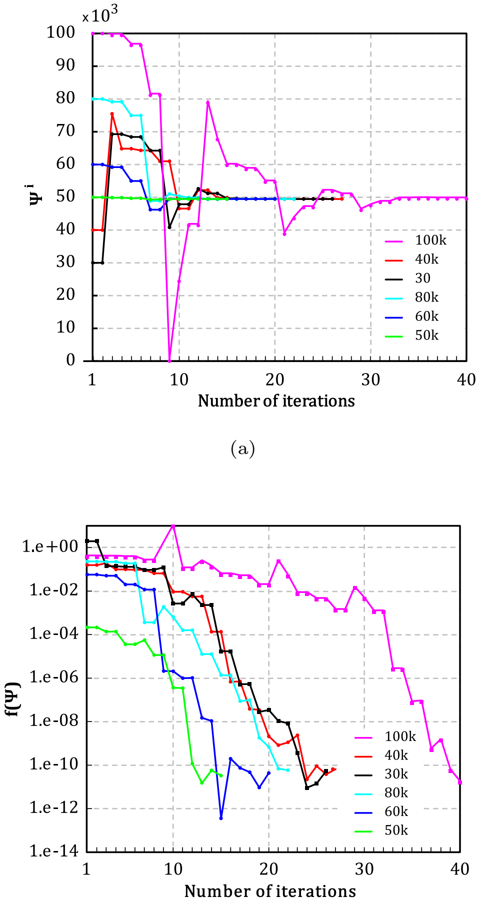
In contrast to Section 6.1, the arguments for the structural solver $\mathcal{S}$ now explicitly state ψ^i to denote the dependency of the material parameters which are updated in every optimization iteration. Upon convergence, $\boldsymbol{X}^{i,*} = \tilde{\boldsymbol{X}}^i = \boldsymbol{X}^i$ is met, with $\boldsymbol{X}^{i,*}$ the zero pressure geometry. Then, forward analyses are performed to bring the zero pressure geometry in its end-diastolic state (p_{dia}), from which it is further inflated to peak-systole (p_{sys}). As such, the end-diastolic and peak-systolic geometries are obtained,

$$\boldsymbol{x}^i_{dia} = \mathcal{S}(\Omega(\boldsymbol{X}^{i,*}, \boldsymbol{0}), p_{dia}, \psi^i) \tag{7.7}$$

$$\boldsymbol{x}^i_{sys} = \mathcal{S}(\Omega(\boldsymbol{X}^{i,*}, \boldsymbol{0}), p_{sys}, \psi^i) \tag{7.8}$$

from which the inner diameters , D^i_{dia} and D^i_{sys}, can be computed. Finally, the simulated distension d_s is obtained using Equation (7.2) and compared to the measured one d_m, in order to evaluate the objective function, Equation (7.3). Using this information, the nonlinear optimization function computes a new guess ψ^{i+1} for the material parameters. Upon convergence of the optimization procedure, $\psi^* = \psi^{i+1}$ represent the material parameters which result in the desired distension when applying the pulse pressure in a model in which the diastolic pressure is in equilibrium with the image-based and prestressed configuration $\Omega(\boldsymbol{x_m}, \boldsymbol{\sigma}^*)$.

Example 7.3.1. As an example, a Neo-Hookean material model – i.e. only $C_{10} \neq 0$ in Equation (7.1) – is fitted for a simplified model of an abdominal mouse aorta using the methodology described in this section. Therefore, a cylindrical segment is modelled with a length of $1\,mm$, an inner diameter of $0.32\,mm$ and a wall thickness to diameter ratio of 0.1. A quadratic mesh with hybrid, hexahedral elements was applied with 3 layers in the transversal direction to represent the wall thickness, 32 elements in the circumferential direction and 4 elements along the axis of symmetry. The distension was assumed to be $10\,\%$ of the inner diameter in the end-diastolic ($60\,mmHg$) to peak-systolic ($90\,mmHg$) pressure range. The BDM-based optimization procedure is preformed for $p_m = p_{dia}$ and a converged result ($C_{10} = 49503.6$) is obtained regardless of the initially provided guess for the material parameter (Figure 7.5).



(a)

(b)

Figure 7.5: Evolution of (a) the material constant and (b) the objective function, Equation (7.3), throughout the BDM-based material parameter optimization procedure.

7.4 Conclusion

In this chapter, two examples are used to demonstrate the backward displacement method. Although the structural solver was a black box, implementation was straightforward as only an update of the grid was required in every iteration. First, a simplified model of a small artery (represented by a straight cylindrical tube) was studied for validation purpose and to evaluate the importance of *in vivo* stress incorporation in an image-based geometry. In this regard, Figure 7.2 clearly showed the overestimation of the distension when not incorporating prestress. The second example concerned a mouse-specific abdominal aorta with four side branches to explore the ability of the backward displacement method to restore a more complex cardiovascular structure at its zero-pressure state. Compared to a straight cylindrical tube, only a relatively low number of additional iterations was needed to incorporate *in vivo* stress into a much more complex geometry, Figure 7.1. Moreover, the same figure showed that the quasi-Newton technique increased the convergence rate significantly, compared to the fixed point approach. Finally, it is demonstrated that the backward displacement method can be applied in a parameter optimization framework to solve for the unknown material parameters using *in vivo* obtainable data only.

Three

From medical image to computational FSI model

CHAPTER 8

Numerical modelling of an abdominal mouse aorta

8.1 Introduction

This chapter explains the development of a numerical model to simulate the fluid flow in an image-based abdominal mouse aorta, and to simulate the deformation of the arterial wall as well as the corresponding stress distribution throughout the cardiac cycle. The choice for the case of an abdominal mouse aorta is motivated by its clinical relevance, as pointed out below, the in-house availability of mouse-specific datasets obtained by Trachet et al. [75, 76] and the access to the small animal imaging facilities of Infinity (the preclinical imaging research facility of Ghent University).

In the last few years there is an increasing interest in numerical models of abdominal aortic aneurysms (AAA) to:

- better understand the growth and development of the aneurysm;
- support diagnosis through the assessment of its rupture potential.

In order to verify these simulations, validation is an important but difficult task. Given the ethical constraints, the slow time course of the disease in humans and the absence of true baseline data (i.e. the data obtained before the development of the disease in a healthy aorta), these validating studies are difficult to perform in humans. This is particularly true for aneurysm rupture research, as rupture will normally be prevented by surgery for the sake of the patient and access to post-mortem tissue is not always possible.

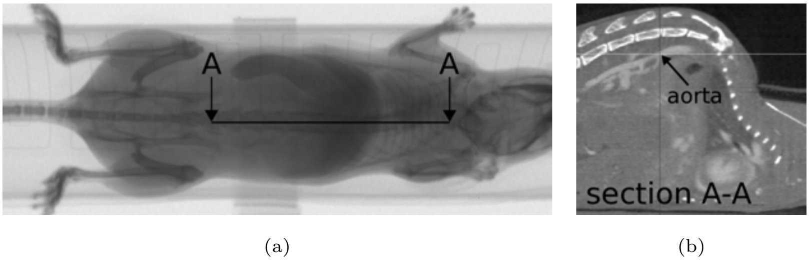
In order to overcome these issues when dealing with human related studies, aneurysm developing mouse-models can be analyzed instead. Previous research at IBiTech - BioMMeda investigated the role of disturbed hemodynamics in aneurysm formation in mice [74]. Therefore, Trachet et al. proposed an experimental-computational framework which combined flow velocity measurements (based on high-frequency ultrasound) with the assessment of the geometry for the fluid domain (based on contrast-enhanced micro-CT) to set up mouse-specific computational fluid dynamics (CFD) simulations of the hemodynamic situation in the abdominal mouse aorta, only based on *in vivo* data [75–77].

Limitations of this modelling framework were the time required by an experienced operator to perform the preprocessing on the one hand and the assumption of rigid walls by using a CFD model on the other hand. Regarding the preprocessing (from medical image to computational model), the in Part I introduced extended Treemesh method (XTM) allows for a speed up of the meshing phase by providing automated hexahedral mesh generation (for both the fluid domain and the arterial wall). Moreover, the software implementation provides a graphical user interface which guides the operator through the meshing procedure and allows for a straightforward (local) grid refinement and the inclusion of aneurysm structures, as was demonstrated in Figures 4.5 and 2.14. Regarding the compliance of the arterial wall, Part III introduces the fluid-structure interaction (FSI) between the blood flow and the vessel wall in the abdominal mouse aorta. Consequently, a redistribution of the measured flows over the side branches [75] – to meet the criterion of zero mass imbalance at each time step of the CFD simulation – is not a necessity anymore due to the buffering capacity of the vessel wall. Moreover, this allows to study the stress distribution throughout the arterial wall, which is of importance when one wants to asses e.g. the rupture potential of aneurysms. To accurately quantify these stresses in an *in vivo* obtained image-based geometry, the incorporation of the *in vivo* stress state is indispensable. In this regard, the in Part II introduced backward displacement method (BDM) allows for a straightforward inclusion of the *in vivo* stress field resulting from the internal blood pressure, which is present at the moment of image acquisition.

The aim of this chapter is thus to set up a fluid-structure interaction simulation of an abdominal mouse aorta, based on *in vivo* measurements. Therefore, the modelling aspects covered in Part I and Part II of this work were included, what demonstrates their applicability when modelling the fluid-structure interaction in complex vascular structures. First, the *in vivo* data acquisition is discussed (Section 8.2) along with the application of the extended Treemesh method which results in a computational grid (Section 8.3). Then, a CFD model is constructed for which the sensitivity of the flow results on the grid density is analyzed and compared to results obtained with tetrahedral grids (Section 8.4). Next, a computational structural mechanics (CSM) model for the arterial wall is set up and prestressed using the backward displacement method. This is done for material parameters which were identified such that the *in vivo* measured distension was obtained in the physiological pressure range (Section 8.5). Finally, the flow model and the structural model (including prestress) are coupled by solving the FSI problem using a partitioned approach and the resulting pressure drop, flow field, deformation and stress field are discussed (Section 8.6).

8.2 In vivo measurements: Geometry and flows

The geometrical model of the abdominal mouse aorta was obtained from an inhouse bred ApoE -/- mouse on a C57Bl/6 background (age: 37 weeks, body weight 30.6 g), which corresponds to case AA7 in [75]. Once anesthetized (1.5 % Isoflurane), the mouse was intravenously injected with Aurovist (Nanoprobes) to improve the contrast of the medical images, Figure 8.1, which were obtained by a micro-CT scanner dedicated for small animal imaging (GE FLEX Triumph, Gamma Medica-Ideas). The reconstructed images were converted into DICOM standard format with a voxelsize of 75 μm and imported into the software package Mimics (Materialise) to perform the segmentation process, in which a 3D



(a) (b)

Figure 8.1: (a) Overview of the micro-CT scanning area and (b) detailed view of the slice (A - A) in the reconstructed data set, denoting the aorta.

geometry is extracted from a stack of 2D images. Although contrast-enhanced images were being used, manual intervention was still required to separate the aorta from the venous segments. As such, semi-automated segmentation resulted in a three-dimensional (3D) reconstruction of the complete aortic lumen: from the ascending aorta to the distal abdominal aorta, including the major side branches. In the rest of this chapter, we will solely focus on the abdominal part, for which the aortic wall was modelled in pyFormex and presented in red in Figure 8.2. The thickness of the arterial wall was assumed to be 10 % of the inner diameter, a reasonable assumption supported by findings obtained in an earlier study [77].

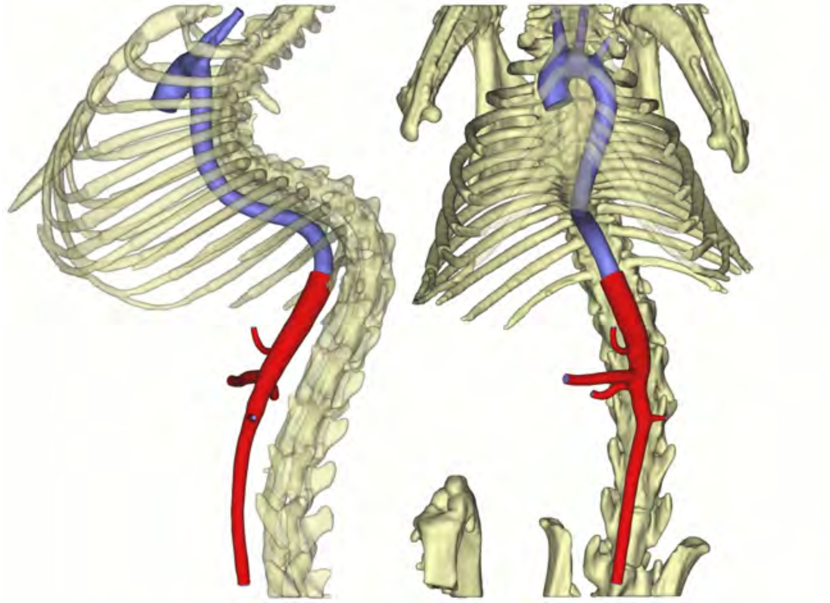


Figure 8.2: Geometry of the lumen (blue) resulting from the segmentation process of the micro-CT images using Mimics (Materialise). The arterial wall (red) is constructed for the region of interest using pyFormex.

After the micro-CT scan, anesthesia was maintained and mice immediately underwent an ultrasound (US) scan. Ultrasound data were obtained with a high-frequency ultrasound apparatus (Vevo 2100, VisualSonics). All measurements were performed by an experienced operator. The animal was secured on the table in supine position while ECG, respiratory rate and body temperature were monitored. First, B-mode imaging was used followed by color Doppler imaging to locate side branches, and pulsed Doppler to assess the flow velocities in the abdominal aorta and in its side branches, see Figure 8.3. The following abbreviations are introduced to denote the different in- and outlets: thoracic aorta (TA), celiac artery (CA), right renal artery (RRA), mesenteric artery (MA), left renal artery (LRA), distal abdominal aorta (AA). To enable correlation of flow velocity measurements with cross sections obtained by micro-CT, the distance from the location of each aortic flow velocity measurement to the nearest bifurcation was measured along the aorta. The corresponding (constant) areas of the image-based geometry, $A_{m,i}$, were used to approximate the measured flow $Q_{m,i}(t)$ at each of the in- and outlets $i = \{\text{TA, CA, RRA, MA, LRA, AA}\}$ by

$$Q_{m,i}(t) = \frac{1}{2} v_{m,i}(t) A_{m,i} \tag{8.1}$$

as the maximum velocity components ($v_{m,i}$) of parabolic flow profiles were measured. These flows are represented in Figure 8.4 for the mean heart beat rate of 359.3 bpm what corresponds to 0.1670 s per cardiac cycle. To obtain a time-averaged mass imbalance equal to zero, a scaling of the measured flows has to be applied. Scaling the inflow requires only one factor and minimizes this factor. As such, the thoracic inlet flow is scaled with an imbalance correction factor f_c of 1.07. This can be justified by the fact that it is more likely to underestimate the velocity measured by ultrasound than to overestimate it.

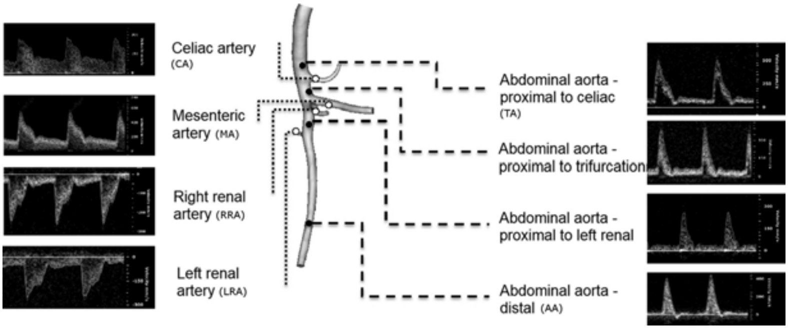

Celiac artery (CA)
Mesenteric artery (MA)
Right renal artery (RRA)
Left renal artery (LRA)
Abdominal aorta - proximal to celiac (TA)
Abdominal aorta - proximal to trifurcation
Abdominal aorta - proximal to left renal
Abdominal aorta - distal (AA)

Figure 8.3: Pulsed Doppler ultrasound flow velocity measurements as they were acquired throughout the abdominal aorta. Note that the scaling of the velocity waveforms is not the same. Figure adapted from [75].

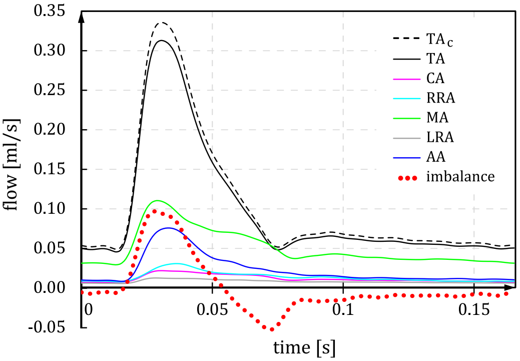
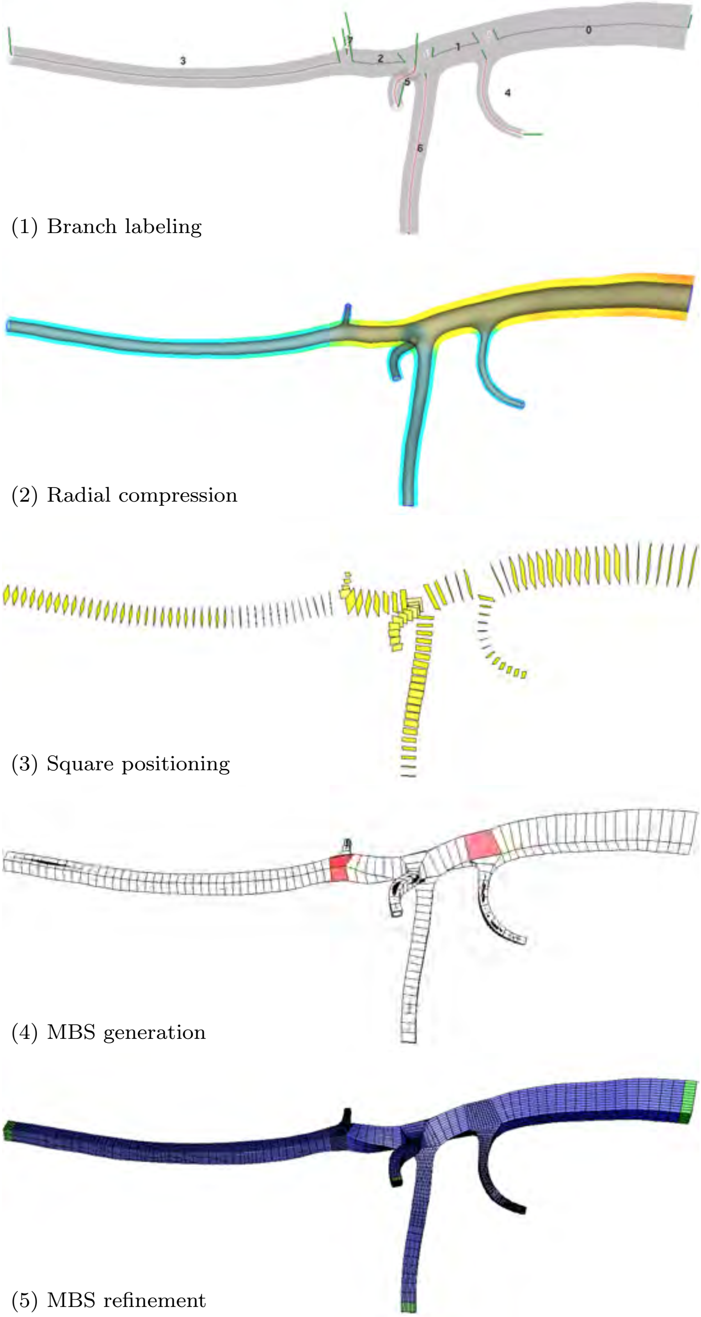


Figure 8.4: Flows obtained by scaling the ultrasound flow velocity measurements with respect to the location of the measurement. The black dashed line represents the corrected thoracic inflow to obtain a time-averaged mass imbalance equal to zero. The red dotted line shows the corrected mass imbalance.

8.3 Mesh generation

After segmentation, the outer surface of the lumen geometry is represented by a triangulated surface mesh and exported into the stereolithographic (STL) file format. This geometry file is subsequently passed on to the extended Treemesh method, which was implemented in a pyFormex-based meshing framework. To reduce the computation time of the final FSI simulation, the computational domain was limited by restricting the length of the in- and outlet branches resulting from the segmentation process. A stepwise software wizard provides parameter dialog boxes which guide the – even non-expert – user through the XTM meshing procedure. The intermediate results are presented in Figure 8.5. Combining this straightforward parameter modification with the possibility of direct visual feedback by means of a graphical user interface (GUI) speeds up non-default manipulations such as local grid refinements. These were introduced to obtain higher aspect ratios towards the in- and outlets than in the bifurcations regions. Therefore, a set of axial edge refinement factors was defined $\boldsymbol{f}_{edge,a} = [1, 1, 1, 1, 3, [3, 2], [2, 1], 3]$, based on the defaults (Equation (2.6) and Equation (2.8)). Once defined, several grids were auto-generated to assess the grid dependency of the simulation results, see Section 8.4.2. For the global grid refinement of the abdominal aorta the radial compression ratio and the square scaling factor were kept constant as well (f_R=0.6 and f_s=0.75), and this for different combinations of d_0, $f_{edge,t}$ and f_{BL}, Table 8.1. To recapitulate, these global refinement parameters were defined as follows

- d_0: The distance between the centerline nodes, which provides a measure



(1) Branch labeling

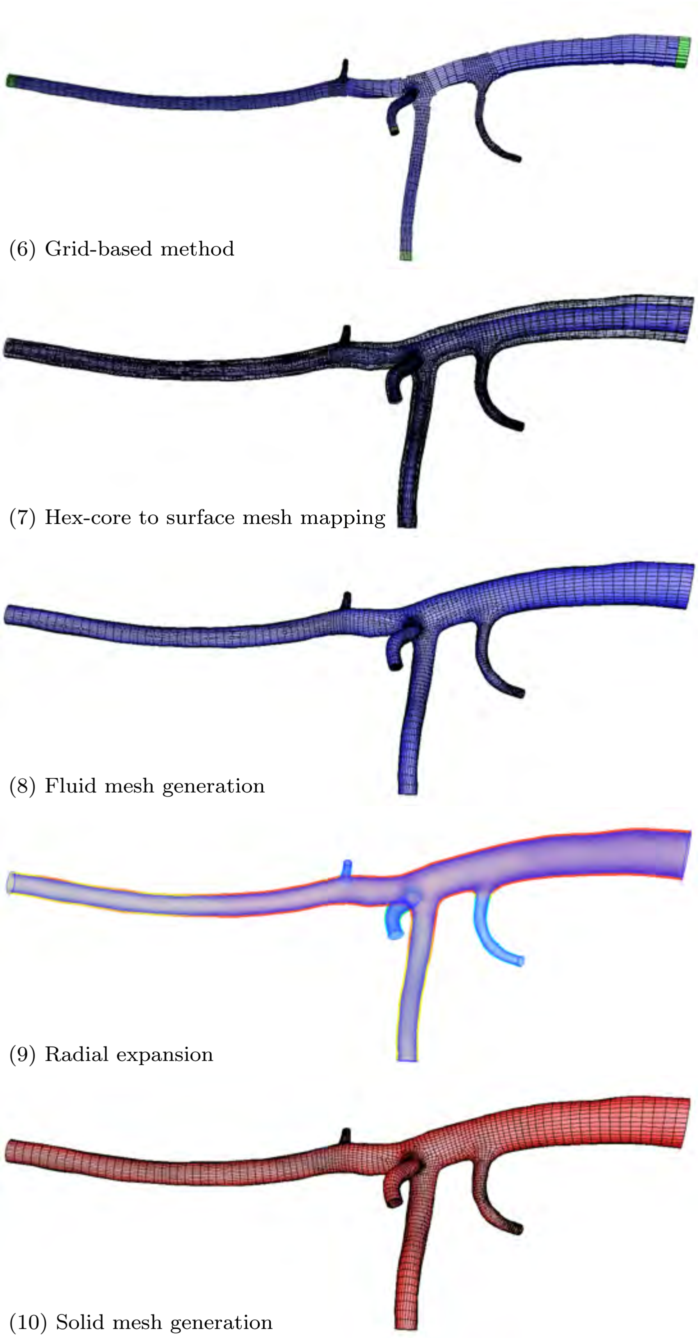
(2) Radial compression

(3) Square positioning

(4) MBS generation

(5) MBS refinement

Figure 8.5: Overview of the subsequent steps in the XTM mesh generation for the abdominal mouse aorta (cont'd).



(6) Grid-based method

(7) Hex-core to surface mesh mapping

(8) Fluid mesh generation

(9) Radial expansion

(10) Solid mesh generation

Figure 8.5: Overview of the subsequent steps in the XTM mesh generation for the abdominal mouse aorta.

for the maximum edge size of the cells at the hexahedral core.

- f_R: The radial compression ratio, which defines the relative distance between the hex-core and the outer surface mesh.
- f_s: The square scaling factor, which sets the edge lengths of the vessel enclosing squares ($f_s > f_R$), relative to the local diameter of the maximum inscribed sphere in the original geometry.
- $f_{edge,t}$: The transversal edge refinement, which sets the number of subdivisions for each side of each square.
- f_{BL}: The number of uniformly distributed cells between the hex-core and the outer surface mesh, which defines the boundary layer grid.

# cells	$d_0[mm]$	$f_{edge,t}$	f_{BL}
53 k	0.250	11	5
98 k	0.200	13	6
201 k	0.143	17	7
413 k	0.122	21	10
798 k	0.096	26	12
1602 k	0.077	33	15
3216 k	0.061	41	19

Table 8.1: Grid refinement parameters and resulting number of cells in the fluid grid

8.4 Computational fluid dynamics model

In this section the numerical model for the fluid domain is discussed. In Section 8.4.1, the solver settings and boundary conditions are highlighted. Whereas in Section 8.4.2, steady state CFD simulations are performed to assess the grid sensitivity of the pressure drop, the flow and the wall shear stress. This last subsection includes the comparison with results obtained for tetrahedral grids.

8.4.1 Solving the flow problem

The equations that describe the unsteady flow of an incompressible, Newtonian fluid are the conservation of mass and the Navier-Stokes equations, given by

$$\nabla.\boldsymbol{v}_f = 0 \tag{8.2a}$$

$$\rho_f \frac{\partial \boldsymbol{v}_f}{\partial t} + \rho_f \boldsymbol{v}_f.\nabla(\boldsymbol{v}_f) = -\nabla p + \nabla.\boldsymbol{\tau} + \boldsymbol{f}_f \tag{8.2b}$$

with $\boldsymbol{v}_f$ the flow velocity, ρ_f the fluid density, t the time, p the pressure, $\boldsymbol{\tau}$ the viscosity stress tensor and $\boldsymbol{f}_f$ the body forces on the fluid per unit of volume.

Solver settings. All flow simulations have been performed with the commercial finite volume CFD solver Fluent 14.5 (Ansys). Given the satisfying results in earlier studies (Chapters 3 and 4), the pressure and velocity fields were solved using the SIMPLE algorithm with second-order upwind discretization for the momentum equation and standard pressure interpolation (i.e. according to the momentum equation coefficients). For the transient solutions in Section 8.6, an implicit time integration scheme was used with second-order accurate temporal discretization. Convergence was obtained when the scaled residuals (i.e. the residuals scaled relative to the largest absolute value of the residuals obtained in the first five iterations) of continuity and momentum decreased below a value of 10^{-6}.

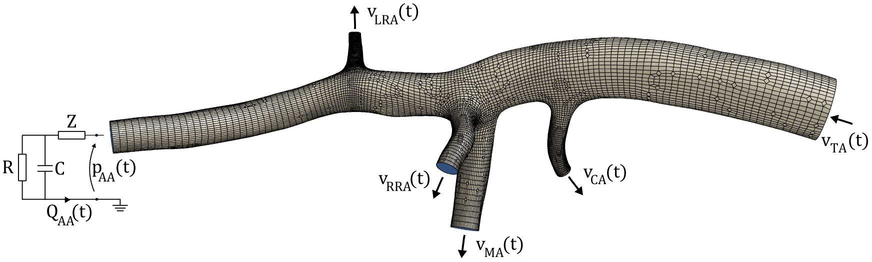


Figure 8.6: Fluid domain denoting the boundary conditions: a velocity inlet at the thoracic artery (TA), velocity outlets at the celiac artery (CA), right renal artery (RRA), mesenteric artery (MA) and left renal artery (LRA) and a windkessel pressure outlet at the distal abdominal aorta (AA).

Boundary conditions. Time-varying uniform velocity profiles were imposed normal to the boundaries at the inlet of the aorta (TA) and the outlets of the side branches (CA, RRA, MA, RLA), based on the measurements of the maximum velocity components $v_{m,i}(t)$. As these measurements were not taken at the exact same locations as the ending cross sections of our fluid domain, $A_{f,i}$, we account for this difference in cross sectional area using Equation (8.3).

$$v_{TA}(t) = \frac{1}{2} v_{m,TA}(t)\, \frac{A_{m,TA}}{A_{f,TA}}\, f_c \tag{8.3a}$$

$$v_i(t) = \frac{1}{2} v_{m,i}(t)\, \frac{A_{m,i}}{A_{f,i}} \qquad i = \text{CA, RRA, MA, LRA} \tag{8.3b}$$

Note that a multiplication factor f_c of 1.07 was included in Equation (8.3a) to correct for the non-zero time-averaged mass imbalance, and that the cross sectional areas were not variable what will result in higher flows for the FSI simulation when compared with the corresponding CFD simulation. A pressure outlet boundary condition was prescribed at the distal abdominal aorta (AA) by means of a three element windkessel model which applies a static pressure at the outlet, $p_{AA}(t)$, given the flow through the outlet, $Q_{AA}(t)$. The corresponding electrical circuit is represented in Figure 8.6 where Z represents the characteristic impedance, R the hydraulic resistance and C the wall compliance. The parameters of the model were defined such that physiological pressure variations were retrieved when applying the measured flow rate at the distal abdominal aorta. ($Z = 367.1\,\frac{mmHg}{ml/s}$, $R = 2390.8\,\frac{mmHg}{ml/s}$, $C = 6.9\,10^{-5}\,\frac{ml}{mmHg}$). At the walls the no-slip condition was imposed. Regarding the fluid parameters, the density was set to $1060\,\frac{kg}{m^3}$ and the blood was modeled as a Newtonian fluid with a constant dynamic viscosity of $0.0035\,Pa\,s$.

8.4.2 Grid sensitivity analysis[1]

In the subsequent section, the sensitivity of the fluid mesh on the flow variables is studied by gradually increasing the number of cells until the solution no longer changes upon a further increase in number of cells. As there is a trade-off between the accuracy of the solution and the number of cells in the simulation (and thus the simulation time), it is important to analyze this grid dependency. As a result, huge computation times can be avoided by limiting the desired number of cells in the fluid domain to this value that provides the desired resolution of the flow variable(s) of interest. In contrast to the solver settings and boundary conditions provided in the previous section, the grid sensitivity analysis was performed for a steady state CFD simulation, and this for the complete image-based geometry. At the inlet of the fluid domain a plug flow profile was specified for the flow measured at peak systole, $Q_{TA,sys} = 0.312\,ml/s$. Compared to imposing a fully developed inlet profile, the downstream effect of imposing a plug profile instead was negligible, as reported in an earlier CFD study in which the comparison with a whole aorta was presented [75]. At the outlets, flow fractions of the total outflow were specified, as listed in Table 8.2, based on the mean outflow fractions resulting from the same study.

Along the centerline of the aorta, the error in the total pressure drop Δp and the difference Δv between the maximum and minimum velocity were analyzed for multiple hexahedral grids with increasing mesh densities, see Table 8.1, relative to their respective values obtained for the finest grid, i.e. the reference grid with $\Delta p_{ref} = 6.822\,mmHg$ and $\Delta v_{ref} = 32.310\,cm/s$.

[1] This section is based on material presented in the article *"Unstructured hexahedral mesh generation of complex vascular trees using a multi-block grid-based approach."*, submitted to the Journal *Computer Methods in Biomechanics and Biomedical Engineering*, 2014.

Branch		flow fractions [%]	flow [ml/s]
Thoracic aorta	(TA)	100.00	0.3120
Coeliac artery	(CA)	13.30	-0.0415
Right renal artery	(RRA)	33.33	-0.1040
Mesenteric artery	(MA)	15.12	-0.0472
Left renal artery	(LRA)	11.29	-0.0352
(Distal) abdominal aorta	(AA)	26.96	-0.0841

Table 8.2: The mean outflow fractions resulting from a study involving ten animals [75] and the corresponding flows [ml/s] at peak systole for each in- and outlet of the abdominal aorta and its side branches. These values were imposed as boundary bonditions during the CFD grid sensitivity analysis (Section 8.4.2).

$$e_{\Delta p} = \frac{|\Delta p - \Delta p_{ref}|}{\Delta p_{ref}} \tag{8.4}$$

$$e_{\Delta v} = \frac{|\Delta v - \Delta v_{ref}|}{\Delta v_{ref}} \tag{8.5}$$

These errors are reported in Table 8.3, together with the mean and the maximum error of the pressure and velocity with respect to all nodes n of the same centerline,

$$e_p(n) = \frac{|p(n) - p_{ref}(n)|}{\Delta p_{ref}} \tag{8.6}$$

$$e_v(n) = \frac{|v(n) - v_{ref}(n)|}{\Delta v_{ref}} \tag{8.7}$$

To allow an adequate comparison for different grids, the wall shear stress (WSS) is evaluated on a global level by considering the difference between the surface area enclosed by a WSS iso-contour A_{iso} and the corresponding value for the reference grid, normalized by the total surface area, A_{tot}. This results in the following error function,

$$e_{wss}(wss) = \frac{|A_{iso}(wss) - A_{iso,ref}(wss)|}{A_{tot}} \tag{8.8}$$

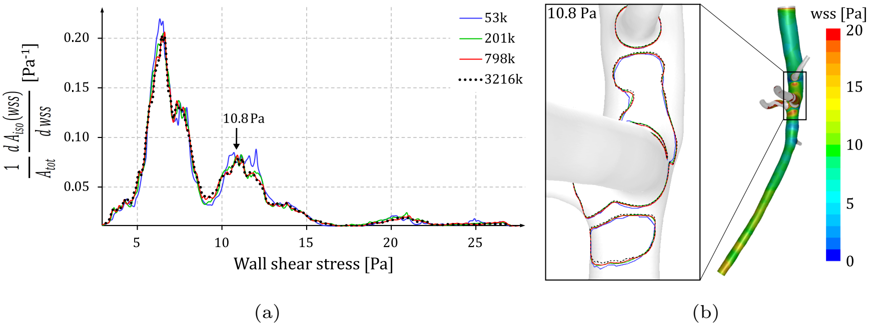
The mean and maximum errors with respect to the range of wall shear stress values (wss) are presented in Table 8.3 as well. As the errors are small, the derivative of this function is visualized in Figure 8.7(a) in function of WSS and for values larger than 5 % of the maximum surface area change. To evaluate WSS on a local level the iso-contour of 10.8 Pa, which corresponds to both a high WSS value and a high change in WSS surface area, is visualized in Figure 8.7(b) for different mesh densities in the region distal to the celiac artery and proximal to the trifurcation.

Table 8.3 shows small errors for all three flow variables and a converging trend. 53 k, 98 k and 798 k cells are required in the fluid domain to obtain a maximum

#	Pressure drop along centerline				Velocity along centerline				Wall shear stress area	
Fluid cells	Δp [mmHg]	$e_{\Delta p}$ [%]	$e_{mean,p}$ [%]	$e_{max,p}$ [%]	Δv [cm/s]	$e_{\Delta v}$ [%]	$e_{mean,v}$ [%]	$e_{max,v}$ [%]	$e_{mean,wss}$ [%]	$e_{max,wss}$ [%]
53 k	6.877	0.795	0.151	0.795	30.528	5.52	2.028	6.101	0.268	1.728
98 k	6.853	0.443	0.171	0.443	31.106	3.73	1.460	5.096	0.193	1.088
201 k	6.844	0.319	0.0536	0.319	31.696	1.90	0.825	3.174	0.138	0.827
413 k	6.836	0.204	0.0318	0.204	31.960	1.08	0.476	2.269	0.100	0.693
798 k	6.827	0.0638	0.0359	0.107	32.206	0.324	0.186	1.090	0.0629	0.437
1602 k	6.826	0.0449	0.00875	0.0449	32.267	0.134	0.0977	0.428	0.0323	0.265
3216 k	6.822				32.310					

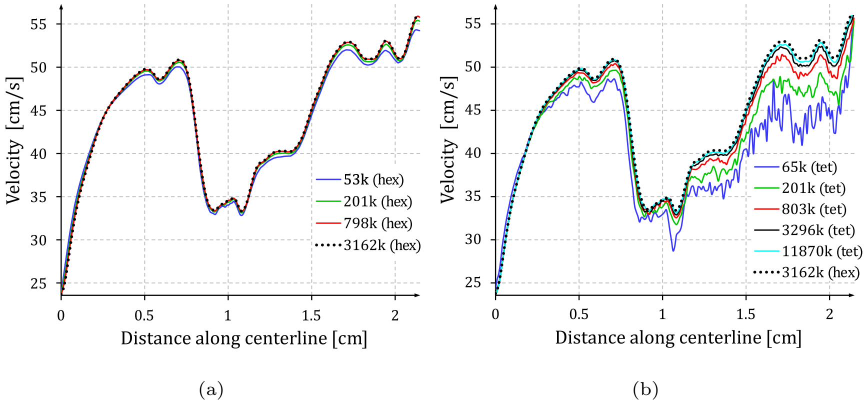
Table 8.3: Mean and maximum values of the pressure, velocity and the wall shear stress error variables for different grid densities. For maximum errors of approximately 1 % in the pressure, velocity and wall schear stress, respectively 53 k, 798 k and 98 k cells are required.

error of about 1 % for respectively the pressure along the centerline, the global change in WSS surface area and the velocity along the centerline. Moreover, Figure 8.7(b) shows only a small error and a converging trend for the WSS surface area on a local level. The efficacy of hexahedral grids becomes clear when comparing the velocity along the centerline to results obtained for a grid sensitivity analysis involving tetrahedral grids (with a prismatic boundary layer grid). Figure 8.8 demonstrates that 201 k hexahedral cells provide the same accuracy as 11870k tetrahedral cells. As a compromise between accuracy and computation time, the grid refinement parameters resulting in a fluid grid of 201 k cells are selected for the FSI simulation.



(a) (b)

Figure 8.7: (a) Change in enclosed surface area in function of the wall shear stress for hexahedral grids with different mesh densities. (b) Contour plot of the wall shear stress (right) and iso-contours (left) at the trifurcation region (for the peak in wall shear stress change at 10.8 Pa and for different mesh densities)



(a) (b)

Figure 8.8: (a) The velocity along the centerline of the abdominal mouse aorta for hexahedral grids with different mesh densities and (b) compared to tetrahedral grids with different mesh densities.

8.5 Computational structural mechanics model

In this section the numerical model for the arterial wall is discussed. In Section 8.5.1, the solver settings and boundary conditions are highlighted. Section 8.5.2 focuses on the application of a proper material model and the corresponding parameters in order to obtain the *in vivo* measured distension of the arterial wall. The resulting material model is applied in Section 8.5.3 to compute the *in vivo* stress field present at the moment of medical imaging.

8.5.1 Solving the structural problem

The velocity of the structure $\boldsymbol{v}_s$ is determined by the Cauchy momentum equation that describes the non-relativistic momentum transport in any continuum.

$$\rho_s \frac{D\boldsymbol{v}_s}{Dt} - \nabla.\boldsymbol{\sigma}_s = \boldsymbol{f}_s \tag{8.9}$$

In this equation ρ_s represents the density of the solid, $\boldsymbol{\sigma}_s$ the Cauchy stress tensor and $\boldsymbol{f}_s$ the body forces per unit of volume on the structure. The notation D/Dt refers to the material derivative.

Solver settings. All computational structural mechanics (CSM) simulations have been performed with the commercial finite element solver Abaqus/Standard (Simulia). All simulations take into account geometric nonlinearities due to large deformations. In the backward displacement method, subsequent static analyses were performed, whereas the transient simulations in Section 8.6 made use of an implicit (backward-Euler) time integration scheme. As demonstrated in Section 8.3 and explained in Section 2.2.4, a computational grid was generated for the arterial wall, based on the geometry of the lumen (extracted from a micro-CT scan) and a wall thickness to inner diameter ratio of 10%, Section 8.2. The arterial wall was modeled using quadratic hexahedral elements with a hybrid formulation. $44\,k$ of these elements (with three layers in the transversal direction to model the wall thickness) were required to obtain sufficiently converged results when applying pressures in the physiological pressure range.

Boundary conditions. Local cylindrical coordinate systems were defined at the in- and outlets. At the corresponding ending cross sections, displacement boundary conditions were applied allowing radial displacement of the nodes only, except for the shortest side branch (left renal artery) where in plane movement of the nodes was allowed too. Loads were only imposed at the inner surface of the vessel wall. Both the normal component and the shear component of the traction

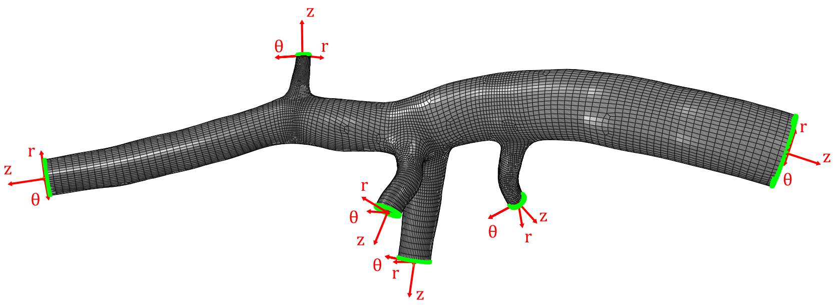


Figure 8.9: The solid model denoting the local cylindrical coordinate systems (red) and the nodes at which the displacement boundary conditions are applied (green).

vector were applied at the fluid-structure interface, whereas the backward displacement method only assumed the end-diastolic pressure at the inner wall. An overview of the structural model is shown in Figure 8.9. Definition of the node sets, the local coordinate systems and the boundary conditions was integrated in the XTM-framework to speed up the generation of the computational model.

8.5.2 In vivo-based material parameter identification

The nonlinear mechanical behaviour of incompressible arterial tissue is typically described using a hyperelastic material model. Towards the fluid-structure interaction approach, a correct representation of the vessel wall is indispensable. Therefore, our main interest is obtaining material parameters which allow to mimic the distension of the arterial wall after the inclusion of the *in vivo* stress field. This distension can be measured *in vivo* using ultrasound (M-mode), and is the result of the end-diastolic to peak-systolic pressure variation which can be obtained using *in vivo* tail-cuff measurements [58, 86]. A BDM-based material parameter optimization framework (BDMPO), in which a simplified model of an artery is repeatedly prestressed and inflated using static analyses until the distension criterion is met, is discussed in Section 7.3.3.

For an image-based geometry of an abdominal mouse (triggered under anesthetic and end-diastole conditions), the end-diastolic and peak-systolic blood pressures are assumed to be $p_{dia} = 60\,mmHg$ and $p_{sys} = 90\,mmHg$ respectively [91], and a corresponding – *in vivo* obtained – distension of $d = 10\%$ is considered. The BDMPO procedure is applied on a simplified model of a thick-walled cylindrical tube with an inner diameter of $0.32\,mm$, which corresponds to the image-based diameter at the outlet of the distal abdominal aorta. As discussed in Example 7.3.1, a single parameter Neo-Hookean material model can be fitted, however, the lack of material stiffening at higher strains and the absence of surrounding tissue in the CSM model, made the Neo-Hookean material model unsuitable for the final FSI simulations. Recent studies describe more sophisticated material models (including the stiffening effect and the anisotropic behaviour for the different layers of the arterial wall), however, fitting them to this limited amount of *in vivo* data would over parameterize the problem [78]. Therefore, the (incompressible) Arruda-Boyce model [6] has been applied, with a strain energy density function defined by Equation (8.10)

$$W = \mu \sum_{i=1}^{5} \frac{\alpha_i}{\lambda_m^{2i-2}} (I_1^i - 3^i) \tag{8.10a}$$

$$\{\alpha_1 = \frac{1}{2}, \alpha_2 = \frac{1}{20}, \alpha_3 = \frac{11}{1050}, \alpha_4 = \frac{19}{7000}, \alpha_5 = \frac{519}{673750}\} \tag{8.10b}$$

The model can be seen as a fifth-order reduced polynomial, in which the five coefficients (C_{10},...,C_{50}) are nonlinear functions of only two parameters μ and λ_m. Equation (8.11) relates μ to the initial shear modulus μ_0, and λ_m represents the locking stretch, which approximately denotes the stretch at which the slope of the stress-strain curve will rise significantly.

$$\mu_0 = \mu\big(1 + \frac{3}{5\lambda_m^2} + \frac{99}{175\lambda_m^4} + \frac{513}{875\lambda_m^6} + \frac{42039}{67375\lambda_m^8}\big) \tag{8.11}$$

As such, the Arruda-Boyce model is able to offer the desired effect of material stiffening and this for only two material parameters. Moreover, the model allows for a physical interpretation, what the general fifth-order reduced polynomial fails to provide.

To provide sufficient stiffening in the physiological pressure range, a value of $\lambda_m = 1.01$ is considered. The BDMPO procedure was then performed to find the unknown material parameter $\mu^* = 24358\,Pa$, given an initial guess ($\mu^{i=1} = 30367\,Pa$) which was based on Equation (8.11) and the initial shear modulus μ_0 resulting from the in Example (7.3.1) obtained material constant $C_{10} = \frac{\mu_0}{2} = 49504\,Pa$ for the Neo-Hookean material model. Although a simplified model allows for a straightforward inclusion of axial prestretch and residual stresses, they are not taken into account as they are not incorporated in the complex model of the complete abdominal aorta.

8.5.3 In vivo stress incorporation using the BDM

As described above, a simplified model of an arterial segment at the distal abdominal aorta was used in a BDM-based optimization procedure to fit the parameter μ of the Arruda-Boyce material model for a given value of λ_m. The resulting parameters ($\mu = 24358\,Pa$ and $\lambda_m = 1.01$) are now included in the complete image-based structural model of the abdominal mouse aorta, which was assumed to be obtained for the same end-diastolic pressure of $p_{dia} = 60\,mmHg$.

In order to obtain the desired deformation of the arterial wall in a structural simulation, the incorporation of the *in vivo* stress state in the image-based geometry is indispensable. This *in vivo* stress field was computed, based on the in Part II proposed backward displacement method. Twenty-nine iterations were required to obtain a relative residual below 10^{-3}. This corresponded to a mean distance of $2.5\ 10^{-4}\,mm$ and a maximum distance of $1.19\ 10^{-3}\,mm$ which was still present between the nodes of the image-based geometry and the nodes of the geometry resulting from the forward problem. The resulting zero-pressure geometry is shown in Figure 8.10, the *in vivo* stress field (present at end-diastolic blood pressure) is shown in Figures 8.18(a) and 8.19(a).

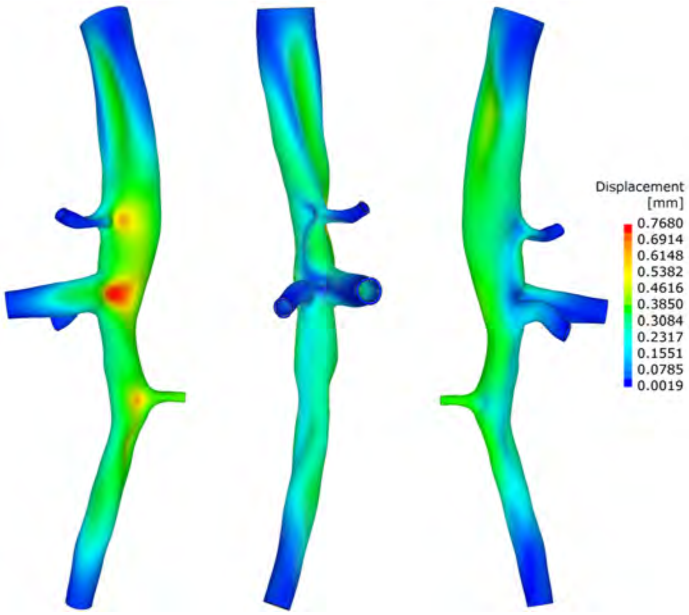


Figure 8.10: Zero-pressure geometry (for three different views), with contour plots of the displacement denoting the distance between the zero-pressure geometry and the image-based geometry.

8.6 Fluid-structure interaction model

8.6.1 Solving the coupled problem

The fluid-structure interaction problem is solved using a partitioned approach in which a flow solver and a structural solver are coupled. In this dissertation, the flow solver Fluent (Ansys) numerically solves the Navier-Stokes equations determining the fluid flow, the structural solver Abaqus/Standard (Simulia) solves the momentum equation governing the structural deformation and the in-house developed code Tango performs the coupling [20]. The coupling of two black box solvers has the advantage that codes, which are dedicated, developed, and optimized for either the flow problem or the structural problem, can be applied. Where the flow problem is solved for a given interface displacement, the structural problem is solved for a stress boundary condition applied on the fluid-structure interface. Given $\boldsymbol{x}$ the discretized displacement of the nodes at the interface, computed by the structural solver $\mathcal{S}$, and $\boldsymbol{y} = \mathcal{F}(\boldsymbol{x})$ the stress distribution on the interface, computed by the flow solver $\mathcal{F}$, the following root finding problem exists.

$$\mathcal{R}(\boldsymbol{x}) = \mathcal{S}(\mathcal{F}(\boldsymbol{x})) - \boldsymbol{x} = \mathbf{0} \tag{8.12}$$

The interface displacement, computed by the structural solver, is passed on to the IQN-ILS algorithm which provides a new prediction of the interface displacement used by the flow solver. Coupling iterations are performed between the flow solver and the structural solver in each time step to meet the equilibrium at the interface. The IQN-ILS algorithm is a quasi-Newton algorithm in which the inverse of the Jacobian is approximated by a least-squares model as discussed in Section 6.3 and in [19, 20]. The interior fluid grid nodes are moved according to a spring analogy, with a spring constant inversely proportional to the edge length. This results in a smooth redistribution of the nodes in the fluid domain. To discretize the flow equations on a moving grid that follows the motion of the structure at the fluid-structure interface, the arbitrary Lagrangian-Eulerian (ALE) method is used.

The startup of the FSI simulation is challenging as there is no true equilibrium state to start from and as the IQN-ILS algorithm does not hold information of previous time steps or coupling iterations yet to define the approximation of the inverse of the Jacobian. Moreover, FSI simulations are computationally expensive what makes it important to obtain a periodic solution as fast as possible. Therefore, quasi-static FSI analyses are performed first. In a first step, equilibrium is assessed for the image-based configuration including the end-diastolic blood pressure and the initial stress field, but for in- and outlet velocities equal to zero. Then, the pressure and the velocities are linearly increased until they match the values corresponding to the point $t = 0.052\,s$ in Figure 8.4, where the mass imbalance becomes zero. Finally, another equilibrium step is performed in which the pressure and the flows are retained. As this state does not involve wall motion, the inflow equals the total outflow what motivates the choice of the point of zero mass imbalance to start the simulations from. This approach will reduce the number of cycles which are required to obtain a periodic solution. As shown in Figure 8.11, the time after which periodicity of the solution is obtained is also sensitive to the choice of the initial windkessel pressure, p_{wk}, to which the pressure is linearly increased duing the first quasi-static analysis. This pressure was first approximated by the windkessel model and the measured outflow at the distal abdominal aorta, and then further tuned. For an initial windkessel pressure of $84.5\,mmHg$ and for windkessel parameters as mentioned in Section 8.4.1, a periodic solution was obtained with a distal abdominal aortic pressure varying between $59.7\,mmHg$ and $84.8\,mmHg$. Further results are presented for the fourth cardiac cycle. The difference in amplitude of the pressure pulse between the third and the fourth cycle is less than 0.02%.

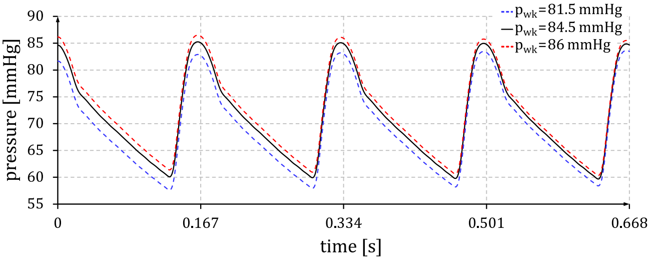
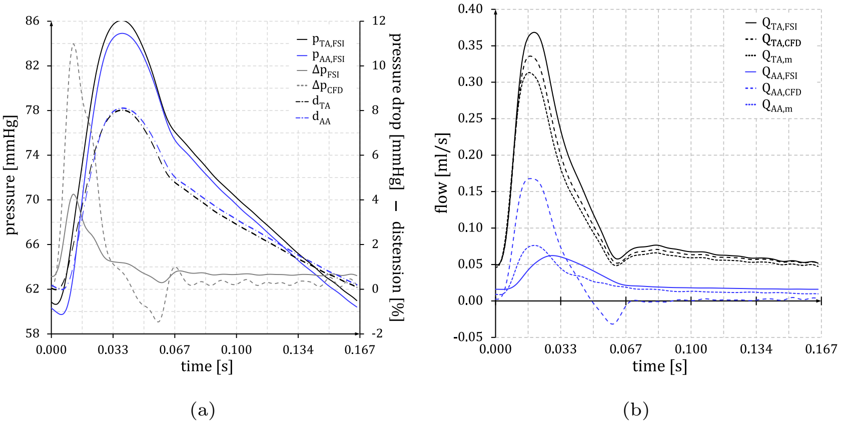


Figure 8.11: The pressure at the distal abdominal aorta in function of time for the first four cardiac cycles of the FSI simulation and for different choices of the initial windkessel pressure. A strong dependency is noticed between the moment when a regime solution is obtained and the windkessel pressure which is imposed at the start of the FSI simulation.

Using this startup procedure, FSI simulations were performed for grids which resulted in mesh independent results: the fluid domain counted $156\,k$ cells (equivalent to the case with $201\,k$ cells in Section 8.4.2) and the solid domain was composed of $44\,k$ quadratic elements. The absolute convergence criterions for the coupling iterations were set to a value of $10^{-6}\,m$ for the (L^2-norm) residual of the interface displacement and to a value of $5\,Pa$ for the residual of the interface load. This resulted in an average of 5 coupling iterations per time step. A time step size of $1.67\,ms$ was used, which resulted in a computation time of $3\,h\,33\,min$ per cardiac cycle when performed on a Dell PowerEdge R620 server with two ten-core Intel Xeon E5-2680v2 CPUs at 2.8Ghz of which one core was used by the coupling code and eighteen cores were used by the flow solver and the structural solver. This time step size, which corresponded to 100 time steps per cardiac cycle, was chosen as a compromise between accuracy and computation time as a simulation with a time step size of 0.176 ms and a computation time of 13 h 52 min per cardiac cycle did not change the simulation results significantly, with a difference in amplitude of the pressure pulse of only 1.12 %. .

8.6.2 Simulation results

FSI simulations were performed as described in the previous section. The most important simulation results are given below and compared to rigid wall CFD simulations.



(a)

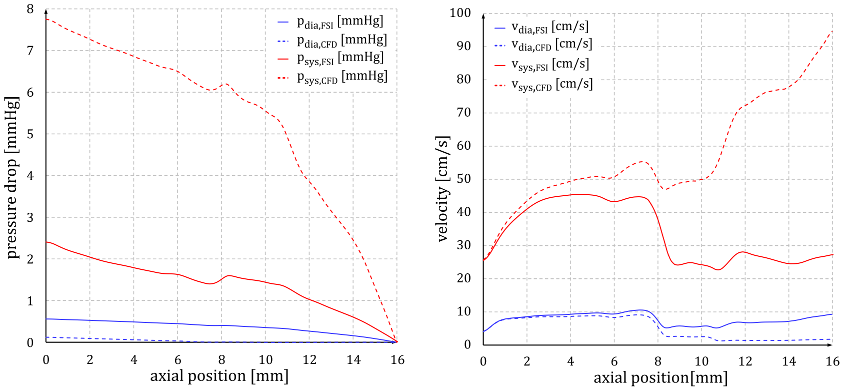
(b)

Figure 8.12: Evolution of (a) pressure [$mmHg$], pressure drop [$mmHg$] and distension [%] and (b) flow rate [ml/s] as a function of time for one cardiac cycle and for both the inlet (TA) and the outlet (AA) of the aorta. The pressure drop is presented for both the FSI simulation (full line) and the CFD simulation (dashed line).

Figure 8.12 presents the pressure, distension and flow as a function of time, and this for both the inlet (TA) and the outlet (AA) of the abdominal aorta. In Figure 8.12(a), the pressure is presented for one cardiac cycle. The pressure ranges from 60.6 to 86.0 $mmHg$ at the inlet and from 59.7 to 84.8 $mmHg$ at the outlet. The pressure drop between the inlet and the outlet as a function of time is shown as well, with a maximum value of 4.2 $mmHg$ which is 2.62× lower than the maximum pressure drop obtained during the CFD simulation. For the distension, a maximum value of 8.0% is reached at the inlet and a value of 8.1% at the outlet.

In Figure 8.12(b), the flow is presented for one cardiac cycle, again for both the inlet and the outlet of the abdominal aorta. As velocity boundary conditions were applied and not the flows, the resulting inflow for the FSI case was higher than in the CFD case, which was performed with rigid walls for the image-based (end-diastolic) geometry. Note that the flow through the inlet of the CFD model equals the approximation $Q_{TA,m}$ of the measured inflow, Equation (8.1), scaled with a factor of 1.07 to match the criterion of a time-averaged zero mass imbalance. The resulting outflow at the distal part of the aorta is presented as well. Apart from a phase shift, it is noted that the maximum value of the FSI calculated outflow corresponds better to the measured outflow than the flow resulting from the CFD analysis which predicts a significant amount of backflow in the end-systolic phase, followed by a close to zero outflow during diastole.

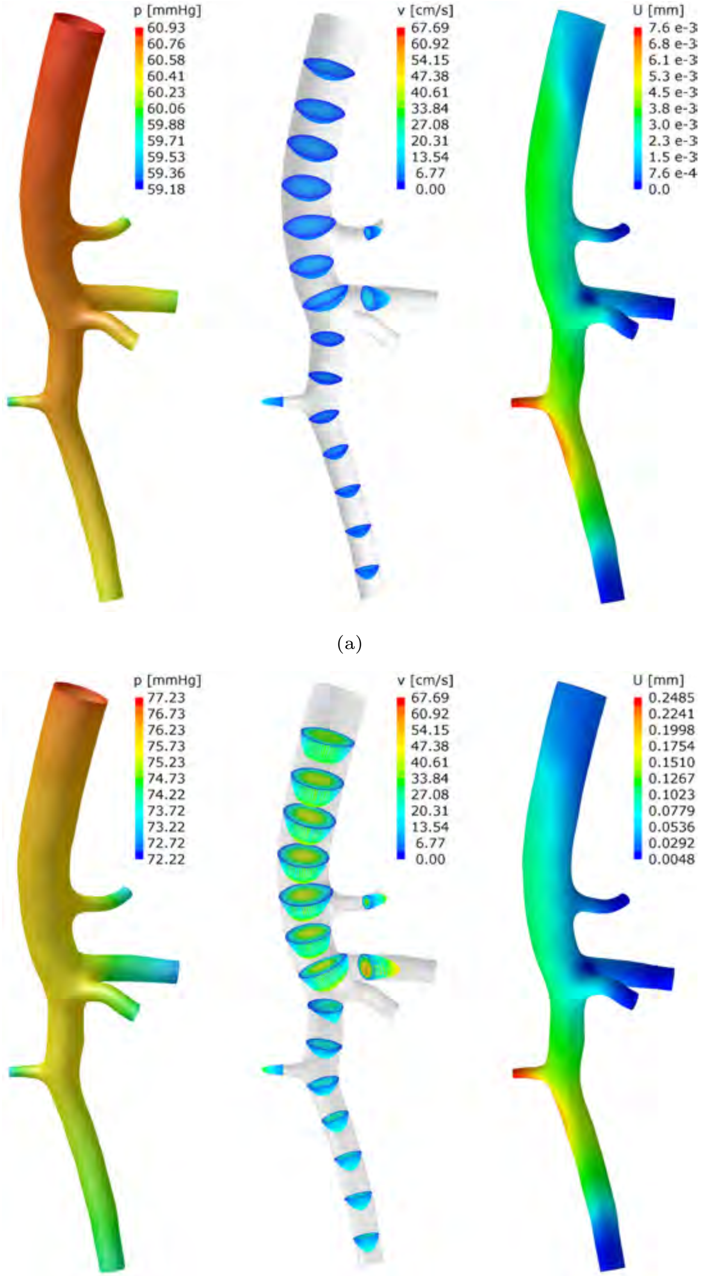
Figure 8.13 presents the pressure drop and the velocity along the 16 mm centerline, for both end-diastolic and peak-systolic flow conditions and for both the FSI simulations and the corresponding CFD simulations. From Figure 8.13(a) it is clear that the CFD predicted pressure drops overestimate the FSI results at peak-systolic flow conditions ($t = 0.02\,s$) and underestimate the FSI results at end-diastolic flow conditions ($t = 0.1653\,s$). The same is true for the velocities along the centerline, Figure 8.13(b), where the initial increase in velocity marks the transition region required to fully develop the flow (as a uniform velocity profile was applied at the inlet).



(a)

(b)

Figure 8.13: (a) Pressure drop [mmHg] and (b) velocity magnitude [cm/s] along the centerline for both the FSI simulation (full line) and the CFD simulation (dashed line), for peak-systolic flow conditions (red) and end-diastolic flow conditions (blue).



(a)

(b)

Figure 8.14: Contour plots of the pressure (left), velocity vectors coloured by magnitude at cross sectional planes along the centerline (middle) and contour plots of the interface displacement (right) at the moment of (a) end-diastole and (b) peak-systole. Note that only the subfigures representing the velocity profiles apply the same color scale.

Figure 8.14 visualizes previous findings by means of contour plots of the pressure, velocity vectors at cross sectional planes along the centerline and contour plots of the interface displacement. This interface displacement is presented relative to the image-based geometry, obtained for a diastolic blood pressure. The contour plots are shown for both the end-diastolic flow conditions, Figure 8.14(a) and the peak-systolic flow conditions, Figure 8.14(b). The velocity profile at peak-systole requires a longer distance to become fully developed, what is in agreement with Figure 8.13(b). The overestimation by CFD compared to FSI of the flows and velocities at systole and the underestimation at diastole are clearly visualized when comparing the velocity profiles at the distal abdominal aorta with those obtained for a rigid wall CFD simulation, Figure 8.15 (left).

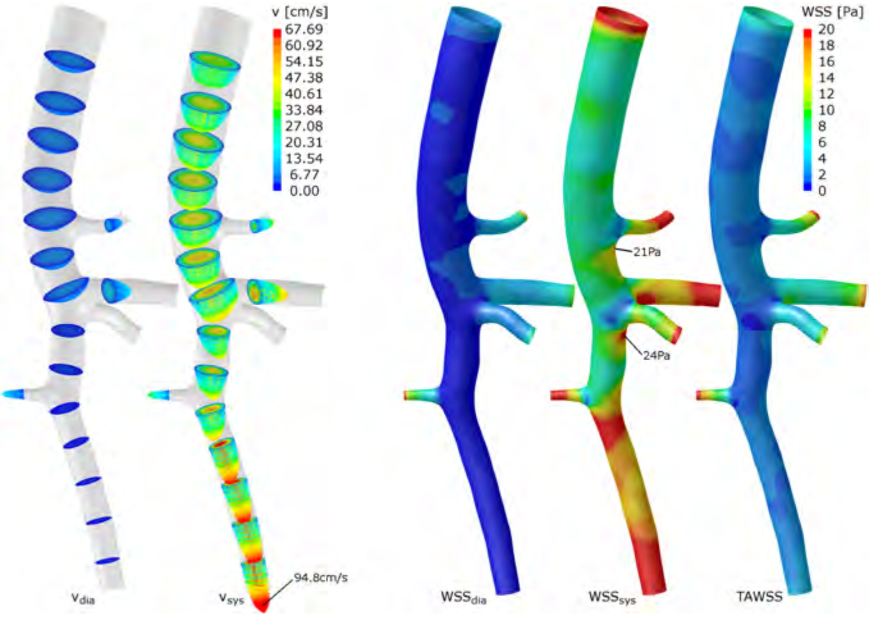
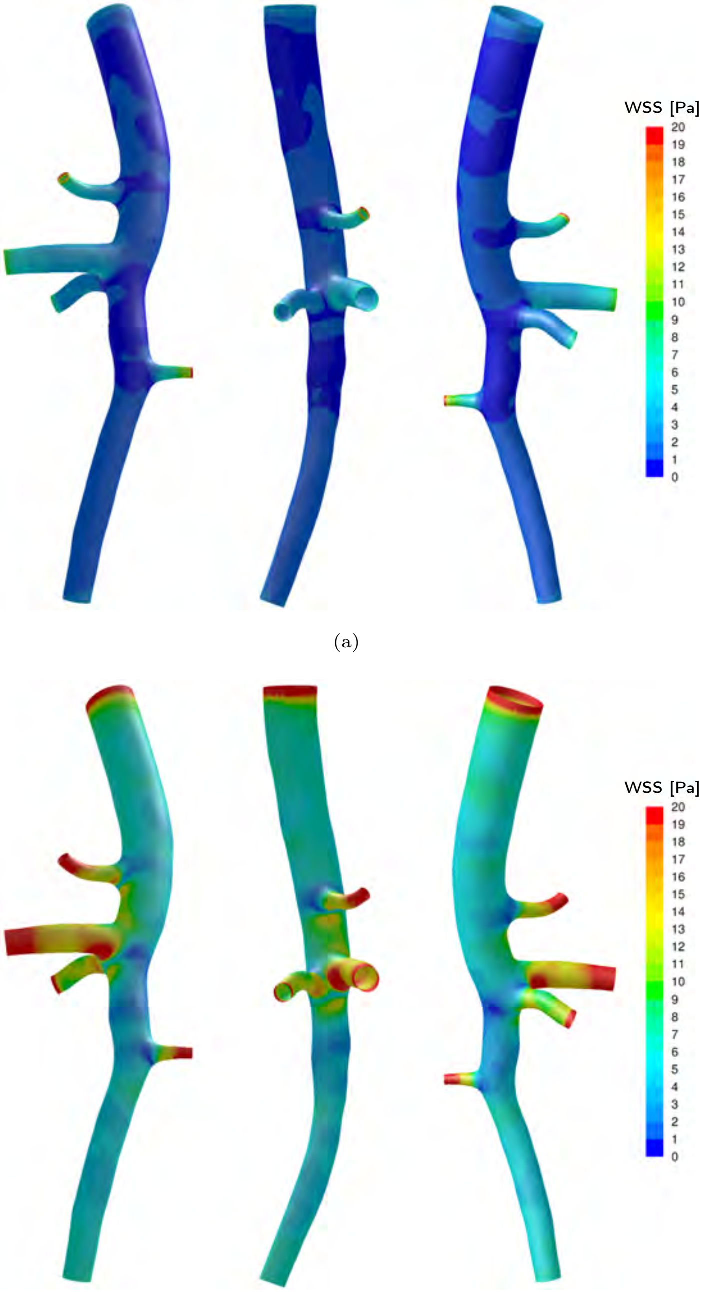


Figure 8.15: CFD results for end-diastolic and peak-systolic flow conditions. The velocity vectors at cross sectional planes along the centerline (left) and the contour plots of the wall shear stresses (right), including the time-averaged wall shear stress. Note that the same color scales were applied to represent the FSI results.

Figure 8.16 presents the wall shear stress at end-diastolic and peak-systolic flow conditions. Figure 8.17 presents the time-averaged wall shear stress for one cardiac cycle. With respect to the main vessel, zones of high wall shear stress can be found below the celiac bifurcation and the trifurcation, whereas zones of low wall shear stress are found lateral to where the side branches sprout. When compared to the



(a)

(b)

Figure 8.16: Contour plots of the wall shear stress (WSS) in [Pa] at the moment of (a) end-diastole and (b) peak-systole, for three different views.

wall shear stress obtained in the CFD simulation, Figure 8.15 (right), the CFD results show a peak-systolic wall shear stress which is overestimated, whereas the end-diastolic and the time-averaged wall shear stress are underestimated. However, a similar spatial distribution is found for the CFD results, but with less distinct zones of low wall shear stress lateral to the side branches.

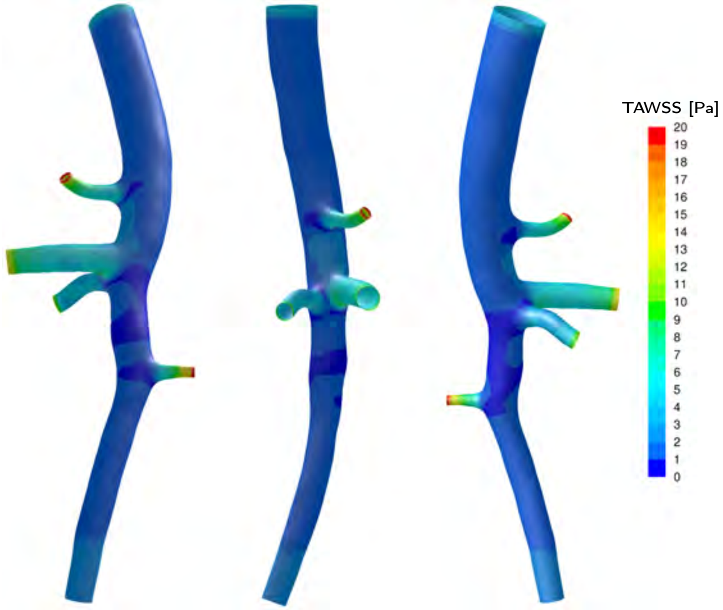


Figure 8.17: Contour plot of the time-averaged wall shear stress (TAWSS) in [Pa] for three different views.

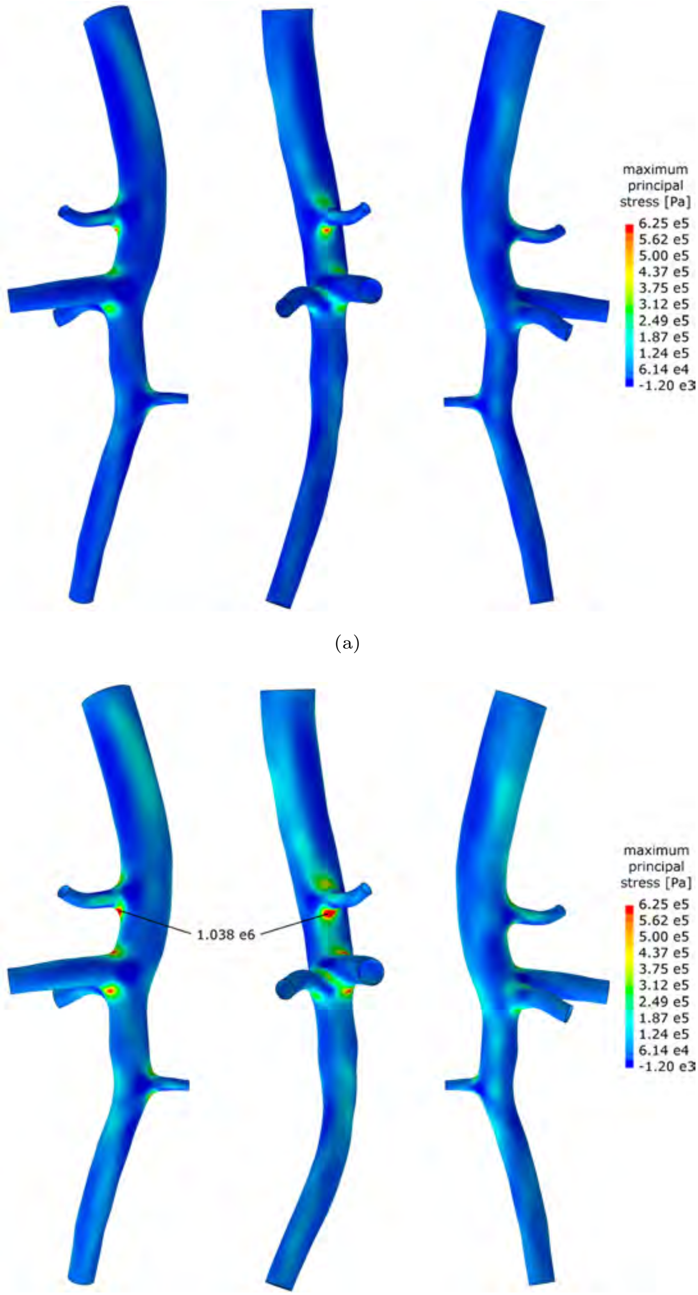
Finally, the maximum principal stress distribution is presented at the inner surface of the arterial wall, Figure 8.18, and the outer surface of the arterial wall, Figure 8.19. When the diastolic blood pressure or the systolic blood pressure is imposed, the spatial distribution of the stress patterns is similar, with regions of elevated stress around the side branches. However, when comparing the results for the inner wall with those for the outer wall, these bifurcation regions show a significant difference in stress distribution.

8.7 Conclusion

In this chapter, the modelling aspects covered in Part I and Part II of this work were applied to model the fluid-structure interaction in an abdominal mouse aorta. The choice for an abdominal mouse aorta was based on its clinical relevance and the in-house availability of mouse-specific datasets obtained in previous research.

Solving the hemodynamics and the arterial wall mechanics separately is complex already, but the coupled nature of the problem makes it even more challenging.

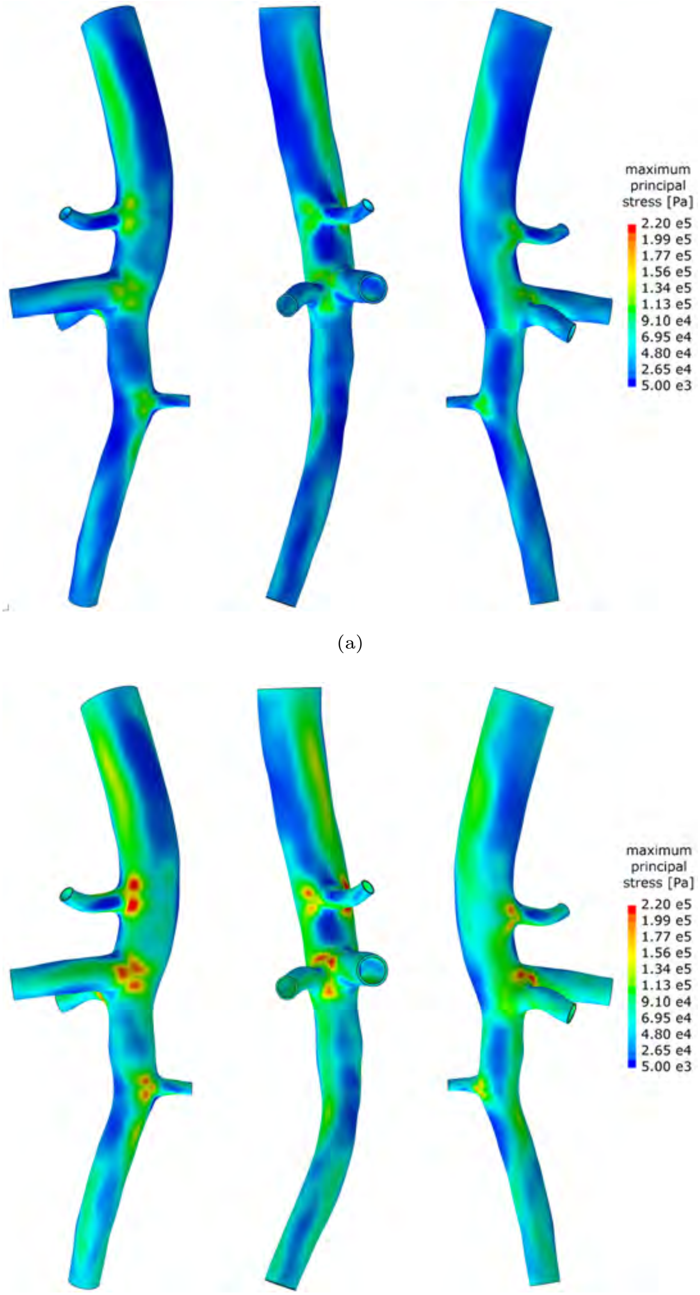
First, the *in vivo* data acquisition was discussed along with the application of the extended Treemesh method which resulted in a computational grid. Then, a CFD model was constructed for which the sensitivity of the flow results on the grid density was analyzed and compared to the results obtained with tetrahedral grids. Next, a computational structural mechanics (CSM) model for the arterial wall was set up and prestressed using the backward displacement method. This was done for material parameters which were identified such that realistic distensions were obtained in the physiological pressure range. Finally, the flow simulation and the structural simulation (including prestress) were coupled by solving the FSI problem using a partitioned approach. Besides pressure drops, velocity fields, and wall shear stresses – which are overestimated by a rigid wall CFD model at peak-systole and underestimated at end-diastole – the FSI simulation also provided data on the static pressure, the deformation of the arterial wall and the stress distribution in the arterial wall.



(a)

(b)

Figure 8.18: Contour plots of the maximum principal stress at the inner wall (i.e. at the fluid-structure interface) at the moment of (a) end-diastole and (b) peak systole, for three different views. Note that the same color scale is applied to both subfigures.



(a)

(b)

Figure 8.19: Contour plots of the maximum principal stress at the outer wall at the moment of (a) end-diastole and (b) peak systole, for three different views. Note that the same color scale is applied to both subfigures.

Conclusions

Overall conclusions and future perspectives

Conclusion

In the last decade, a clear trend is observed towards realistic models of parts of the vasculature, based on *in vivo* measurements. This is done by studying patient-specific geometries and/or including the coupling between the behaviour of the blood flow and the vessel wall. Unfortunately, this also implies that the numerical models become more challenging to develop and to solve. From a preprocessing point of view, complex geometries tend to increase the operator time while from a computational point of view, larger computational domains and fluid-structure interaction increase the computation time. This is especially true when the domain of interest involves a vascular pathology such as a local narrowing or a local dilatation of the arterial wall. Therefore, the aim of this work was to develop computational strategies which allow to generate efficient and accurate fluid-structure interaction models of complex, image-based vascular structures.

In an attempt to overcome these issues of high operator times and high computation times when addressing complex geometries, Part I of this work introduces a novel approach to generate full-hexahedral fluid meshes of complex vascular structures. The generated grids feature computational efficiency by aligning the grid lines with the predominant direction of the flow and allowing local (anisotropic) refinements. Moreover, high-quality cells are generated even in high-curvature regions, which improves the robustness of the method and allows for automation of the procedure to save valuable operator time. First, an algorithm was proposed which unambiguously defined the branching topology of a vascular network in terms of parent vessels. By interconnecting vessel enclosing squares according to the previously defined branching topology, it was shown that the generation of a multi-block structure could be automated. Then, the grid-based method was applied to a refined multi-block structure in combination with a vessel geometry.

This resulted in an unstructured set of high-quality hexahedral cells which resembled the geometry well and formed a vessel-aligned core for the fluid domain. By performing this procedure on a radially compressed geometry of the surface geometry instead, it was possible to introduce buffer layers for the generation of a high-quality boundary layer grid, even in high-curvature regions. Robustness of unstructured hexahedral mesh generation using a multi-block grid-based approach was demonstrated by applying the methodology to common complex geometries of treelike structures including mild and severe stenoses, aneurysms and trifurcations. Extending the multi-block structure with additional blocks, allowed for meshing the most complex geometries such as giant saccular aneurysms. It was demonstrated that only six parameters, of which two were optional, were required to control the mesh density, moreover, vectorizing some of these parameters allowed for local refinements in both the axial direction and the cross sections. The software implementation in pyFormex combines a graphical user interface with a software wizard to guide (non-expert) users through the meshing procedure and speed up non-default manipulations such as local grid refinements and additional block structure inclusions. In conclusion, unstructured hexahedral mesh generation using a multi-block grid-based approach is able to combine the best of two worlds: the *automated mesh generation* which was previously only available for tetrahedral grids, with the *computational efficacy* of vessel aligned hexahedral grids.

Numerical modelling of complex vascular structures is associated with the trend towards more realistic, patient-specific models of which the geometries are obtained *in vivo* by medical imaging. Part II of this dissertation deals with the *in vivo* stress and strain fields present in the arterial wall (due to internal pressure load) at the moment of medical image acquisition. To emphasize their importance in this field of research, we showed that the incorporation of *in vivo* stress in arteries is necessary to properly estimate the stress in the arterial wall and the deformation of the arterial wall when simulating the physiological pressure load inside image-based vascular structures. To account for this unknown stress field, an inverse problem was formulated using a backward displacement function and a function which represents the outcome of the structural solver. As such, we were able to solve for the zero-pressure geometry and the *in vivo* stress tensor field starting from the *in vivo* measured geometry of the loaded structure, using fixed point iterations. In order to accelerate the backward displacement method, the corresponding root finding problem was solved using a quasi-Newton algorithm instead, with an approximation for the inverse of the Jacobian from a least squares model. Furthermore, it was demonstrated that the convergence rate of the proposed techniques decreased only slightly for a much more complex structure using the same constitutive material law. Next, it was shown for a simplified artery that the backward displacement method can be combined with an optimization procedure to retrieve the material parameters based on *in vivo* measurements only. Finally and most importantly, the backward displacement method allows for a straightforward implementation of the algorithm in combination with existing structural solvers as only an update of the coordinates of the mesh is required, this was explicitly confirmed by other authors [84].

In the final part, the modelling aspects covered in Part I and Part II of this work were applied to model the fluid-structure interaction of an abdominal mouse aorta. The choice for an abdominal mouse aorta was based on its clinical relevance and the in-house availability of mouse-specific datasets obtained in previous research. Solving the hemodynamics and the arterial wall mechanics separately is complex already, but the coupled nature of the problem makes it even more challenging. First, the *in vivo* data acquisition is discussed along with the application of the extended Treemesh method which resulted in proper computational grids. Then, a CFD model was constructed for which the sensitivity of the flow results on the grid density was analyzed and compared to the results obtained with tetrahedral grids. Next, a computational structural mechanics (CSM) model for the arterial wall was set up and prestressed using the backward displacement method. This was done for material parameters which were identified such that the *in vivo* measured distension was obtained in the physiological pressure range. Finally, the flow simulation and the structural simulation (including prestress) were coupled by solving the FSI problem using a partitioned approach. Besides pressure drops, velocity fields, and wall shear stresses – which are overestimated by a rigid wall CFD model at peak-systole and underestimated at end-diastole – the FSI simulation also provided data on the static pressure, the deformation of the arterial wall and the stress distribution in the arterial wall.

Recommendations for future work

The current work has highlighted some important aspects in the numerical modelling process of complex, image-based vascular structures. However, there are still some issues that need to be addressed in the future.

The proposed branching topology in terms of parent vessels does not allow for loops such as an arterial bypass or the Circle of Willis in the cerebral circulation.

Extending the multi-block structure by fully enclosing such a region tackles this problem. However, this reduces the multi-block grid-based approach to a standard grid-based approach with low quality cells at high-curvature regions and without vessel alignment.

When extending the multi-block structure with additional blocks or when providing local refinements, the operator needs to specify which regions require special care. As such, the current software implementation only allows for semi-automated mesh generation for these cases, e.g. by providing separate geometry files of the most complex regions. This means that an additional speed up can be realized by introducing an algorithm for the detection of complex/critical regions.

Often, anisotropic material models are used. In order to assign fiber directions, each finite element requires a local coordinate system, which can be specified based on the orientation of the element [46]. However, due to the unstructured character of the presented grids, alternative methods are required [1, 37].

Regarding *in vivo* stress incorporation, this dissertation only considered the effect of the internal pressure which is present at the moment of medical imaging, and not the residual stresses which would still be present in the zero-pressure geometry. Although the proposed backward displacement method would allow the inclusion of axial prestretch in straight, non-bifurcating vessels, this should be further investigated for more complex geometries for which large deformations might hamper proper convergence. If this would be the case, external tissue support may bring a solution [45]. If the *in vivo* axial prestretch would be unknown, the proposed BDM-based material parameter optimization procedure can be extended by including the unkown axial prestretch in the set of material parameters and by including an additional axial-force constraint in the objective function [66]. Neglecting the residual stresses which are still present after excision of the vessel, leads typically to significant stress gradients across the wall thickness what is in contradiction to the uniform stress hypothesis [24, 25]. A BDM-based residual stress optimization framework can be set up, similar to the BDM-based material parameter optimization framework, but with an unknown residual stress field to solve for and an objective function which evaluates the stress gradient across the wall. The residual stress field and its updates can be introduced in the zero-pressure geometry by means of a thermo-mechanical analysis [49].

Appendices

List of Figures

List of Tables

List of Algorithms

List of Publications

Peer-reviewed papers, covering this reseach

- Part I
  - Bols J., Taelman L., De Santis G., Degroote J., Verhegghe B., Segers P. and Vierendeels J. *"Unstructured hexahedral mesh generation in complex vascular structures using a grid-based approach."* Computer Methods in Biomechanics and Biomedical Engineering, submitted.
  - Steinman D.A., et al. *"Variability of Computational Fluid Dynamics Solutions for Pressure and Flow in a Giant Aneurysm: The ASME 2012 Summer Bioengineering Conference CFD Challenge."* ASME: Journal of Biomechanical Engineering, 135(02): 021015, 2013.
  - Taelman L., Bols J., Degroote J., Muthurangu V., Panzer J., Vierendeels J. and Segers P. *"Insights into the effect of a local stiffness and narrowing on central hemodynamics in repaired aortic coarctation."* Annals of Biomedical Engineering, submitted.
- Part II
  - Bols J., Degroote J., Trachet B., Verhegghe B., Segers P. and Vierendeels J. *"A computational method to assess the in vivo stresses and unloaded configuration of patient-specific blood vessels."* Journal of Computational and Applied Mathematics, 246:10-17, 2013.
  - Bols J., Degroote J., Trachet B., Verhegghe B., Segers P. and Vierendeels J. *"Inverse modelling of image-based patient-specific blood vessels: zero-pressure geometry and in vivo stress incorporation."* ESAIM: Mathematical Modelling and Numerical Analysis, 47:1059-1075, 2013.
- Part III
  - Trachet B., Bols J., De Santis G., Vandenberghe S., Loeys B., Segers P. *"The impact of simplified boundary conditions and aortic arch inclusion*

on CFD simulations in the mouse aorta: a comparison with mouse-specific reference data." ASME: Journal of Biomechanical Engineering, 133(12): 121006, 2011.

- Trachet B., Renard M., Van der Donckt C., Deleye S., Bols J., De Meyer G.R.Y., Staelens S., Loeys B.L., Segers P. *"Longitudinal follow-up of ascending versus abdominal aortic aneurysm formation in angiotensin II-infused ApoE-/- mice"* Artery Research, 8(1): 16-23, 2014.

Conference proceedings, covering this reseach

- Bols J., De Santis G., Degroote J., Verhegghe B., Segers P. and Vierendeels J. *"Improved method to auto-generate high-quality hexahedral meshes of complex vascular trees"* 20th Congress of the European Society of Biomechanics, ESB 2013, August 25-28, Patras (Greece).
- Bols J., De Santis G., Degroote J., Verhegghe B., Segers P. and Vierendeels J. *"Automated hexahedral mesh generation in a complex vascular tree: the extended Treemesh method"* Proceedings of the ASME Summer Bioengineering Conference, SBC 2013, June 26-29, Sunriver (Oregon, USA).
- Bols J., Degroote J., Trachet B., Verhegghe B., Segers P. and Vierendeels J. *"Inverse method to include prestress into image-based geometries of the arterial wall"* 6th European Congress on Computational Methods in Appliead Sciences and Engineering, ECCOMAS 2012, September 10-14, Vienna (Austria).
- Bols J., De Santis G., Degroote J., Verhegghe B., Segers P. and Vierendeels J. *"CFD challenge: solutions using the commercial finite volume solver, fluent, and a pyFormex-generated full hexahedral mesh"* Proceedings of the ASME Summer Bioengineering Conference, SBC 2012, June 20-23, Fajardo (Puerto Rico, USA).
- Bols J., Trachet B., Degroote J., De Santis G., Verhegghe B., Segers P. and Vierendeels J. *"A numerical model of the aorta of a mouse: a first step towards life-saving treatment"* 10th National Day on Biomedical Engineering, NCBME 2011, December 2, Brussels (Belgium).
- Bols J., Degroote J., Trachet B., Verhegghe B., Segers P. and Vierendeels J. *"A computational method to assess initial stresses and unloaded configuration of patient-specific blood vessels"* 5th International Conference on Advanced Computational Methods in Engineering, ACOMEN 2011, November 14-17, Liège (Belgium).

– Bols J., Trachet B., Degroote J., De Santis G., Mortier P., Verhegghe B., Segers P. and Vierendeels J. *"Structural simulation of a mouse-specific abdominal aorta"* Proceedings of the ASME Summer Bioengineering Conference, SBC 2011, June 22-25, Farmington (Pennsylvania, USA).

Bibliography

[1] V. Alastrué, A. Garía, E. Peña, J. F. Rodríguez, M. A. Martínez, and M. Doblaré. Numerical framework for patient-specific computational modelling of vascular tissue. *International Journal for Numerical Methods in Biomedical Engineering*, 26(1):35–51, 2010.

[2] L. Antiga. *Patient-Specific Modeling of Geometry and Blood Flow in Large Arteries*. PhD thesis, Politecnico di Milano, 2002.

[3] L. Antiga, B. Ene-Iordache, L. Caverni, G. P. Cornalba, , and A. Remuzzi. Geometric reconstruction for computational mesh generation of arterial bifurcations from CT angiography. *Computerized Medical Imaging and Graphics*, 26(4):227 – 235, 2002.

[4] L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. *IEEE Transactions on Medical Imaging*, 22(5):674–684, 2003.

[5] L. Antiga, M. Piccinelli, L. Botti, B. Ene-Iordache, A. Remuzzi, and D. A. Steinman. An image-based modeling framework for patient-specific computational hemodynamics. *Medical & Biological Engineering & Computing*, 46 (11):1097–1112, 2008.

[6] Ellen M. Arruda and Mary C. Boyce. A three-dimensional constitutive model for the large stretch behavior of rubber elastic materials. *Journal of the Mechanics and Physics of Solids*, 41(2):389 – 412, 1993.

[7] A. Arzani, P. Dyverfeldt, T. Ebbers, and S. C. Shadden. In vivo validation of numerical prediction for turbulence intensity in an aortic coarctation. *Annals of Biomedical Engineering*, 40(4):860–870, 2012.

[8] Y. Bazilevs, M. C. Hsu, Y. Zhang, W. Wang, T. Kvamsdal, S. Hentschel, and J. Isaksen. Computational vascular fluid-structure interaction: method-

ology and application to cerebral aneurysms. *Biomechanics and Modeling in Mechanobiology*, 9(4):481–498, 2010.

[9] B. Berthier, R. Bouzerar, and C. Legallais. Blood flow patterns in an anatomically realistic coronary vessel: influence of three different reconstruction methods. *Journal of Biomechanics*, 35(10):1347 – 1356, 2002.

[10] R. Biswas and R. C. Strawn. Tetrahedral and hexahedral mesh adaptation for CFD problems. *Applied Numerical Mathematics*, 26(1-2):135 – 151, 1998.

[11] J. Bols, J. Degroote, B. Trachet, B. Verhegghe, P. Segers, and J. Vierendeels. A computational method to assess initial stresses and unloaded configuration of patient-specific blood vessels. In *Fifth international conference on advanced computational methods in engineering (ACOMEN 2011)*, page 10. Université de Liège, 2011.

[12] J. Bols, J. Degroote, G. De Santis, B. Trachet, P. Segers, and J. Vierendeels. Cfd challenge: solutions using the commercial finite volume solver, fluent, and a pyformex-generated full hexahedral mesh. In *Proceedings of the ASME 2012 Summer Bioengineering Conference*, pages 1–2, 2012.

[13] J. R. Cebral, F. Mut, M. Raschi, E. Scrivano, R. Ceratto, P. Lylyk, and C. M. Putman. Aneurysm rupture following treatment with flow-diverting stents: Computational hemodynamics analysis of treatment. *American Journal of Neuroradiology*, 32(1):27–33, 2011.

[14] S. de Putter, B. J. B. M. Wolters, M. C. M. Rutten, M. Breeuwer, F. A. Gerritsen, and F. N. van de Vosse. Patient-specific initial wall stress in abdominal aortic aneurysms with a backward incremental method. *Journal of Biomechanics*, 40(5):1081–1090, 2007.

[15] G. De Santis. *Novel mesh generation method for accurate image-based computational modelling of blood vessels.* PhD thesis, Ghent University, 2012.

[16] G. De Santis, P. Mortier, M. De Beule, P. Segers, P. Verdonck, and B. Verhegghe. Patient-specific computational fluid dynamics: structured mesh generation from coronary angiography. *Medical & Biological Engineering & Computing*, 48(4):371–380, 2010.

[17] G. De Santis, M. De Beule, P. Segers, P. Verdonck, and B. Verhegghe. Patient-specific computational haemodynamics: generation of structured and conformal hexahedral meshes from triangulated surfaces of vascular bifurcations. *Computer Methods in Biomechanics and Biomedical Engineering*, 14 (9):797–802, 2011.

[18] G. De Santis, M. De Beule, K. Van Canneyt, P. Segers, P. Verdonck, and B. Verhegghe. Full-hexahedral structured meshing for image-based computational vascular modeling. *Medical Engineering & Physics*, 33(10):1318 – 1325, 2011.

[19] J. Degroote, KJ. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers and Structures*, 87(11-12):793–801, June 2009.

[20] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid-structure interaction. *Computers and Structures*, 88(7-8):446–457, April 2010.

[21] J. Degroote, I. Couckuyt, J. Vierendeels, P. Segers, and T. Dhaene. Inverse modelling of an aneurysm's stiffness using surrogate-based optimization and fluid-structure interaction simulations. *Structural and Multidisciplinary Optimization*, 46(3):457–469, 2012.

[22] V. D. Fachinotti, A. Cardona, and P. Jetteur. Finite element modelling of inverse design problems in large deformations anisotropic hyperelasticity. *International Journal for Numerical Methods in Engineering*, 74(6):894–910, 2008.

[23] M. D. Ford, H. N. Nikolov, J. S. Milner, S. P. Lownie, E. M. Demont, W. Kalata, F. Loth, D. W. Holdsworth, and D. A. Steinman. PIV-measured versus CFD-predicted flow dynamics in anatomically realistic cerebral aneurysm models. *Journal of Biomechanical Engineering*, 130(2):021015, 2008.

[24] Y. C. Fung. What are residual stresses doing in our blood vessels? *Annals of Biomedical Engineering*, 19(3):237–249, 1991.

[25] Y.C. Fung. *Biomechanics: Motion, Flow, Stress, and Growth.* Springer, 1990.

[26] F. Gao, Z. Guo, M. Sakamoto, and T. Matsuzawa. Fluid-structure interaction within a layered aortic arch model. *J Biol Phys*, 32(5):435–54, 2006.

[27] M. W. Gee, C. Reeps, H. H. Eckstein, and W.A. Wall. Prestressing in finite deformation abdominal aortic aneurysm simulation. *Journal of Biomechanics*, 42:1732–1739, 2009.

[28] M. W. Gee, Ch. Förster, and W. A. Wall. A computational strategy for prestressing patient-specific biomechanical problems under finite deformation. *International Journal for Numerical Methods in Biomedical Engineering*, 26(1):52–72, 2010.

[29] S. Govindjee and P. A. Mihalic. Computational methods for inverse finite elastostatics. *Computer Methods in Applied Mechanics and Engineering*, 136 (1-2):47 – 57, 1996.

[30] S. Govindjee and P. A. Mihalic. Computational methods for inverse deformations in quasi-incompressible finite elasticity. *International Journal for Numerical Methods in Engineering*, 43(5):821–838, 1998.

[31] N. M. Grosland, K. H. Shivanna, V. A. Magnotta, N. A. Kallemeyn, N. A. DeVries, S. C. Tadepalli, and C. Lisle. IA-FEMesh: An open-source, interactive, multiblock approach to anatomic finite element model development. *Computer Methods and Programs in Biomedicine*, 94(1):96 – 107, 2009.

[32] R. Haelterman, J. Degroote, D. Van Heule, and J. Vierendeels. The Quasi-Newton Least Squares method: A new and fast secant method analyzed for linear systems. *SIAM Journal on Numerical Analysis*, 47(3):2347–2368, 2009.

[33] R. Haelterman, J. Degroote, D. Van Heule, and J. Vierendeels. On the similarities between the Quasi-Newton Inverse Least Squares method and GMRes. *SIAM Journal on Numerical Analysis*, 47(6):4660–4679, 2010.

[34] K. Ho-Le. Finite element mesh generation methods: A review and classification. *Computer-Aided Design*, 20(1):27–38, February 1988.

[35] M. C. Hsu and Y. Bazilevs. Blood vessel tissue prestress modeling for vascular fluid-structure interaction simulation. *Finite Elements in Analysis and Design*, 47(6):593–599, June 2011.

[36] H. J. Kim, I. E. Vignon-Clementel, C. A. Figueroa, J. F. LaDisa, K. E. Jansen, J. A. Feinstein, and C. A. Taylor. On coupling a lumped parameter heart model and a three-dimensional finite element aorta model. *Annals of Biomedical Engineering*, 37(11):2153–2169, 2009.

[37] D. E. Kiousis, S. F. Rubinigg, M. Auer, and G. A. Holzapfel. A methodology to analyze changes in lipid core and calcification onto fibrous cap vulnerability: the human atherosclerotic carotid bifurcation as an illustratory example. *J Biomech Eng*, 131(12):121002, 2009.

[38] K. Kovalev. *Unstructured Hexahedral Non-conformal Mesh Generation*. PhD thesis, Vrije Universiteit Brussel, 2005.

[39] S. Lee. Solution method for transitional flow in a vascular bifurcation based on in vivo medical images. Master's thesis, University of Illinois at Chicago, 2002.

[40] Y. Liu, K. Pekkan, S. C. Jones, and A. P. Yoganathan. The effects of different mesh generation methods on computational fluid dynamic analysis and power loss assessment in total cavopulmonary connection. *Journal of Biomechanical Engineering*, 126(5):594–603, 2004.

[41] Q. Long, X.Y. Xu, M. W. Collins, M. Bourne, and T.M. Griffith. Magnetic resonance image processing and structured grid generation of a human abdominal bifurcation. *Computer Methods and Programs in Biomedicine*, 56 (3):249 – 259, 1998.

[42] P. W. Longest and S. Vinchurkar. Effects of mesh style and grid convergence on particle deposition in bifurcating airway models with comparisons to experimental data. *Medical Engineering & Physics*, 29(3):350 – 366, 2007.

[43] J. Lu, X. Zhou, and M. L. Raghavan. Inverse elastostatic stress analysis in pre-deformed biological structures: Demonstration using abdominal aortic aneurysms. *Journal of Biomechanics*, 40(3):693–6, 2007.

[44] M.A.G. Merkx, M. van 't Veer, L. Speelman, M. Breeuwer, J. Buth, and F.N. van de Vosse. Importance of initial stress for abdominal aortic aneurysm wall motion: Dynamic MRI validated finite element analysis. *Journal of Biomechanics*, 42(14):2369 – 2373, 2009.

[45] P. Moireau, N. Xiao, M. Astorino, C. A. Figueroa, D. Chapelle, C. A. Taylor, and J.-F. Gerbeau. External tissue support and fluid-structure simulation in blood flows. *Biomechanics and Modeling in Mechanobiology*, 11(1-2):1–18, 2012.

[46] P. Mortier, G. A. Holzapfel, M. De Beule, D. Van Loo, Y. Taeymans, P. Segers, P. Verdonck, and B. Verhegghe. A novel simulation strategy for stent insertion and deployment in curved coronary bifurcations: Comparison of three drug-eluting stents. *Annals of Biomedical Engineering*, 38(1):88–99, 2010.

[47] S. Patankar. *Numerical Heat Transfer and Fluid Flow*. Series in computational methods in mechanics and thermal sciences. Taylor & Francis, 1980.

[48] P. M. Pinsky, D. van der Heide, and D. Chernyak. Computational modeling of mechanical anisotropy in the cornea and sclera. *Journal of cataract and refractive surgery*, 31(1):136–45, 2005.

[49] S. Polzer, J. Bursa, T. C. Gasser, R. Staffa, and R. Vlachovsky. A numerical implementation to predict residual strains from the homogeneous stress hypothesis with application to abdominal aortic aneurysms. *Annals of Biomedical Engineering*, 41(7):1516–1527, 2013.

[50] P. J. Prendergast, C. Lally, S. Daly, A. J. Reid, T. C. Lee, D. Quinn, and F. Dolan. Analysis of prolapse in cardiovascular stents: A constitutive equation for vascular tissue and finite-element modelling. *Journal of Biomechanical Engineering*, 125(5):692–699, 2003.

[51] pyFormex. pyFormex is a program for generating, transforming and manipulating large geometrical models of 3D structures by sequences of mathematical operations. `http://www.pyformex.org`.

[52] A. G. Radaelli, L. Augsburger, J. R. Cebral, M. Ohta, D. A. Rufenacht, R. Balossino, G. Benndorf, D. R. Hose, A. Marzo, R. Metcalfe, P. Mortier, F. Mut, P. Reymond, L. Socci, B. Verhegghe, and A. F. Frangi. Reproducibility of haemodynamical simulations in a subject-specific stented aneurysm model: a report on the virtual intracranial stenting challenge 2007. *Journal of Biomechanics*, 41(10):2069–2081, 2008.

[53] M. L. Raghavan, B. Ma, and M. Fillinger. Non-invasive determination of zero-pressure geometry of arterial aneurysms. *Annals of Biomedical Engineering*, 34(9):1414–1419, 2006.

[54] A. J. Ramme, K. H. Shivanna, V. A. Magnotta, and N. M. Grosland. Gaussian curvature analysis allows for automatic block placement in multi-block hexahedral meshing. *Computer Methods in Biomechanics and Biomedical Engineering*, 14(10):893–904, 2011.

[55] A. J. Ramme, K. H. Shivanna, A. J. Criswell, N. A. Kallemeyn, V. A. Magnotta, and N. M. Grosland. Growing multiblock structures: a semi-automated approach to block placement for multiblock hexahedral meshing. *Computer Methods in Biomechanics and Biomedical Engineering*, 15 (10):1043–1052, 2012.

[56] M. Raschi, F. Mut, G. Byrne, C. M. Putman, S. Tateshima, F. Vinuela, T. Tanoue, K. Tanishita, and J. R. Cebral. CFD and PIV analysis of hemodynamics in a growing intracranial aneurysm. *International Journal for Numerical Methods in Biomedical Engineering*, 28(2):214–228, 2012.

[57] V. L. Rayz, L. Boussel, G. Acevedo-Bolton, A. J. Martin, W. L. Young, M. T. Lawton, R. Higashida, and D. Saloner. Numerical simulations of flow

in cerebral aneurysms: comparison of CFD results and in vivo MRI measurements. *Journal of biomechanical engineering*, 130:051011, 2008 Oct 2008.

[58] A. K. Reddy, Taffet G. E., Madala S., L. H. Michael, M. L. Entman, and C. J. Hartley. Noninvasive blood pressure measurement in mice using pulsed doppler ultrasound. *Ultrasound in Medicine & Biology*, 29(3):379 – 385, 2003.

[59] F. Riveros, S. Chandra, E. A. Finol, T. C. Gasser, and J. F. Rodriguez. A pull-back algorithm to determine the unloaded vascular geometry in anisotropic hyperelastic aaa passive mechanics. *Annals of Biomedical Engineering*, 41(4):694–708, 2013.

[60] C. S. Roy. The elastic properties of the arterial wall. *The Journal of Physiology*, 3:125–159, 1881.

[61] R. Schneiders. A grid-based algorithm for the generation of hexahedral element meshes. *Engineering with Computers*, 12(3-4):168–177, 1996.

[62] J. F. Shepherd, C. J. Tuttle, C. T. Silva, and Y. Zhang. Quality improvement and feature capture in hexahedral meshes. Technical Report Technical Report UUSCI-2006-029, SCI Institute, University of Utah, Salt Lake City, UT, USA, 2006.

[63] Y. Shimogonya, T. Ishikawa, Y. Imai, N. Matsuki, and T. Yamaguchi. Can temporal fluctuation in spatial wall shear stress gradient initiate a cerebral aneurysm? A proposed novel hemodynamic index, the gradient oscillatory number (GON). *Journal of Biomechanics*, 42(4):550 – 554, 2009.

[64] L. Speelman, E. M. H. Bosboom, G.W.H. Schurink, J. Buth, M. Breeuwer, M.J. Jacobs, and F.N. van de Vosse. Initial stress and nonlinear material behavior in patient-specific AAA wall stress analysis. *Journal of Biomechanics*, 42(11):1713 – 1719, 2009.

[65] L. Speelman, A. C. Akyildiz, B. den Adel, J. J. Wentzel, A. F. W. van der Steen, R. Virmani, L. van der Weerd, J. W. Jukema, R. E. Poelmann, E. H. van Brummelen, and F. J. H. Gijsen. Initial stress in biomechanical models of atherosclerotic plaques. *Journal of Biomechanics*, 44(13):2376 – 2382, 2011.

[66] J. Stalhand and A. Klarbring. Aorta in vivo parameter identification using an axial force constraint. *Biomechanics and Modeling in Mechanobiology*, 3 (4):191–199, 2005. ISSN 1617-7959.

[67] J. Stalhand, A. Klarbring, and M. Karlsson. Towards in vivo aorta material identification and stress estimation. *Biomechanics and Modeling in Mechanobiology*, 2(3):169–186, 2004.

[68] D. A. Steinman, Y. Hoi, P. Fahy, L. Morris, M.T. Walsh, N. Aristokleous, A.S. Anayiotos, Y. Papaharilaou, A. Arzani, S. C. Shadden, P. Berg, G. Janiga, J. Bols, P. Segers, N. W. Bressloff, M. Cibis, F. H. Gijsen, S. Cito, J. Pallareacutes, L. D. Browne, J. A. Costelloe, A. G. Lynch, J. Degroote, J. Vierendeels, W. Fu, A. Qiao, S. Hodis, D. F. Kallmes, H. Kalsi, Q. Long, V. O. Kheyfets, E. A. Finol, K. Kono, A. M. Malek, A. Lauric, P. G. Menon, K. Pekkan, M. Esmaily Moghadam, A. L. Marsden, M. Oshima, K. Katagiri, V. Peiffer, Y. Mohamied, S. J. Sherwin, J. Schaller, L. Goubergrits, G. Usera, M. Mendina, K. Valen-Sendstad, D. F. Habets, J. Xiang, H. Meng, Y. Yu, G. E. Karniadakis, N. Shaffer, and F. Loth. Variability of computational fluid dynamics solutions for pressure and flow in a giant aneurysm: the ASME 2012 summer bioengineering conference CFD challenge. *Journal of Biomechanical Engineering*, 135(2):13, 2013.

[69] S. F. C. Stewart, E. G. Paterson, G. W. Burgreen, P. Hariharan, M. Giarra, V. Reddy, S. W. Day, K. B. Manning, S. Deutsch, M. R. Berman, M. R. Myers, and R. A. Malinauskas. Assessment of CFD performance in simulations of an idealized medical device: Results of FDA's first computational interlaboratory study. *Cardiovascular Engineering and Technology*, 3(2), 2012.

[70] Q. Sun, A. Groth, and T. Aach. Comprehensive validation of computational fluid dynamics simulations of in-vivo blood flow in patient-specific cerebral aneurysms. *Medical Physics*, 39(2):742–754, 2012.

[71] L. Taelman. *Fluid-structure interaction in (repaired) aortic coarctation.* PhD thesis, Ghent University, 2014.

[72] C. A. Taylor and J. D. Humphrey. Open problems in computational vascular biomechanics: Hemodynamics and arterial wall mechanics. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3514–3523, 2009.

[73] K. F. Tchon, C. Hirsch, and R. Schneiders. Octree-based hexahedral mesh generation for viscous flow simulations. Technical Report A97-32470, American Institute of Aeronautics and Astronautics, 1997.

[74] B. Trachet. *The Role of Disturbed Hemodynamics in Aneurysm Formation in Mice.* PhD thesis, Ghent University, 2012.

[75] B. Trachet, J. Bols, G. De Santis, S. Vandenberghe, B. Loeys, and P. Segers. The impact of simplified boundary conditions and aortic arch inclusion on CFD simulations in the mouse aorta: a comparison with mouse-specific reference data. *Journal of Biomechanical Engineering*, 133(12):121006(13), 2011.

[76] B. Trachet, M. Renard, G. De Santis, S. Staelens, J. De Backer, L. Antiga, B. Loeys, and P. Segers. An integrated framework to quantitatively link mouse-specific hemodynamics to aneurysm formation in angiotensin II-infused ApoE -/- mice. *Annals of Biomedical Engineering*, 39(9):2430–2444, 2011.

[77] B. Trachet, M. Renard, C. Van der Donckt, S. Deleye, J. Bols, G.R.Y. De Meyer, S. Staelens, B.L. Loeys, and P. Segers. Longitudinal follow-up of ascending versus abdominal aortic aneurysm formation in angiotensin II-infused ApoE -/- mice. *Artery Research*, 8(1):16 – 23, 2014.

[78] A. van der Horst, C. N. van den Broek, F. N. van de Vosse, and M. C. M. Rutten. The fiber orientation in the coronary arterial wall at physiological loading evaluated with a two-fiber constitutive model. *Biomechanics and Modeling in Mechanobiology*, 11(3-4):533–542, 2012.

[79] P. van Ooij, A. Guedon, C. Poelma, J. Schneiders, M. C. M. Rutten, H. A. Marquering, C. B. Majoie, E. vanBavel, and A. J. Nederveen. Complex flow patterns in a real-size intracranial aneurysm phantom: phase contrast MRI compared with particle image velocimetry and computational fluid dynamics. *NMR in Biomedicine*, 25(1):14–26, 2012.

[80] C. S. Verma, P. F. Fischer, S.E. Lee, and F. Loth. An all-hex meshing strategy for bifurcation geometries in vascular flow simulation. In *In 14th International Meshing Roundtable*, pages 363 – 375, 2005.

[81] H. K. Versteeg and M. Malalasekera. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*. Pearson Education Limited, 2007.

[82] S. Vinchurkar and P. W. Longest. Evaluation of hexahedral, prismatic and hybrid mesh styles for simulating respiratory aerosol dynamics. *Computers & Fluids*, 37(3):317 – 331, 2008.

[83] The Vascular Modeling Toolkit (vmtk). vmtk is a collection of libraries and tools for 3D reconstruction, geometric analysis, mesh generation and surface data analysis for image-based modeling of blood vessels. `http://www.vmtk.org`.

[84] H. M. Wang, X. Y. Luo, H. Gao, R. W. Ogden, B. E. Griffith, C. Berry, and T. J. Wang. A modified holzapfel-ogden law for a residually stressed finite strain model of the human left ventricle in diastole. *Biomechanics and Modeling in Mechanobiology*, 13(1):99–113, 2014.

[85] J. Wendt. *Computational Fluid Dynamics: An Introduction*. Springer Berlin Heidelberg, 2009.

[86] S. E. Whitesall, J. B. Hoff, A. P. Vollmer, and L. G. D'Alecy. Comparison of simultaneous measurement of mouse systolic arterial blood pressure by radiotelemetry and tail-cuff methods. *American Journal of Physiology - Heart and Circulatory Physiology*, 286(6):H2408–H2415, 2004.

[87] T. Young. Hydraulic investigations, subservient to an intended croonian lecture on the motion of the blood. *Philos. Trans. Roy. Soc. (London)*, 98: 164–186, 1808.

[88] B.A. Younis, S. Spring, S. O. Neumann, and B. Weigand. Simulation of flow in an exact replica of a diseased human carotid artery. *Applied Mathematical Modelling*, 31(11):2599 – 2609, 2007.

[89] Y. Zhang, Y. Bazilevs, S. Goswami, C. L. Bajaj, and T. J. R. Hughes. Patient-specific vascular NURBS modeling for isogeometric analysis of blood flow. *Computer Methods in Applied Mechanics and Engineering*, 196(29-30):2943 – 2959, 2007.

[90] Z. Zhang, C. Kleinstreuer, and C. S. Kim. Flow structure and particle transport in a triple bifurcation airway model. *Journal of Fluids Engineering*, 123 (2):320–330, 2000.

[91] X. Zhao, D. Ho, S. Gao, C. Hong, D. E. Vatner, and S. F. Vatner. *Arterial Pressure Monitoring in Mice*. John Wiley & Sons, Inc., 2011.