



# Development of a centerline-based arterial network modeling and meshing framework for hemodynamic studies of cerebrovascular pathologies

Méghane Decroocq

## ► To cite this version:

Méghane Decroocq. Development of a centerline-based arterial network modeling and meshing framework for hemodynamic studies of cerebrovascular pathologies. Discrete Mathematics [cs.DM]. INSA de Lyon; Tohoku gakuin university (Sendai, Japon), 2023. English. NNT: 2023ISAL0034 . tel-04248316

**HAL Id: tel-04248316**

**<https://theses.hal.science/tel-04248316>**

Submitted on 18 Oct 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



N° d'ordre NNT : 2023ISAL0034

**THESE de DOCTORAT DE L'INSA LYON,  
membre de l'UNIVERSITE DE LYON**

délivré en partenariat international avec  
**L'Université de Tohoku**

**Ecole Doctorale N° 512  
Infomaths**

**Spécialité / discipline de doctorat :  
Informatique et applications**

Soutenue publiquement le 31/05/2023, par :  
**Méghane Decroocq**

---

**Development of a centerline-based arterial  
network modeling and meshing framework for  
hemodynamic studies of cerebrovascular  
pathologies**

-----

**Développement d'une méthode de modélisation  
et maillage de réseaux artériels basée sur la  
ligne centrale pour les études hémodynamiques  
des pathologies cérébrovasculaires**

---

Devant le jury composé de :

Bechmann, Dominique Professeure Université de Strasbourg Rapporteuse  
Passat, Nicolas Professeur Université de Reims Champagne-Ardenne Rapporteur  
Vignon-Clementel, Irène Directrice de recherche INRIA Saclay Examinatrice  
Chopard, Bastien Professeur Université de Genève Examineur  
Cho, Tae-Hee Professeur - Practicien hospitalier Hospices Civils de Lyon Invité  
Lavoué, Guillaume Professeur Ecole Centrale de Lyon Directeur de thèse  
Frindel, Carole MCU INSA Lyon Co-Directrice de thèse  
Ohta, Makoto Professeur Tohoku University Co-Directeur de thèse



## Département FEDORA – INSA Lyon - Ecoles Doctorales

SIGLE	ECOLE DOCTORALE	NOM ET COORDONNEES DU RESPONSABLE
<b>CHIMIE</b>	<b>CHIMIE DE LYON</b> <a href="https://www.edchimie-lyon.fr">https://www.edchimie-lyon.fr</a> Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage secretariat@edchimie-lyon.fr	<b>M. Stéphane DANIELE</b> C2P2-CPE LYON-UMR 5265 Bâtiment F308, BP 2077 43 Boulevard du 11 novembre 1918 69616 Villeurbanne <a href="mailto:directeur@edchimie-lyon.fr">directeur@edchimie-lyon.fr</a>
<b>E.E.A.</b>	<b>ÉLECTRONIQUE, ÉLECTROTECHNIQUE, AUTOMATIQUE</b> <a href="https://edeea.universite-lyon.fr">https://edeea.universite-lyon.fr</a> Sec. : Stéphanie CAUVIN Bâtiment Direction INSA Lyon Tél : 04.72.43.71.70 secretariat.edeea@insa-lyon.fr	<b>M. Philippe DELACHARTRE</b> INSA LYON Laboratoire CREATIS Bâtiment Blaise Pascal, 7 avenue Jean Capelle 69621 Villeurbanne CEDEX Tél : 04.72.43.88.63 <a href="mailto:philippe.delachartre@insa-lyon.fr">philippe.delachartre@insa-lyon.fr</a>
<b>E2M2</b>	<b>ÉVOLUTION, ÉCOSYSTÈME, MICROBIOLOGIE, MODÉLISATION</b> <a href="http://e2m2.universite-lyon.fr">http://e2m2.universite-lyon.fr</a> Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.e2m2@univ-lyon1.fr	<b>Mme Sandrine CHARLES</b> Université Claude Bernard Lyon 1 UFR Biosciences Bâtiment Mendel 43, boulevard du 11 Novembre 1918 69622 Villeurbanne CEDEX <a href="mailto:sandrine.charles@univ-lyon1.fr">sandrine.charles@univ-lyon1.fr</a>
<b>EDISS</b>	<b>INTERDISCIPLINAIRE SCIENCES-SANTÉ</b> <a href="http://ediss.universite-lyon.fr">http://ediss.universite-lyon.fr</a> Sec. : Bénédicte LANZA Bât. Atrium, UCB Lyon 1 Tél : 04.72.44.83.62 secretariat.ediss@univ-lyon1.fr	<b>Mme Sylvie RICARD-BLUM</b> Institut de Chimie et Biochimie Moléculaires et Supramoléculaires (ICBMS) - UMR 5246 CNRS - Université Lyon 1 Bâtiment Raulin - 2ème étage Nord 43 Boulevard du 11 novembre 1918 69622 Villeurbanne Cedex Tél : +33(0)4 72 44 82 32 <a href="mailto:sylvie.ricard-blum@univ-lyon1.fr">sylvie.ricard-blum@univ-lyon1.fr</a>
<b>INFOMATHS</b>	<b>INFORMATIQUE ET MATHÉMATIQUES</b> <a href="http://edinfomaths.universite-lyon.fr">http://edinfomaths.universite-lyon.fr</a> Sec. : Renée EL MELHEM Bât. Blaise PASCAL, 3e étage Tél : 04.72.43.80.46 infomaths@univ-lyon1.fr	<b>M. Hamamache KHEDDOUCI</b> Université Claude Bernard Lyon 1 Bât. Nautibus 43, Boulevard du 11 novembre 1918 69 622 Villeurbanne Cedex France Tél : 04.72.44.83.69 <a href="mailto:hamamache.kheddouci@univ-lyon1.fr">hamamache.kheddouci@univ-lyon1.fr</a>
<b>Matériaux</b>	<b>MATÉRIAUX DE LYON</b> <a href="http://ed34.universite-lyon.fr">http://ed34.universite-lyon.fr</a> Sec. : Yann DE ORDENANA Tél : 04.72.18.62.44 yann.de-ordenana@ec-lyon.fr	<b>M. Stéphane BENAYOUN</b> Ecole Centrale de Lyon Laboratoire LTDS 36 avenue Guy de Collongue 69134 Ecully CEDEX Tél : 04.72.18.64.37 <a href="mailto:stephane.benayoun@ec-lyon.fr">stephane.benayoun@ec-lyon.fr</a>
<b>MEGA</b>	<b>MÉCANIQUE, ÉNERGÉTIQUE, GÉNIE CIVIL, ACOUSTIQUE</b> <a href="http://edmega.universite-lyon.fr">http://edmega.universite-lyon.fr</a> Sec. : Stéphanie CAUVIN Tél : 04.72.43.71.70 Bâtiment Direction INSA Lyon mega@insa-lyon.fr	<b>M. Jocelyn BONJOUR</b> INSA Lyon Laboratoire CETHIL Bâtiment Sadi-Carnot 9, rue de la Physique 69621 Villeurbanne CEDEX <a href="mailto:jocelyn.bonjour@insa-lyon.fr">jocelyn.bonjour@insa-lyon.fr</a>
<b>ScSo</b>	<b>ScSo*</b> <a href="https://edsciencessociales.universite-lyon.fr">https://edsciencessociales.universite-lyon.fr</a> Sec. : Mélina FAVETON INSA : J.Y. TOUSSAINT Tél : 04.78.69.77.79 melina.faveton@univ-lyon2.fr	<b>M. Bruno MILLY</b> Université Lumière Lyon 2 86 Rue Pasteur 69365 Lyon CEDEX 07 <a href="mailto:bruno.milly@univ-lyon2.fr">bruno.milly@univ-lyon2.fr</a>

\*ScSo : Histoire, Géographie, Aménagement, Urbanisme, Archéologie, Science politique, Sociologie, Anthropologie

---

# Contents

Abstract (English) . . . . .	
Abstract (French) . . . . .	1
<b>CHAPTER 1 - Introduction</b>	<b>2</b>
1.1 Anatomy of the cerebral arterial network . . . . .	3
1.2 Cerebrovascular pathologies . . . . .	4
1.2.1 Stenosis . . . . .	5
1.2.2 Cerebral aneurysm . . . . .	6
1.3 Computational fluid dynamics . . . . .	7
1.3.1 Fluid characterization . . . . .	7
1.3.2 Flow dynamics governing equations . . . . .	7
1.3.3 Numerical methods . . . . .	8
1.3.4 Boundary conditions . . . . .	9
1.3.5 Convergence criterion . . . . .	9
1.4 Application to cerebrovascular pathologies . . . . .	9
1.5 Meshing for CFD . . . . .	10
1.5.1 Medical imaging . . . . .	11
1.5.2 Segmentation-based meshing . . . . .	12
1.5.3 Centerline-based meshing . . . . .	14
1.5.4 Mesh quality . . . . .	15
1.6 Databases . . . . .	17
1.7 Contributions of this thesis . . . . .	18

<b>CHAPTER 2 - Modeling</b>	<b>20</b>
2.1 Introduction . . . . .	20
2.2 State-of-the-art . . . . .	21
2.3 Vessels . . . . .	23
2.3.1 Splines . . . . .	23
2.3.2 Control points optimization . . . . .	24
2.3.3 Smoothing parameter optimization . . . . .	25
2.3.4 Two-step approximation strategy . . . . .	26
2.4 Bifurcations . . . . .	28
2.4.1 Model description . . . . .	28
2.4.2 Parameter estimation from centerlines . . . . .	29
2.4.3 Extension to $n$ -furcation . . . . .	30
2.4.4 Tangent continuity . . . . .	31
2.5 Validation . . . . .	33
2.5.1 Validation pipeline . . . . .	33
2.5.2 Validation dataset . . . . .	34
2.5.2.1 Ground-truth centerlines . . . . .	34
2.5.2.2 Deterioration operations . . . . .	34
2.5.3 Approximation methods . . . . .	37
2.5.4 Quality metrics . . . . .	38
2.5.5 Results . . . . .	40
2.5.5.1 Smoothing criteria comparison . . . . .	40
2.5.5.2 Approximation methods comparison . . . . .	41
2.6 Conclusion . . . . .	45
<b>CHAPTER 3 - Meshing</b>	<b>46</b>
3.1 Introduction . . . . .	46
3.2 State-of-the-art . . . . .	47
3.3 Bifurcations . . . . .	50
3.3.1 Decomposition . . . . .	50
3.3.2 Surface meshing . . . . .	51
3.3.3 Relaxation . . . . .	53

3.3.4	Apex smoothing . . . . .	54
3.3.5	Extension to $n$ -furcations . . . . .	56
3.4	Vessels . . . . .	57
3.5	Volume meshing . . . . .	58
3.6	Validation . . . . .	58
3.6.1	Meshing success rate . . . . .	58
3.6.2	Mesh quality . . . . .	61
3.6.3	Computational time . . . . .	63
3.7	Conclusion . . . . .	64
<b>CHAPTER 4 - Comparison with segmentation-based methods</b>		<b>66</b>
4.1	Introduction . . . . .	66
4.2	State-of-the-art . . . . .	67
4.3	Surface mesh comparison . . . . .	68
4.3.1	Comparison pipeline . . . . .	68
4.3.2	Datasets and model training . . . . .	70
4.3.3	Quantitative evaluation . . . . .	71
4.3.3.1	Topological and geometrical features . . . . .	71
4.3.3.2	Results . . . . .	72
4.3.4	Visual evaluation . . . . .	73
4.3.4.1	Centerline extraction and post-treatment . . . . .	73
4.3.4.2	Results . . . . .	73
4.4	Volume mesh comparison . . . . .	76
4.4.1	Tube model . . . . .	76
4.4.1.1	Geometry . . . . .	77
4.4.1.2	Tetrahedral meshing . . . . .	77
4.4.1.3	Hexahedral meshing . . . . .	78
4.4.1.4	Mesh quality . . . . .	79
4.4.1.5	CFD simulations . . . . .	80
4.4.1.6	Mesh independence . . . . .	81
4.4.1.7	Simulation accuracy . . . . .	83
4.4.2	Bifurcation model . . . . .	85

4.4.2.1	Geometry . . . . .	85
4.4.2.2	Meshing . . . . .	86
4.4.2.3	Mesh independence . . . . .	88
4.4.2.4	Wall shear stress . . . . .	90
4.5	Conclusion . . . . .	92
<b>CHAPTER 5 - Editing framework</b>		<b>95</b>
5.1	Introduction . . . . .	95
5.2	State-of-the-art . . . . .	95
5.3	Software overview . . . . .	97
5.3.1	Input data . . . . .	97
5.3.2	Main functionalities . . . . .	97
5.3.3	Interface layout . . . . .	98
5.3.4	Spatial graph encoding . . . . .	99
5.4	Editing functionalities . . . . .	100
5.4.1	Centerlines . . . . .	100
5.4.1.1	Visualization . . . . .	100
5.4.1.2	Data point editing . . . . .	101
5.4.1.3	Branch removal . . . . .	101
5.4.1.4	Branch angle . . . . .	102
5.4.2	Model . . . . .	102
5.4.2.1	Visualization . . . . .	102
5.4.2.2	Vessel trajectory edition . . . . .	102
5.4.3	Mesh . . . . .	103
5.4.3.1	Visualization . . . . .	103
5.4.3.2	Quality check . . . . .	103
5.5	Medical image superimposition . . . . .	104
5.6	Pathology modeling . . . . .	104
5.6.1	Aneurism . . . . .	104
5.6.2	Stenosis . . . . .	106
5.7	Applications . . . . .	107
5.7.1	Open database of cerebral arterial network meshes . . . . .	107

5.7.1.1	Editing method . . . . .	108
5.7.1.2	Database . . . . .	110
5.7.2	3D printing . . . . .	112
5.7.3	CFD in a stenotic vessel case . . . . .	113
5.8	Conclusion . . . . .	116
<b>CHAPTER 6 - Conclusion</b>		<b>117</b>
	Appendix . . . . .	119
	Communications . . . . .	124
	Acknowledgments . . . . .	126
<b>Bibliography</b>		<b>128</b>

---

# List of Figures

1.1	Anatomy of the cerebral vascular network. . . . .	4
1.2	Ischemic stroke origin and outcome. . . . .	5
1.3	Different types of cerebral aneurysms. . . . .	6
1.4	Pipeline from medical image to CFD ready mesh. . . . .	12
1.5	Challenges related to segmentation-based meshing. . . . .	13
1.6	Centerline representation of the vascular network. . . . .	14
1.7	Different types of mesh . . . . .	16
1.8	Scaled Jacobian values for different hexahedral cells. . . . .	17
1.9	MRA images and centerline extracted for one patient of the BraVa database. .	18
2.1	Example of spline basis functions. . . . .	24
2.2	Approximation splines with different values of smoothing parameter $\lambda$ . . . . .	25
2.3	Approximation of a noisy centerline with the proposed method. . . . .	27
2.4	Five cross-sections bifurcation model. . . . .	29
2.5	Pipeline of the bifurcation parameter estimation. . . . .	30
2.6	8 cross-section model of trifurcation generalized from the bifurcation model of [93]. . . . .	31
2.7	Illustration of the end constraints applied in the vessel approximation to maintain the continuity of the model. . . . .	32
2.8	Approximation strategy evaluation pipeline. . . . .	33
2.9	Four ground-truth centerlines created for this study. . . . .	34
2.10	Deterioration operations applied to the centerlines. . . . .	36

2.11	Matching time parameters of the splines by minimum distance projection. . . .	38
2.12	Comparison of five different optimization criteria for the smoothing parameter used for the penalized spline approximation. . . . .	41
2.13	Mesh resulting from the approximation of distorted data by three of the meth- ods compared in this evaluation. . . . .	42
2.14	Extended results of the approximation model validation. . . . .	44
3.1	Hexahedral mesh of airways generated by the method of Viville et al. . . . .	48
3.2	Geometric decomposition of the bifurcation model. . . . .	51
3.3	Computation of the nodes of the end cross-sections and the separation planes. .	52
3.4	Initial surface mesh and mesh after projection for two types of initialization. .	53
3.5	Relaxation of the bifurcation mesh. . . . .	54
3.6	Apex smoothing pipeline. . . . .	55
3.7	Apex smoothing with different radii of curvature. . . . .	56
3.8	Mesh of a trifurcation. . . . .	56
3.9	Rotation applied to the mesh referential to connect bifurcation to ends and bifurcations to bifurcations. . . . .	57
3.10	O-grid pattern and volume meshing method. . . . .	58
3.11	Top view of 4 meshes among the 60 generated from the patients of the BraVa database. . . . .	59
3.12	Top view of 4 meshes among the 60 generated from the patients of the BraVa database. . . . .	60
3.13	Example of success and failure of our method for one patient of the BraVa database. . . . .	61
3.14	Distribution of the scaled Jacobian values of the mesh cells. . . . .	62
4.1	Pipeline used to compare segmentation-based meshing and centerline-based meshing. . . . .	69
4.2	Principle of convolutional neural network-based image segmentation. . . . .	70
4.3	Meshes produced with segmentation-based and centerline-based methods for a patient of the BraVa database. . . . .	75



4.4	Enhanced visualization of the segmentation-based mesh, the centerline-based mesh obtained with our method and the centerline-based mesh obtained with VessMorphoVis. . . . .	76
4.5	Meshes produced for the CFD study of the tube model. . . . .	77
4.6	Meshing time as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	80
4.7	Graph of the sectional maximal velocity as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. . . . .	81
4.8	Number of iterations until convergence as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	82
4.9	Simulation time as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	83
4.10	Velocity field and velocity profile for a cross-section of the tube model for coarse and fine tetrahedral and hexahedral meshes. . . . .	84
4.11	$RE_{mean}$ and $RE_{max}$ relative error values as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	85
4.12	The bifurcation geometry used for the CFD simulations, with flow extensions. .	86
4.13	Meshes produced for the CFD study of the bifurcation model. . . . .	87
4.14	Meshing time as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	88
4.15	Number of iterations until convergence as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	89
4.16	Simulation time as a function of the number of cells for tetrahedral and hexahedral meshes. . . . .	89
4.17	Graph of the sectional maximal velocity as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. . . . .	90
4.18	Wall shear stress values computed by CFD for hexahedral and tetrahedral meshes.	91
4.19	Graph of the mean wall shear stress as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. . . . .	92
5.1	Overview of functionalities of the software. . . . .	98
5.2	The interface layout. . . . .	99

5.3	The 4 graphs used to encode the vascular network, with the edge and node attributes. . . . .	100
5.4	Structured hexahedral meshing of cerebral arteries with a fusiform or saccular aneurysm by deformation. . . . .	105
5.5	Cross-section pattern before and after deformation. . . . .	106
5.6	Design of a stenosis in a vascular network. . . . .	107
5.7	Software use case for the manual correction of a vascular network of the BraVa database. . . . .	109
5.8	Database of 10 meshes of the large cerebral arterial network produced for the centerline of the BraVa database using the meshing method and software developed in this thesis. . . . .	112
5.9	3D printing of a whole brain arterial network model. . . . .	113
5.10	Comparison of the segmentation-based and centerline-based mesh. . . . .	114
5.11	CFD simulation results for the mesh without stenosis and with stenosis. . . . .	115
5.12	Velocity along the centerline for the different branch of the MCA territory model in the healthy case and the stenosis case. . . . .	115

---

# List of Tables

1.1	Advantages and disadvantages of segmentation-based meshing method. . . . .	13
1.2	Advantages and disadvantages of centerline-based representation. . . . .	15
2.1	Parameters used for the distortion of the ground truth centerlines . . . . .	35
2.2	Overall evaluation of the approximation methods: mean values of the quality criteria for all the centerlines distorted by radius noise addition. The cells in gray corresponds to the lowest error for each metric. . . . .	43
2.3	Overall evaluation of the approximation methods: mean values of the quality criteria for all the centerlines distorted by spatial noise addition. . . . .	43
3.1	Computational time required to model and mesh large vascular networks. . . .	64
4.1	Description of the different datasets used in our comparison study. . . . .	71
4.2	topological and geometrical features of the segmentation-based meshes for the different datasets. . . . .	72
4.3	Table of the properties of the tetrahedral and hexahedral mesh generated. . . .	79
4.4	Table of the properties of the tetrahedral and hexahedral mesh generated. . .	86

---

# Abstract (English)

Computational fluid dynamics (CFD) is a technique that provides valuable information on blood flow from the vascular geometry, helping to understand, diagnose and predict the outcome of vascular diseases. However, the resolution of current medical images is insufficient to obtain meshes matching the requirements of CFD, especially for large and complex vascular networks such as the cerebral arterial network. In this thesis, we propose a two-step framework to produce CFD-ready meshes from a simplified representation of vascular networks by their centerlines. In the first stage, to address the shortcomings of the centerline-based representation (scattered, noisy), a modeling step was introduced to reconstruct an anatomically realistic model from a-priori knowledge of the vessels and bifurcation geometries. Next, a meshing step was developed to create a high-quality volume mesh with structured, flow-oriented hexahedral cells that meets the requirements of CFD. We showed that the proposed method solves some of the shortcomings of the meshes obtained from segmentation methods of the literature (e.g. merging, disconnected vessels), and leads to more accurate CFD results for less computational cost. This meshing framework was completed by a user interface facilitating the visualization and editing of the vascular network models, and opening our code to non-expert users. Using this interface, we have created an open dataset of meshes for CFD of whole cerebral artery networks that can be used for the evaluation of medical devices and hemodynamic studies.

---

# Abstract (French)

La dynamique des fluides numérique (CFD) est une technique qui fournit des informations précieuses sur l'écoulement du sang à partir de la géométrie vasculaire, ce qui permet de comprendre, de diagnostiquer et de prédire l'issue des maladies vasculaires. Cependant, la résolution des images médicales actuelles n'est pas satisfaisante pour obtenir des maillages répondant aux exigences de la CFD, en particulier pour des réseaux vasculaires larges et complexes telles que les réseaux d'artères cérébrales. Dans cette thèse, nous proposons une méthode en deux étapes pour produire des maillages adaptés à la CFD à partir d'une représentation simplifiée des réseaux vasculaires par leurs lignes centrales. Dans la première étape, pour remédier aux défauts de la représentation basée sur les lignes centrales (dispersées, bruitées), une étape de modélisation a été introduite pour reconstruire un modèle anatomique réaliste à partir d'a-priori sur la géométrie des vaisseaux et des bifurcations. Ensuite, une étape de maillage a été développée pour créer un maillage volumique de bonne qualité avec des cellules hexaédriques structurées et orientées selon l'écoulement du fluide qui répondent aux exigences de la CFD. Nous avons montré que la méthode proposée résout certaines des lacunes des maillages obtenus à partir des méthodes de segmentation de la littérature (e.g. vaisseaux fusionnés ou déconnectés), et conduit à des résultats de CFD plus précis pour un coût de calcul moindre. Cette méthode de maillage a été complétée par une interface utilisateur facilitant la visualisation et l'édition des modèles de réseaux vasculaires, et ouvrant notre code à des utilisateurs non experts. Grâce à cette interface, nous avons créé une base de données de maillages de réseaux entiers d'artères cérébrales qui pourra être utilisée pour l'évaluation de dispositifs médicaux et les études hémodynamiques.

---

# Introduction

Cerebrovascular diseases, such as stroke, can cause severe disability or death [67]. The relationship between the topology and geometry of the vascular network and the onset and the outcome of the disease is increasingly investigated in the literature. Computational fluid dynamics is a key tool for this type of study, as it provides information on the hemodynamics (blood velocity, pressure) from the vessel geometry [73, 78]. The main limitation of the use of CFD is the creation of the computational mesh. Indeed, numerical simulation requires a smooth, anatomically realistic mesh of the arterial wall to provide reliable results. In pathologies like ischemic stroke, the distribution of the vessels in the different vascular territories of the brain impacts the position and evolution of the lesion [19, 35]. It requires reconstructing large and complex cerebral arterial networks with small vessels whose radius is close to the image resolution, which remains very challenging. Besides, in the finite element method, the shape of the cells inside the volume also affects the simulations. Although flow-oriented, structured hexahedral cells have been shown to improve the stability of the simulation while lowering the computational cost [85, 18, 25], tetrahedral cells remain widely used due to their ability to automatically mesh any complex shape. The approaches investigated in the literature to address the meshing of vascular networks can be divided into two categories: the segmentation-based and the centerline-based methods.

The segmentation of magnetic resonance angiography (MRA) images is a non-invasive way to access patient-specific vasculature. A lot of effort was put into developing efficient vessel-enhancing filters [46, 56] and improving the segmentation methods [51, 59, 80]. However, the accuracy of the segmentation does not guarantee the accuracy of the mesh

it entails (e.g. vessels merging due to the image resolution, disconnected vessels, bumps), nor its usability for numerical simulation. Following the tubularity assumption, vessels can be reduced to a centerline-radius description. Unlike image segmentation, the centerline-based representation incorporates the topology of the network in its structure and allows for manual extraction and editing. This simplified representation is more suitable for the construction of big databases of large vascular networks [89] and the creation of high-quality hexahedral meshes. Nevertheless, the representation of vessels by centerlines lowers the geometrical information content, leading to inaccuracies in the shape of the vessels and the position of bifurcations. These limitations make it difficult to reconstruct a smooth and anatomically accurate surface model that matches the requirements of numerical simulation.

In this context, we believe that bridging the gap between the centerline representation of vascular networks and CFD meshes would enable us to take advantage of the many assets of this representation type to run CFD studies in larger vascular networks. The objective of this thesis is to develop a new method to generate high-quality meshes ready for CFD from centerlines alone. The main challenges to overcome are the limitation inherent to centerline-based representation (data sparsity and uncertainty), and the generation of a high-quality hexahedral mesh fulfilling the requirement for CFD.

## **1.1 Anatomy of the cerebral arterial network**

The cerebral arterial network is the network of arteries that supplies blood to the brain. This network can be divided into different territories, arising from the main arteries of the circle of Willis, as shown in Figure 1.1. Three main arteries arise from the neck; the left and right internal carotid arteries (ICA) and the basilar artery (BA) which is formed by the fusion of the right and left vertebral arteries (VA). Those three arteries are connected by the communicating arteries. The right and left posterior communicating arteries (Pcom) connect each ICA to the BA and the anterior communicating artery (Acom) connects both ICAs, forming the only vascular circle in the cerebral arterial network: the circle of Willis. The arteries downstream of this circle - the right and left middle carotid arteries (MCA) and the anterior cerebral artery (ACA) - supply various parts of the brain.

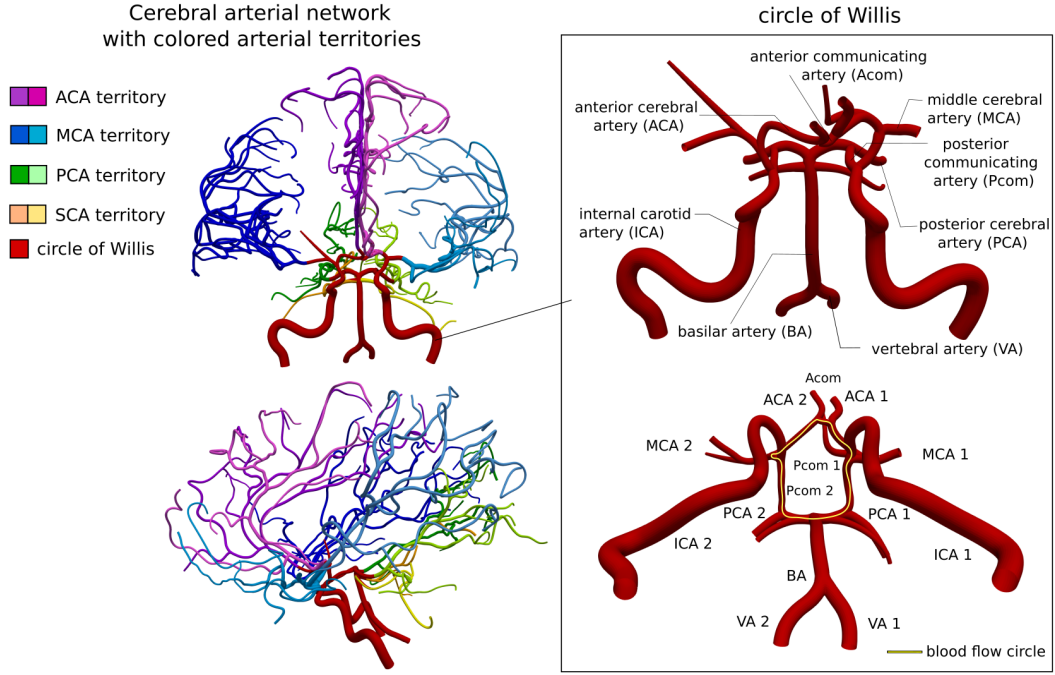


Figure 1.1: Anatomy of the cerebral vascular network. The left panel shows the different arterial territories and the right panel enhances the circle of Willis.

The circle of Willis can be categorized into different variations, as the topology varies between patients [40, 45]. The typical pattern of the complete circle of Willis as shown in figure 1.1 represents around 10% of the occurrences, the rest of the specimens presenting different variations, including the absence of one or several communicating arteries, cause the circle to be incomplete [45].

## 1.2 Cerebrovascular pathologies

Cerebrovascular pathologies are diseases that affect the cerebral arterial network. Especially, stroke remains one of the main causes of death and can lead to severe disability. In 2019, stroke was the second main cause of death and disability worldwide, with 6.6 million death, with 143 million people with stroke-related disability [22]. In 30 years, stroke incidence, prevalence, and mortality increased drastically [22], making stroke one of the major public health stakes. A stroke occurs when the supply of blood in the brain tissues is insufficient, causing the death of these tissues. There are two main types of stroke; ischemic stroke, which occurs when one of the cerebral arteries is obstructed, causing a lack



of blood flow, and hemorrhagic stroke, caused by internal bleeding following the rupture of the arterial wall.

### 1.2.1 Stenosis

The vast majority of stroke falls into the category of ischemic strokes. Ischemic stroke is triggered by arteriosclerosis, which corresponds to the formation of a plaque on the wall of one of the cerebral arteries. This plaque leads to the narrowing of the artery called stenosis. This change in the arterial geometry triggers a change of blood flow pattern that eventually will lead to the creation of a blood clot obstructing the artery. The acute phase of ischemic stroke requires immediate care.

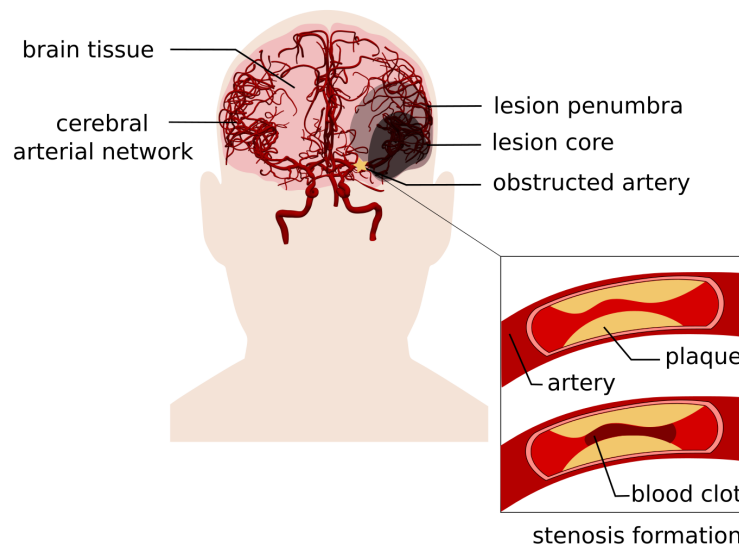


Figure 1.2: Ischemic stroke origin and outcome.

To treat ischemic stroke, the clot can be removed either by dissolving the blot clot by an intravenous injection of lysing drugs (thrombolysis) or by removing the clot mechanically by inserting a catheter in the arterial circulation from the femoral artery until it reaches the occluded vessel (thrombectomy). After this surgical procedure, the part of the lesion where the tissue damage is still reversible, called the penumbra, might recover. The size of the final lesion after the procedure can hardly be predicted [20], as it depends on many factors, including the presence of collateral flow (i.e. blood supply from the peripheral arteries to the lesion core). The clot is then removed by aspiration or using a stent-retriever device. Once the emergency procedure of clot removal is performed, the stenotic

artery is held open using a stent, also inserted using a catheter. The success of the catheter and stent deployment depends on the geometry and topology of the arterial network, the shape of the stenosis, and the blood flow pattern before and after implantation.

### 1.2.2 Cerebral aneurysm

A cerebral aneurysm is a dilatation of the wall of a cerebral artery causing the appearance of a bulge or swelling in the vascular geometry. There are two main types of aneurysm, as represented in Figure 1.3; the saccular aneurysm has a bulging geometry, while the fusiform aneurysm is an abnormal swelling of the arterial wall. Aneurysm can be located either on an arterial segment or on a bifurcation and can have various sizes and shapes (one or several lobes). The local blood flow pattern might be at the origin of this pathology, and the particular blood flow pattern created can also lead to the rupture of the aneurysm wall, although this event remains difficult to predict. Aneurysm rupture leads to life-threatening internal bleeding requiring surgery. Devices such as flow diversion stents or coils can be used as a prevention measure against aneurysm rupture. Endovascular coiling is a surgical procedure that consists in filling the inside of the aneurysm with a flexible wire. Flow diversion stents are endovascular prostheses that are deployed in the artery at the aneurysm inlet location using a catheter. This stent will cover the neck of saccular aneurysms in order to divert the blood flow, causing progressive thrombosis of the aneurysm.

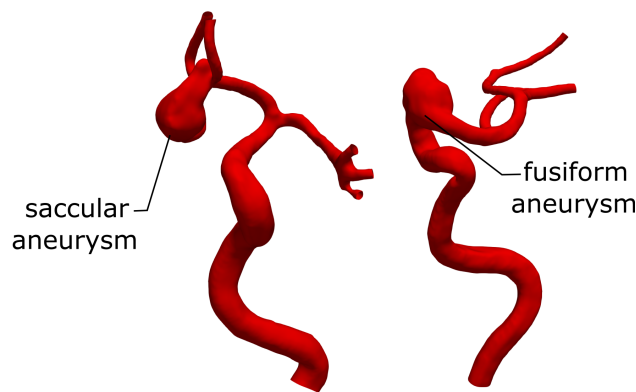


Figure 1.3: Different types of cerebral aneurysms on the internal carotid arteries. The geometries represented in this figure were extracted from the Aneurisk database [3].

## 1.3 Computational fluid dynamics

### 1.3.1 Fluid characterization

Fluids can be categorized into various types. An incompressible fluid ( $\neq$  compressible fluid) is a fluid for which the density  $\rho$  is constant in space and time. Blood is an incompressible fluid. A fluid is said non-Newtonian (resp. Newtonian) if its viscosity depends (resp. does not depend) on the shear rate. The flow regime can also be divided into laminar flow, where the flow, dominated by viscous effects, is ordered, and turbulent flow, dominated by inertial effect where the flow is more chaotic, creating patterns such as vortices or swirls. The flow regime can be predicted by the Reynolds number  $Re$ . The Reynolds number of a fluid flowing in a pipe is written

$$Re = \frac{uD}{\nu}, \quad (1.1)$$

where  $\nu$  is the kinematic viscosity in  $\text{m}^2.\text{s}^{-1}$ ,  $D$  is the diameter of the pipe in m and  $u$  is the fluid velocity in  $\text{m}.\text{s}^{-1}$ . Under a steady flow,  $Re < 2270$  corresponds to a laminar flow,  $2270 < Re < 2450$  is the transition phase and  $Re > 2450$  corresponds to a turbulent flow [92].

### 1.3.2 Flow dynamics governing equations

The velocity profile of an incompressible viscous fluid in laminar regime in a straight cylinder can be solved analytically. If the length of the tube is sufficient, the velocity sectional profile will reach a stable form called the fully developed form. In these settings, the velocity depends only on the radius position  $r$ . This velocity profile is described by the Poiseuille equation:

$$u(r) = \frac{\Delta P(R^2 - r^2)}{4 \times \mu \times L}, \quad (1.2)$$

where  $\Delta P$  is the pressure difference between inlet and outlet in,  $R$  the radius of the pipe,  $\mu$  the dynamic viscosity and  $L$  the tube length.

However, this formula can not be applied to 3D vascular models with varying radii and complex branching patterns. To study these cases, the Navier-Stokes equation, which

generalizes the Poiseuille equation to all types of fluids and geometries, must be used. The Navier-Stokes equation for an incompressible viscous fluid like blood can be written as:

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{\nabla P}{\rho} + \nu \nabla^2 u, \quad (1.3)$$

where  $u$  is the velocity of the fluid in  $\text{m.s}^{-1}$ ,  $P$  is the pressure in Pa,  $\rho$  is the density in  $\text{kg.m}^{-3}$ , and  $\nu$  is the kinematic viscosity in  $\text{m}^2.\text{s}^{-1}$ . As this equation can not be solved analytically, the resolution methods rely on numerical simulation.

### 1.3.3 Numerical methods

Several methods can be employed to solve this equation numerically, including the finite element method (FEM), the finite volume method (FVM), and the finite difference method (FDM). The resolution principle of those methods differs. FEM is based on local approximations of the solution, FVM on the average of the exact solution over a given volume, and FDM on the approximation of the derivatives. However, for all those methods, the fluid domain must be subdivided into smaller parts called finite elements. This spatial discretization of the space is encoded as a 3D volume mesh (i.e. a set of nodes and edges forming faces and cells). In CFD softwares, the finite volume method is often implemented as it can be easily applied to unstructured mesh (e.g. meshes where the cells are not organized in any particular way). The different types of cells and meshes are presented and discussed in the next section. Other methods can be used to compute fluid dynamics. The isogeometric analysis is a computational approach in which meshes are replaced by a NURBS-based representation of the geometry. The advantage of this method is that the surface is represented exactly by a continuous function, which reduces the geometrical error and the numerical errors due to dissipation errors. However, it requires building a parametric representation of the surface, which can be difficult for complex geometry. Finally, Lagrangian particle-based methods can be opposed to the Eulerian grid-based methods presented before. In these methods (Lattice-Boltzman, smoothed particle hydrodynamics), instead of solving the Navier-Stokes equation, the fluid is represented by small particles, whose motion over the lattice is computed by propagation and collision rules. In this work, we focus on the methods based on meshes, such as the finite element or finite volume methods, as these are the resolution methods employed in the most

commonly used CFD softwares (OpenFoam, ANSYS Fluent, Fenics...).

### **1.3.4 Boundary conditions**

Independently of the method used for the resolution, appropriate Boundary Conditions (BCs) must be applied to the model before running the solver. The inlet boundary condition defines the fluid behavior when flowing in the fluid domain. The outlet boundary condition defines the fluid behavior when flowing out of the fluid domain. The wall boundary condition determines how the flow reacts when it gets in contact with a solid wall. These BCs are set as constraints on the variables of the Navier-Stokes equations (velocity or pressure values). They can be constant (i.e. not varying through time), or transient. Once defined, the boundary conditions are applied to the relevant faces of the mesh.

### **1.3.5 Convergence criterion**

CFD resolution is an iterative process in which the solution will be refined progressively. To visualize the progress of the computation, the residuals are monitored. The residual value is the difference between both sides of the equation to solve. Ideally, this difference must be equal to 0: the residuals correspond to the numerical error. The final solution is reached when the residuals have converged. A threshold value of  $10^{-6}$  of residuals is required to consider that the simulation has converged. The final solution reached also depends on the mesh cell density. To guarantee the accuracy of the flow solution, we must check that the density of cells in the mesh is high enough to reach the stability of the solution. This property is called mesh independence. To find the cell density threshold where mesh independence is reached, a value of interest (velocity, wall shear stress...) is plotted against the mesh cell density. When this value becomes stable, the mesh independence can be affirmed.

## **1.4 Application to cerebrovascular pathologies**

As cerebrovascular pathologies are strongly related to blood flow, computational fluid dynamics can provide important information leading to a better understanding of the causes and outcomes of these pathologies. Blood circulation has several particularities

to take into account in CFD simulations. In this work, we focus on arteries; some of the assumptions below might not be valid for other vascular networks such as the venous system. Firstly, blood is a non-Newtonian fluid, however, in various studies of blood flow, blood is assumed to be a Newtonian fluid. This assumption is considered valid in arteries of big diameter such as the aorta but may be incorrect when examining the flow in lower arteries [54]. Secondly, the blood flow is not steady (i.e. with an inlet velocity constant over time) but pulsatile. Indeed, the heart is pumping blood to the rest of the body. As a result of this mechanism, the systole (resp. diastole) is associated with an acceleration (resp. deceleration) of blood flow, and this pattern is repeated cyclically, causing different blood flow patterns to appear. This effect can be modeled via an appropriate setting of the inlet boundary condition. As for the outlet boundary condition, the  $0Pa$  or the steady pressure outlet condition is often used for its simplicity [53]. However, this assumption is a significant simplification of the resistance of the downstream vessels. As an alternative, the outlet BC can be set to a transient pressure profile derived from the inlet BC (Windkessel model) [26], or using a 1D model of the downstream vessels [84]. Finally, the vessel wall is not rigid but elastic, causing the vessel diameter to change according to the variation of pressure along the cardiac cycle [55]. This particularity can be captured using fluid-structure interaction models (FSI) [81].

In the past decades, CFD simulations were conducted to understand the genesis and predict the outcome of vascular diseases, as well as evaluate medical devices or medical treatments. The role of blood flow, and more specifically wall shear stress, in the formation of cerebral aneurysms and the rupture events, was investigated using CFD [61]. In the same way, the impact of stenosis or occlusion on the brain tissues can also be revealed by numerical simulation [35]. Finally, the blood flow pattern before and after stent deployment in stenosis or aneurysms can be analyzed to evaluate and improve the device's efficiency [13], understand the reason for failure [72] or improve the surgical procedures.

## 1.5 Meshing for CFD

As mentioned in the previous section, to run CFD simulations in patient-specific vascular geometries, a surface mesh of the vascular geometry of interest must first be reconstructed. Then, this surface mesh must be discretized into cells to form a volume mesh.

This meshing step is crucial as the shape of the mesh will impact the simulation result, however, it is often the most burdensome and time-consuming step. In this section, we detail the method and algorithms used to reconstruct a patient-specific surface mesh from medical images, as well as the different types of volume meshes.

### **1.5.1 Medical imaging**

Noninvasive imaging techniques have played a crucial role in the in-vivo visualization of the vasculature. Computed tomography angiography (CTA) uses the different X-ray attenuation properties of tissues to show anatomical structures. A contrast agent is injected into the blood circulation, revealing the vessel's geometry and enabling to identify vascular pathologies. Digital subtraction angiography (DSA) images are obtained by subtraction of images before and after injection of contrast agent, isolating the vessels from the rest of the tissues. X-ray or MR images can be obtained in 2D or 3D dimensions and visualize the bolus course in the vessels through time. This imaging modality can provide information on the blood flow velocity. Finally, magnetic resonance angiography (MRA) is a medical imaging modality based on magnetic resonance that enables to visualize the arteries in-vivo and non-invasively. MRA relies on a variety of imaging techniques, divided into two categories; flow-based and contrast agent-based techniques. Flow-based techniques rely on the fact that the blood is flowing to distinguish it from the surrounding static tissues. Phase-contrast MRA (PC-MRA) uses the phase shifts introduced by motion in the presence of a magnetic field gradient. The intensity of the image produced is proportional to the phase shift for some range of velocities. This method enables quantitative velocity estimation. Time-of-flight MRA (TOF-MRA) is another flow-based method that uses the signal produced by unsaturated blood entering magnetically saturated surrounding tissues. TOF-MRA can produce either 2D or 3D images (composed of several 2D slices). Figure 1.4 shows the axial maximum intensity projection of a cerebral TOF-MRA image. Finally, in contrast agent-based MRA methods, a contrast agent is injected into a vein. The intensity of the resulting image depends on the contrast agent concentration, enhancing the arterial network. In this work, TOF-MRA images only were used, therefore, TOF-MRA images are referred to hereafter as MRA.

### 1.5.2 Segmentation-based meshing

The pipeline to reconstruct a 3D mesh from MRA volumes is described in Figure 1.4. The volume is first filtered to enhance the structures of interest; here the vessels. The vessels are then isolated from the rest of the image - the pixel value is 1 inside vessels and 0 outside - by a segmentation step. This segmentation is the most challenging and crucial step as it determines the shape of the surface mesh. Segmentation is still a very active field of research and many methods were designed specifically for the task of vessel segmentation. In Chapter 4, the state-of-the-art of segmentation methods is further detailed. From the segmentation, a surface mesh is produced by the Marching Cube method. This mesh is then post-processed manually or automatically to reconnect the disconnected parts, smooth the surface (Laplacian smooth, Taubin smooth), and correct the mesh defects. Finally, the inside of the surface mesh is divided into cells to form a volume mesh that will be used for the numerical simulation of blood flow.

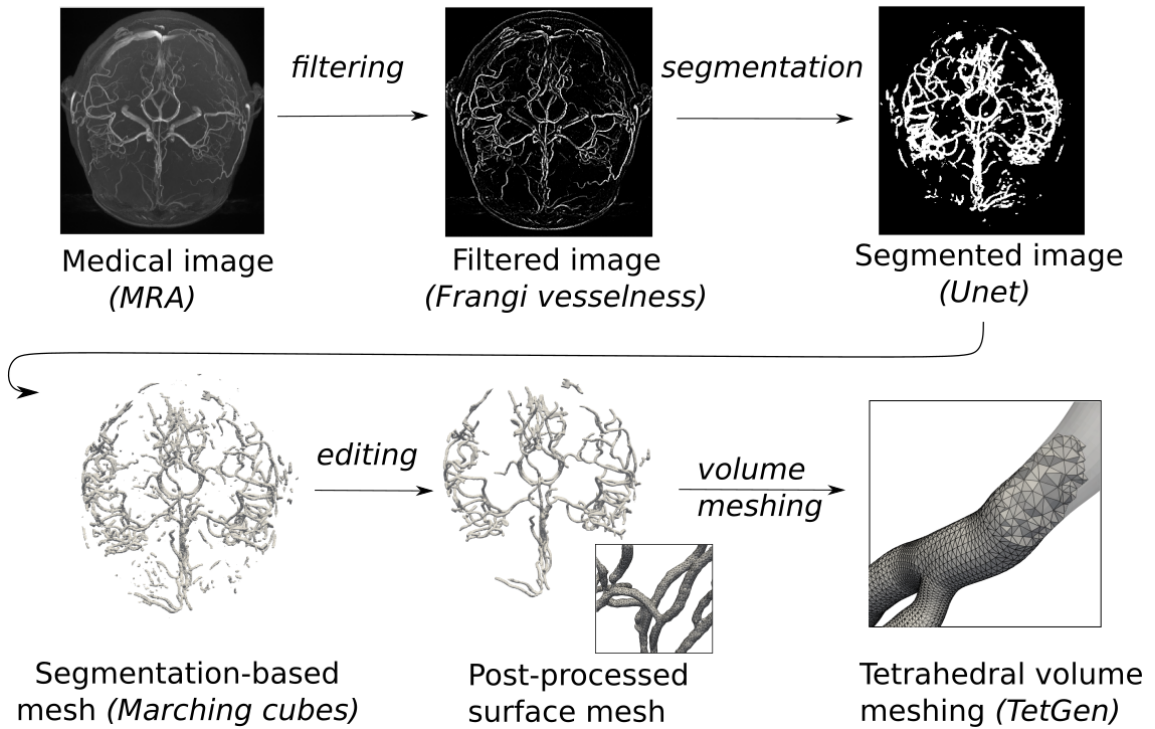


Figure 1.4: Pipeline from medical image to CFD ready mesh.

Segmentation-based meshing has many advantages; the surface mesh can be created automatically, and it captures the specificity of the patient vasculature directly from the



image. However, this method also has some limitations. Indeed, it does not guarantee the accuracy of the topology of the network. Because of the low resolution of medical images, the pixelization of the segmentation might result in merging vessels or bumps in the vessel surface, as further demonstrated in Chapter 4. This challenging aspect of segmentation is illustrated by Figure 1.5. Furthermore, the post-processing of meshes produced from segmentation is burdensome and time-consuming as it often requires modifying the mesh node-by-node or edges-by-edges. The advantages and limitations of the segmentation-based meshing method are summarized in table 1.1.

Table 1.1: Advantages and disadvantages of segmentation-based meshing method.

advantages	limitations
<ul style="list-style-type: none"> <li>• Can mesh complex branching patterns</li> <li>• Can mesh non-circular vessel shape</li> <li>• Reproduces well the patient anatomy</li> </ul>	<ul style="list-style-type: none"> <li>• Geometrical inaccuracies (bumps)</li> <li>• Topological inaccuracies (cycles, disconnection) <ul style="list-style-type: none"> <li>• Difficult to edit</li> </ul> </li> </ul>

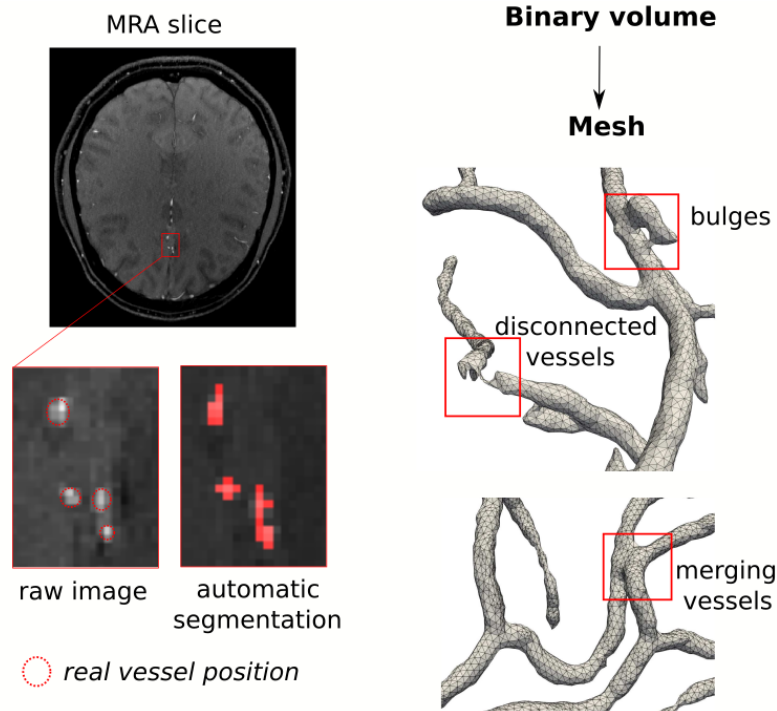


Figure 1.5: Challenges related to segmentation-based meshing: the left side compares the raw MRA slice to the segmentation results. The real position of vessels is enhanced to show the difficulty of the segmentation task when applied to small and close vessels. On the right side, the surface mesh produced from segmentation is shown and the defects observed in segmentation-based meshes are emphasized.

### 1.5.3 Centerline-based meshing

Due to their tubular nature, vessels can also be represented by their centerline only. A centerline is composed of a set of data points with 3 spatial coordinates  $(x, y, z)$  and a radius value  $r$ . The topology of the vascular network (e.g. bifurcations) is encoded by the connection between the data points, as shown in Figure 1.6. Segmentation-based and centerline-based representations give complementary information.

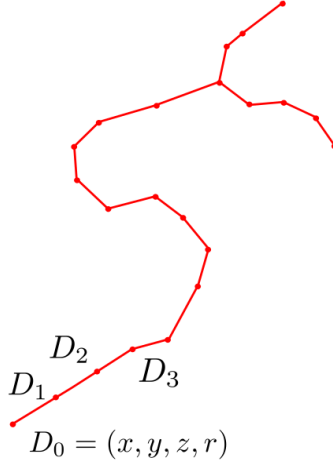


Figure 1.6: Centerline representation of the vascular network.

The advantage of this representation is that it requires only a few data points and their connectivity information to represent vascular networks, which makes it easy to manually edit or manually extract. For this reason, the only public datasets of whole vascular networks were built using centerlines. To our knowledge, there are no databases of whole brain arterial network segmentations. The centerline-based databases are presented in Section 1.6. Moreover, this representation type not only encodes the geometry of the network but also its topology. This makes it possible to automatically apply topological constraints to the vascular network to improve its accuracy (e.g. cycle removal, reconnecting vessels). The main limitation of centerlines is that the geometry information encoded is not sufficient to reconstruct a realistic 3D surface of the vascular wall. Especially, the bifurcations can have complex shapes, making it difficult to reconstruct from only a few data points.

Table 1.2: Advantages and disadvantages of centerline-based representation.

advantages	limitations
<ul style="list-style-type: none"> <li>• Requires few parameters</li> <li>• Easy to edit / post-process</li> <li>• Existing databases</li> </ul>	<ul style="list-style-type: none"> <li>• Sparse data (especially at branching) <ul style="list-style-type: none"> <li>• Radius and spatial uncertainty</li> <li>• Tubularity assumption</li> </ul> </li> </ul>

Centerline extraction can be performed either directly from the medical image or from a segmentation. From segmentation, centerlines can easily be produced by skeletonization methods. However, these centerlines will naturally inherit the defects of the segmentation (e.g. merging or disconnected vessels). The extraction of centerlines from the raw image is a more complex task. It can be performed manually by medical doctors [52, 89], semi-automatically or automatically by classic image methods [8, 77] or more recently deep neural networks [33, 94]. The generation of a mesh from centerlines only was less investigated in the literature than segmentation-based meshing. The main approaches include parametric modeling [25] and implicit functions [1]. As this work focuses on centerline-based meshing, a more detailed state of the art can be found in Chapter 2.

#### 1.5.4 Mesh quality

For applications in CFD with the finite element method, the inside of the surface mesh must be discretized into cells. The cell shapes include tetrahedral, prismatic, and hexahedral. Hexahedral meshes can be further divided into two categories; the structured meshes, where the neighborhood relationships between the cells are defined in the mesh structure (e.g. regular grid), and the unstructured meshes. The different types of mesh/cell are illustrated in Figure 1.7.

In the case of blood vessels, structured and unstructured hexahedral meshing also allows for the creation of flow-oriented cells. Studies of the literature show that both the shape of the cells (tetrahedral, hexahedral) and the type of mesh (structured or unstructured) influence the cost and the stability of the numerical simulation.

Various metrics can be used to measure the quality of the mesh cells. These quality metrics measure how far the cell is from the ideal cell shape. We note  $(e_1, e_2, \dots, e_n)$  the list of the vectors corresponding to the different edges in one cell ( $n = 12$  for hexahedra and  $n = 6$  for tetrahedra).

- **Cell aspect ratio:** For hexahedral cells, it is the ratio between the maximum and

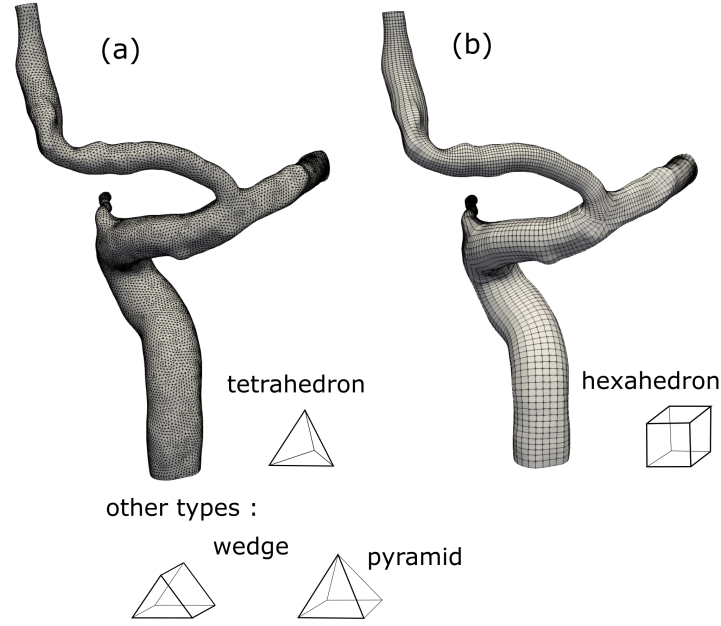


Figure 1.7: Different types of mesh: the same surface was meshed with (a) unstructured, randomly oriented tetrahedral cells and (b) structured, flow-oriented tetrahedral cells.

minimum edge length of the cell  $AspectRatio = \frac{\max_i(e_i)}{\min_i(e_i)}$ . For tetrahedral cells, it is the ratio between the maximum length edge and the radius  $r$  of the inscribed sphere  $\frac{\max_i(e_i)}{2\sqrt{6}r}$ . The ideal aspect ratio is 1.

- **Orthogonality:** It is the angle between the vector connecting the centers of two adjacent cells and the normal vector to the face shared between the two faces. The orthogonality values range between 0 (best quality) and 180 (worst quality). It is recommended to keep the orthogonality below 70 as meshes with cells of orthogonality  $> 85$  can cause numerical instability or diverge.
- **Skewness:** For hexahedral cells, it measures the divergence between the maximum or minimum angle between edges  $(\theta_{max}, \theta_{min})$ , and the ideal angle of 90 degrees  $\max(\frac{\theta_{max}-90}{90}, \frac{90-\theta_{min}}{90})$ . For tetrahedral cells, the cell volume is compared to the ideal volume  $V_{ideal} = \frac{8\sqrt{3}R^3}{2}$  where  $R$  is the radius of the cell external cell.  $Skewness = \frac{V_{ideal}-V}{V_{ideal}}$
- **Scaled Jacobian:** It also measures the deviation of a cell to the ideal cell shape by using the determinant of the Jacobian of the cells. This value is then normalized by the edge length to create the scaled Jacobian, which ranges between 0 (worst

element), and 1 (best element). The equations used to compute this metric are given below. This value is the most used to evaluate the quality of hexahedral meshes. Cells with a value lower than 0 are prohibitive for numerical simulation. The cells should have a value  $> 0.2$  to be considered of high quality. Figure 1.8 gives some examples of Scaled Jacobian values for cells of different shapes.

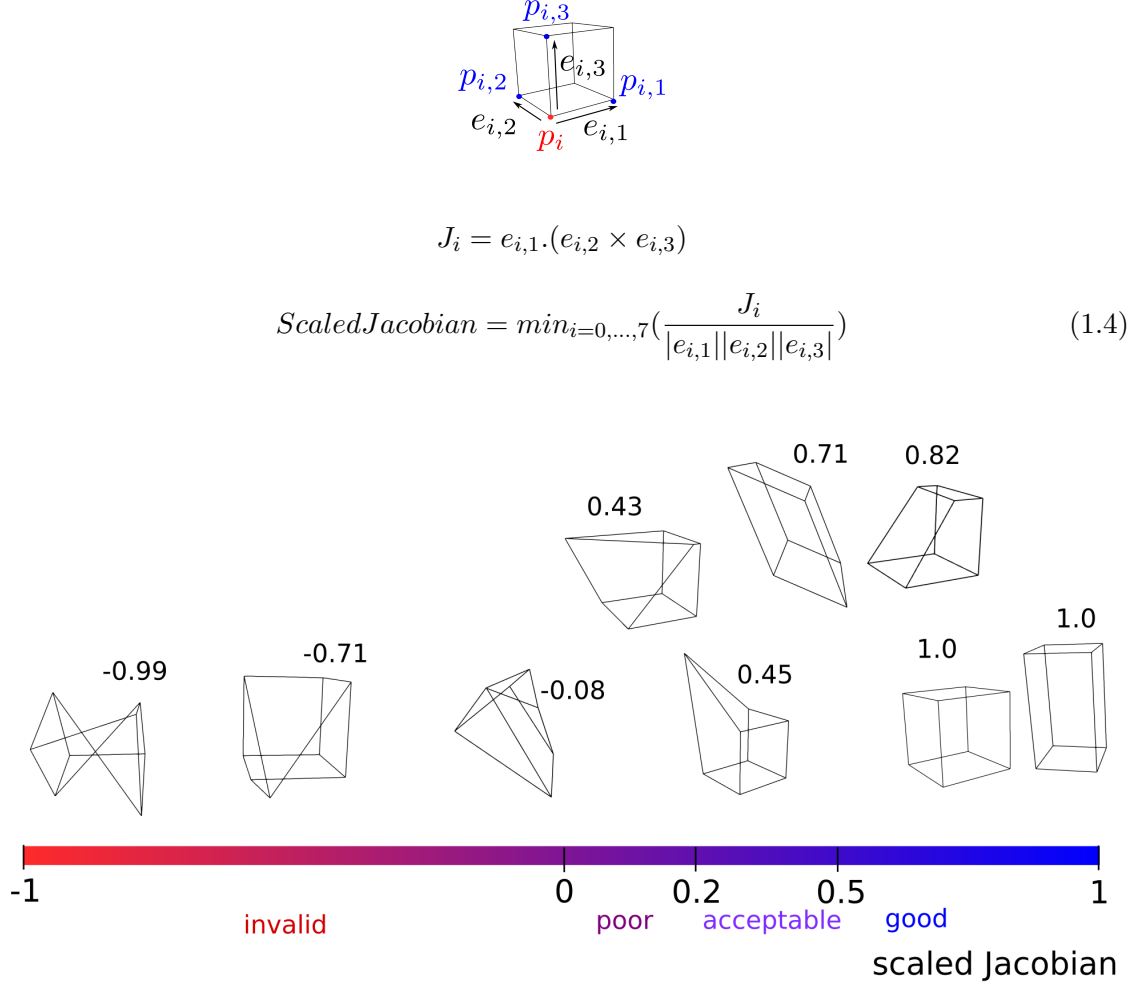


Figure 1.8: Scaled Jacobian values for different hexahedral cells.

## 1.6 Databases

In this work, we used two publicly available datasets. The BraVa database [89] gathers the centerlines of the whole cerebral network for 60 patients. To create this dataset, the data points were manually placed by medical doctors on medical images using the ImageJ

plugin Neurite Tracer [52] and the radius was automatically computed. The centerline extraction is illustrated for one patient of the BraVa database in Figure 1.9. As a result, the data points have a lower spatial resolution than centerlines extracted automatically from a surface mesh and are prone to errors and uncertainty of measurement. The TubeTK database [10] provides MRA brain images and the automatically extracted centerlines. Those centerlines were used as a basis to produce 34 in-house segmentations created by medical doctors.

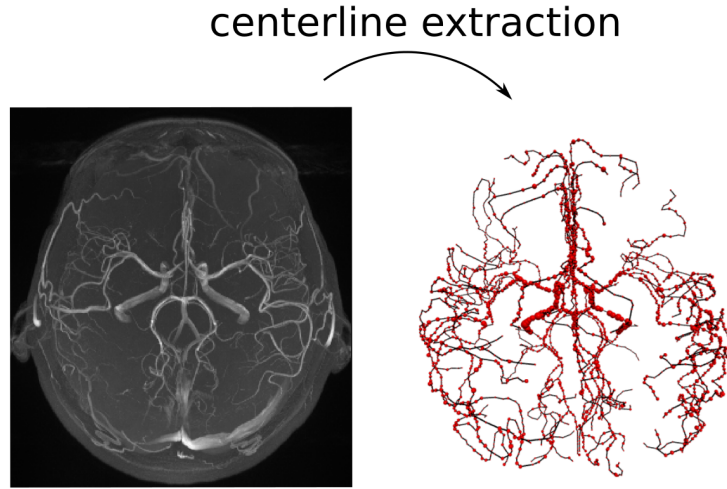


Figure 1.9: MRA images and centerline extracted for one patient of the BraVa database.

Finally, the Aneurisk database [3] provides 3D models of the main arteries of the circle of Willis for patients with an aneurysm, extracted by segmentation from MRA images. Meshes from this database were used to provide ground-truth vascular and pathology geometries.

## 1.7 Contributions of this thesis

In this context, we believe that bridging the gap between the centerline representation of vascular networks and CFD mesh would enable us to take advantage of the many assets of this representation type to run CFD studies in larger vascular networks. Therefore, the objective of this thesis is to develop a new method to generate a high-quality mesh ready for CFD from centerlines and radius only. The main challenges to overcome are

the limitation inherent to centerline-based representation (data sparsity and uncertainty), and the generation of a high-quality hexahedral mesh fulfilling the requirement for CFD. In this work, our purpose is to meet the challenges arising from the state-of-the-art with a two-step framework integrating a modeling and a meshing step of cerebral vascular networks. The shortcomings of segmentation-based meshing are addressed by developing a method based on centerlines. A parametric model is used to overcome the common defect of centerlines and reconstruct a realistic vascular surface. Especially, we introduced a vessel centerline approximation algorithm based on penalized splines which is robust to noise and low-sampling of the data points, and we used a parametric model of bifurcation based on anatomical parameters [93] to reconstruct realistic bifurcation shapes. An original meshing algorithm is proposed to create high-quality hexahedral meshes suitable for CFD simulations, as opposed to the largely used tetrahedral meshes. The proposed method is compared against segmentation-based methods concerning the accuracy of the reconstructed surface and CFD-related performance of the volume mesh. Finally, we developed a user interface that facilitates the use of the proposed framework and allow for manual editing of vascular networks.

In Chapter 1, the background of this work was presented and the notions referred to in this manuscript were explained. In Chapter 2, we present the parametric model that we developed to represent the vessels and bifurcations from centerlines. In Chapter 3, we present the meshing methods that we proposed in order to produce a high-quality volume mesh with flow-oriented, hexahedral cells from the model described in Chapter 2. In Chapter 4, we evaluate the proposed method against other centerline-based and segmentation-based meshing methods, in terms of anatomical realism of the reconstructed surface. We also compare the performance of hexahedral meshes over tetrahedral meshes concerning CFD simulation cost and accuracy. Finally, in Chapter 5, we present the software that we developed to facilitate the visualization, meshing, and editing of vascular networks. Chapter 6 presents the general conclusion of this work.

---

# Modeling

## 2.1 Introduction

In this chapter, we present the parametric model that we developed to reconstruct a continuous vessel surface from the input centerlines data. Using a parametric model allows the application of some constraints on the shape of the vessels or the branching parts (e.g. tubularity), improving the anatomical realism of the reconstructed surface. It also provides a light representation of vascular networks, whose shape is encoded with few parameters.

As our goal is to mesh vascular networks from centerlines only, this modeling step is crucial to overcome the lack of geometrical information on this representation type. The main challenges that we need to overcome are (1) the sparsity of the data points, (2) the uncertainty in the data points coordinates and radius (3) the misplaced branching points. Those characteristics are common to both manually extracted centerlines and automatically extracted centerlines. In the first case, the centerline extraction is a very time-consuming task: it is preferable to use only a few data points. The low resolution of medical images and visualization problems related to the fact that they are 3-dimensional cause noise and misplaced points. In the case of automatically extracted centerlines, as they rely on the medical image, the low resolution or the imaging artifacts also impact the accuracy of the data points' coordinates and radius and the sampling of the centerlines. In this work, we will refer to the uncertainty on the centerline data points' position and radius as **noise**. Generally, automatic extraction methods rely on the segmentation of the medical images, followed by a skeletonization step. As the position of branching points



varies depending on the skeletonization algorithms, the misplacement of the bifurcation points is often observed.

While the characteristics of centerlines, such as the low number of data points, can hinder the reconstruction task, they also offer many advantages. For instance, it makes the editing and visualization of the centerlines easier, a characteristic that we would like to preserve. For this reason, we have committed ourselves not to alter the original data points by some post-treatment (e.g. smoothing, resampling). We propose instead to overcome these limits by fitting a parametric model to the original centerlines.

In our framework, the vessels and bifurcation are treated separately, as they present different characteristics. The vessels are modeled as tubes whose surface is defined by spline functions. We introduced an original approximation method to optimize the model parameters (i.e. the control points of the splines), which enable us to model both the spatial coordinates and the radius in a single function and offer good robustness to noise and low-sampling. The vessel model is combined with the model of bifurcation proposed by [93] to form a parametric model of the entire vascular network. A method to extract the parameters of the bifurcation model directly from centerlines is proposed, and the model was generalized to planar  $n$ -furcations (i.e.  $n$ -furcations where the branches lie in the same plane). If the bifurcation model itself is not new, the use of this type of anatomical bifurcation model - as opposed to geometrical bifurcation models - to reconstruct a realistic vascular shape has not been investigated in previous studies. The bifurcations and vessels are then assembled to form the vascular network. The proposed model is evaluated against other modeling methods.

## 2.2 State-of-the-art

In this section, we review the methods used to recreate an accurate vascular surface model from centerline information. The main issues to overcome in this task arise from the defects commonly observed in the vascular centerline extracted from medical images; discontinuities causing a lack of information - especially at the bifurcation parts - and noise due to the voxelization. In this context, the smoothness of the vessel surface and the accurate reconstruction of the bifurcation part are important locks. The reconstruction methods can be divided into explicit methods, where parametric models are used for the

reconstruction of the surface, and implicit methods where the surface is represented by implicit functions. Implicit methods employ radial basis functions [36], implicit extrusion surfaces [37] or local implicit modeling [41] to reconstruct vascular networks from medical images or centerlines [1]. If they stand out by their ability to reconstruct complex branching topology, they can be sensitive to the sampling and incertitude of the centerline data points, causing the reconstructed surface to look bumpy. Besides, they do not allow as much control on the final mesh as explicit methods (e.g. hexahedral meshing) which makes them less suited for CFD using the finite element method.

In explicit methods, the vessels are considered to be tubular and modeled by approximation functions. Different models of centerlines were proposed in the literature, based on the approximation of data points by Bezier segments for Ghaffari et al. [25], regression splines for Kocinski et al. [43], free knot regression splines or local polynomial smoothing for Sangalli et al. [71]. However, only Sangalli et al. [71] give the detail of the implementation of the approximation method and provide a thorough study of the accuracy of their model regarding the spatial coordinates and the derivatives. The accuracy of both the first and second derivatives is crucial because the vessel curvature impacts the hemodynamics [70]. Moreover, meshing techniques are often based on the normals of the centerline [43, 25]. Finally, the proposed approximation methods [71, 43, 25] focus on the spatial coordinates of the centerlines, excluding the radius.

For the branching part, various bifurcation models were proposed. In the work of Kocinski et al. [43] and Ghaffari et al. [24], the three branches of the bifurcation are modeled separately and joined at the bifurcation center. The junction is then blended to restore the continuity, by a subdivision scheme for [43] and Bezier segments for [24]. This modeling approach, which we call *geometrical model*, facilitates the creation of a hexahedral mesh. However, the realism of the bifurcation shape depends on the accuracy of the position of the bifurcation center and the tangent of the branches, which is hard to estimate correctly from centerlines. Han et al. [31] and Guo et al. [28] modeled bifurcations using three tubes connecting the inlet and outlet sections. Half of each section of the tube is meshed by sweeping and the hole left in the middle is filled afterward. Their method guarantees the smoothness of the model but results in unnatural-looking bifurcations. Finally, Zakaria et al. [93] proposed an model where the bifurcations are represented by two merged tubes. It was validated concerning both the accuracy of the

bifurcation shape compared to the real patient anatomy and the CFD simulations. It does not rely on the geometrical center of the bifurcation but on a set of anatomically meaningful parameters (apex, apical sections, inlet, and outlet sections). In this work, we refer to this model as *anatomical model*. However, the authors extracted the model parameters from a surface mesh, and they did not suggest a way to extract them from centerlines. In this work, we propose to address the shortcoming of the current modeling methods with a centerline approximation strategy which enables to control the trade-off between smoothness and proximity to the input data points, and an anatomical bifurcation model offering more realism than the geometrical bifurcation model commonly used.

## 2.3 Vessels

In this part, we focus on the modeling of the vessels; the case of bifurcations is addressed in the next section. The vessel models rely on the approximation of the centerline data points by splines.

### 2.3.1 Splines

In the proposed model, the vessel surface is reconstructed from centerline data points using spline functions. Spline functions are parametric curves that were proposed by [15]. They have been widely used in domains such as 3D modeling and computed-assisted design for their flexibility and simplicity of construction. The trajectory of the curve is dictated by several control points called  $P$ . The local influence of each control point on the curve trajectory is defined by both a knot vector and a set of functions called basis functions. The knot vector is a list of times  $t$  between 0 and 1 which delimitates the zone of influence of the control points. The basis functions  $N$  define locally the intensity of the influence of the control points. A spline function  $s$  can be written

$$s(u) = \sum_{i=0}^{n-1} N_{i,p}(u) P_i, \quad (2.1)$$

for  $u \in [0, 1]$ , where  $N_{i,p}$  is the  $i$ th basis spline function of order  $p$  and  $\{P_0, P_1, \dots, P_{n-1}\}$  the  $n$  control points of the spline. The shape of the basis splines functions and therefore the part of the spline controlled by a given control point is given by a set of knots.

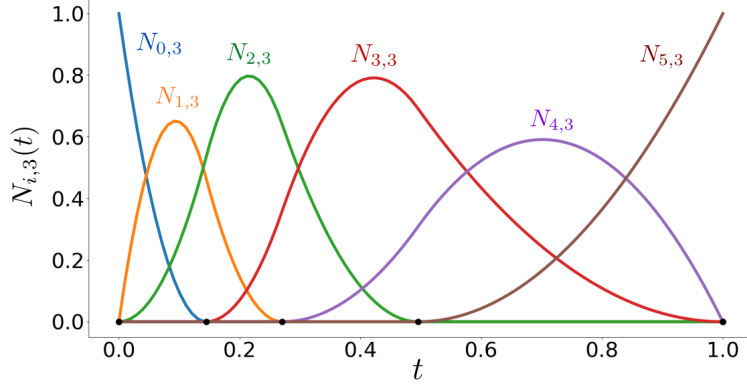


Figure 2.1: Example of spline basis functions.

### 2.3.2 Control points optimization

Here, we aim to approximate a set of  $m$  points  $\{D_0, D_1, \dots, D_{m-1}\}$  with 4 coordinates  $(x, y, z, r)$ . The main challenge in the approximation of noisy data is to find the optimal balance between the proximity of the curve to data points and the smoothness of the curve (i.e. the accuracy of the derivatives). There are two main approaches to control the smoothness of a spline function. The first is to change the number of control points: a low number of control points will result in a smoother curve. In this case, the position of the knots can be optimized by the free-knot regression approach, like in [71]. In the other approach, a relatively large number of control points and a uniform knot vector are used and the smoothness is constrained by a penalty on the second derivatives [14, 21]. For reasons further detailed in the next paragraphs, the second approach was judged more suitable for our task. The vessels are modeled with penalized splines, as introduced by [21].

The optimization of the control points of penalized splines is based on a cost function with two terms. The first term takes into account the closeness to the data point and the second term takes into account the smoothness of the approximation spline. The parameter  $\lambda$  controls the balance between both closeness and smoothness, as illustrated in Figure 2.2. The cost function is defined as

$$f(P_0, \dots, P_{n-1}) = \sum_{k=0}^m |D_k - s(t_k)|^2 + \lambda \sum_{j=2}^n (P_j - 2P_{j-1} + P_{j-2})^2, \quad (2.2)$$

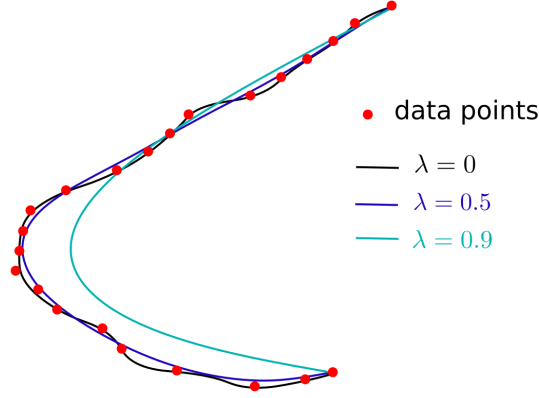


Figure 2.2: Approximation splines with different values of smoothing parameter  $\lambda$ .

where  $t$  is a time parametrization vector that associates each data point to a position on the spline. The centerline approximation strategy is further detailed in the next paragraph.

### 2.3.3 Smoothing parameter optimization

The smoothing parameter  $\lambda$  is automatically selected by minimizing a criterion that measures the balance between the complexity of the spline model and the fidelity to the data points to approximate. Several criteria were proposed in the literature for the optimization of the  $\lambda$  parameter; the Akaike criterion  $AIC$ , the corrected Akaike criterion  $AICC$ , the Bayesian information criterion  $SBC$ , the cross-validation  $CV$  and the generalized cross-validation  $GCV$ . The equations of the different criteria studied are given hereafter:

$$AIC = m \log(SSE/m) + 2tr, \quad (2.3)$$

$$AICC = 1 + \log(SSE/m) + \frac{2(tr + 1)}{m - tr - 2}, \quad (2.4)$$

$$SBC = m \log(SSE/m) + tr \log(m), \quad (2.5)$$

$$CV = \sum_{k=1}^m \left( \frac{D_k - \hat{s}(t_k)}{1 - h_{i,i}} \right)^2, \quad (2.6)$$

$$GCV = \sum_{k=1}^m \left( \frac{D_k - \hat{s}(t_k)}{m - tr} \right)^2, \quad (2.7)$$

where  $tr$  is the trace of the matrix  $H = N(N^t N + \lambda \Delta)^{-1} N^t$ ,  $h_{i,i}$  the  $i_{th}$  diagonal element of  $H$ ,  $D_{1,...,m}$  are the data points,  $\hat{s}$  is the approximation spline, of parametrization  $t$ , degree  $p$  and number of control points  $n$  and  $SSE = \sum_{k=1}^m (D_k - \hat{s}(t_k))^2$  is the squared sum of errors.

The comparison of the performances of these different criteria, whose results are given in Section 2.5.5, led to the choice of  $AIC$  in this work. To reduce the computational time, the minimal value of  $AIC$  is computed by a golden-section search algorithm. We modified the original structure of the golden-section search algorithm to offer the possibility to include a maximal distance from the data points as a constraint in the approximation, as described in Algorithm 1.

---

```

a ← 10e − 6
b ← 10e6
tol ← 10e − 5
gr ←  $\frac{1+\sqrt{5}}{2}$ 

c ← b − (b − a)/gr
d ← a + (b − a)/gr
while abs(b − a) > tol do
    if f(c) < f(d) or d(c) < max_dist then
        b ← d
    else
        a ← c
        c ← b − (b − a)/gr
        d ← a + (b − a)/gr
    opt_lambda = (b + a)/2

```

---

Algorithm 1: Modified golden-section search algorithm.  $gr$  is the golden ratio value,  $f$  is the function to minimize (here the Akaike criterion) and  $d$  is the function returning the maximum distance of the spline to the data points.

### 2.3.4 Two-step approximation strategy

Each centerline data point encodes both a spatial position  $(x, y, z)$  of the vessel center and its radius  $r$ . To reconstruct the vessel surface from centerlines, the information contained in those two variables must be reflected by the model. However, the spatial

position and the radius are variables that cover a different range of values. Furthermore, depending on the extraction algorithm used, they might have different incertitude levels. For this reason, we believe that they should be approximated separately. The choice of penalized splines allows us to dissociate  $\lambda$  values for the position and the radius in a two-step approximation algorithm. With this approach, the spatial and radius coordinates can be modeled using a single spline function.

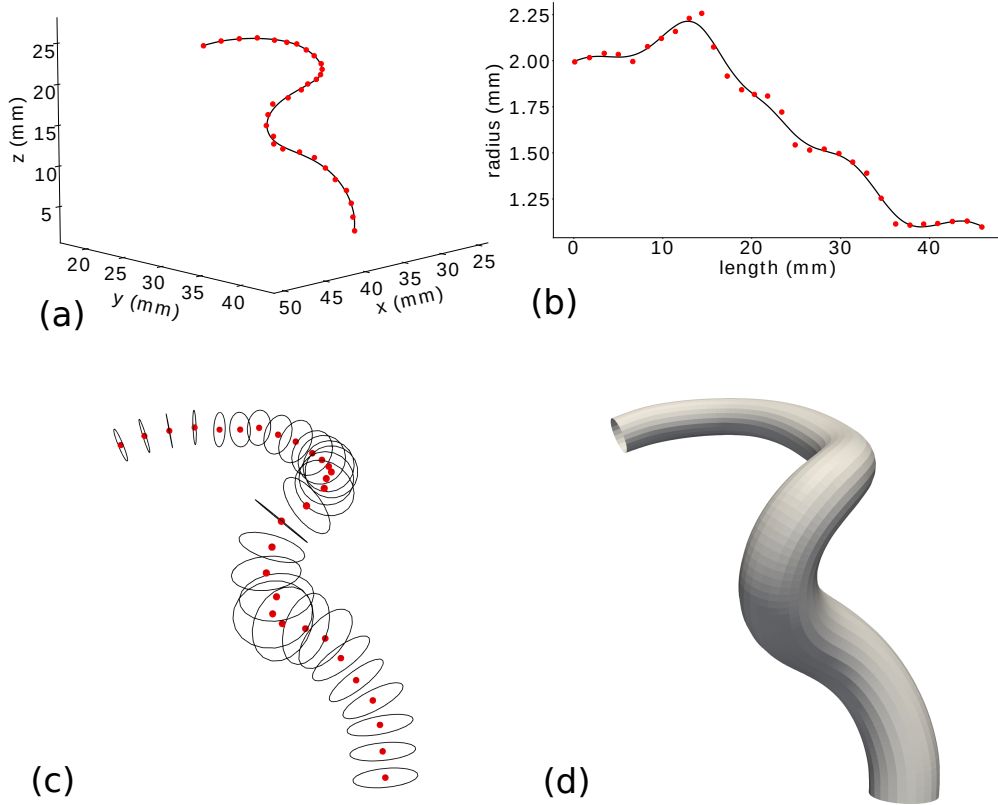


Figure 2.3: Approximation of a noisy centerline with the proposed method. (a) and (b) show respectively the approximation of the spatial coordinates and the radius. Figure (c) shows the input centerline data in 3-dimension as red dots with a radius represented by black circles. In (d), the surface defined by the approximating spline-based model is represented.

For the approximation, we use a uniform knot vector and a parametrization obtained by the chord-length method. The number of control points is set so that the non-penalized approximation curve (i.e. produced by solving equation 2.2 with  $\lambda = 0$ ) has a root mean square distance from the original data lower than a given value, which was set in this work to  $10^{-1}$  for spatial coordinates and  $10^{-3}$  for the radius. This guarantees that the number

of control points is sufficient to capture correctly the complexity of the trajectory of the centerline.

We first solve the linear system arising from equation 2.2 for the spatial coordinates  $(x, y, z)$  of the centerline data points. This system can be written as

$$P_{(x,y,z)} = (N^T N + \lambda_s \Delta)^{-1} N^T D_{(x,y,z)}, \quad (2.8)$$

where  $N$  is the matrix of representation of the basis spline functions and  $\Delta$  is the matrix representation of the difference operator which appears in the second term of the cost function 2.2. The system is solved by matrix inversion. The optimal value for  $\lambda_s$  is obtained by minimizing the Akaike criterion  $AIC_2$ , as written in Equation 2.3.

Then, the linear system is solved for the data  $(t, r)$  where  $t$  is the time parametrization of each data point and  $r$  their radius value:

$$P_{(t,r)} = (N^T N + \lambda_r \Delta)^{-1} N^T D_{(t,r)}. \quad (2.9)$$

The value of  $\lambda_r$  is also selected by minimizing AIC on the time/radius data. The spatial coordinates and radius of the optimized control points are then concatenated to form the 4-coordinates control points of the final spline. Figure 2.3 illustrates this two-part approximation scheme. The proposed approximation method is compared with other conventional approximation methods regarding the robustness to noise and low sampling of the data points in section 2.5.

## 2.4 Bifurcations

### 2.4.1 Model description

Zakaria et al. [93] proposed a parametric model for non-planar bifurcations. Their model was validated regarding both the anatomy and numerical simulation of blood flow and showed a good agreement with real cerebral bifurcations. It requires only a few parameters and is well-suited for the reconstruction of bifurcations from sparse data. In this model, bifurcations are created by merging two tubes that represent the daughter vessels. The tubes are defined by a shared inlet cross-section  $C_0$ , separate apical cross-sections  $AC_1$ ,  $AC_2$  and outlet sections  $C_1$  and  $C_2$ . The apical cross-sections  $AC_{1,2}$  are



located at the apex point  $AP$  of the bifurcation, where both tubes merge. The outlet sections  $C_{1,2}$  are cut one diameter away from the apex. In total, five cross-sections and their normals are required to build the model. A cross-section  $C$  is considered circular and is represented by the three spatial coordinates of its center  $P_c$ , a radius  $r_c$ , and a normal  $\vec{n}_c$ . The centerline of each tube is defined by a centerline  $spl_{1,2}$ . The first segment of the centerline connects the inlet section  $C_0$  to the apical section, and the second connects the apical section to the outlet sections. The tangent of the centerline segments matches the normal of the joined cross-sections. The radius along the segments evolves linearly between  $r_{C_0}$ ,  $r_{AC_{1,2}}$  and  $r_{AC_{1,2}}$ ,  $r_{C_{1,2}}$ . The bifurcation model is illustrated in Figure 2.4. The unrealistic sharp angle produced between tubes at the apex is rounded by a segment of constant radius of curvature  $R$ .

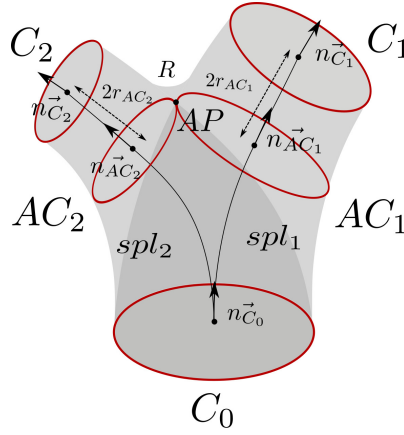


Figure 2.4: Five cross-sections bifurcation model introduced by [93].

### 2.4.2 Parameter estimation from centerlines

We introduce an algorithm to estimate the parameters of the bifurcations directly from the input centerline data. The inlet data points (in light blue in Figure 2.5 (a)) are merged with each of the outlet data points (resp. in deep blue and green in Figure 2.5 (a)) to form two input centerlines going through the bifurcation, as shown in Figure 2.5 (b). The two vessels based on these centerlines are modeled independently by splines using the approximation strategy presented in section 2.3. The apex  $AP$  of the bifurcation is set as the point where the surface of the two vessel models first intersect (red dot on the Figure 2.5 (c)).  $AP$  is then projected on the model splines  $spl_1$  and  $spl_2$ . The tangent and position of the obtained projection points then define the normal and the center of

the apical cross-sections  $AC_1$  and  $AC_2$ . The outlet sections  $C_1$  (resp.  $C_2$ ) are computed as the point of the spline where the length from the apex projection point is twice the radius of the apical section, as suggested by [93].

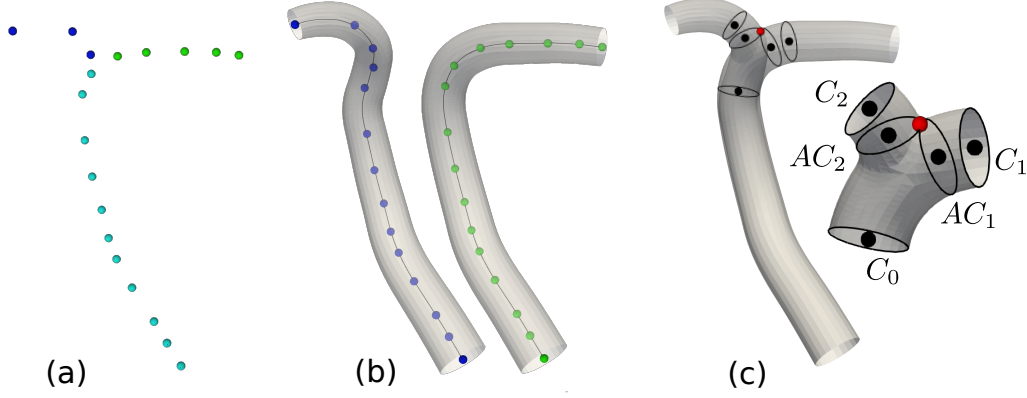


Figure 2.5: Pipeline of the bifurcation parameter estimation. (a) shows the inlet and outlet data points, (b) the independent vessel models, and (c) the parameter extraction and resulting bifurcation.

### 2.4.3 Extension to $n$ -furcation

If the cerebral arterial network is composed of a majority of bifurcations, multifurcations may also be present (e.g. trifurcations are frequently found on the basilar artery). To address this requirement, we generalized the model of [93] to  $n$ -furcations. However, the meshing strategy proposed in the next chapter could not be generalized to  $n$ -furcation with branches lying in different planes, referred to as non-planar  $n$ -furcation. Therefore, we focus here on the case of planar  $n$ -furcations. The generalized  $n$ -furcation model is built with  $n - 1$  splines,  $2n + 1$  cross-sections and  $n - 1$  apex points, as illustrated for the case  $n = 3$  in Figure 2.6 (a). We adapted the decomposition scheme presented in section 2.4 to compute  $n + 1$  separation plans, as in Figure 2.6 (b). Figure 2.6 (c) shows an example of a planar trifurcation mesh obtained with this generalization.



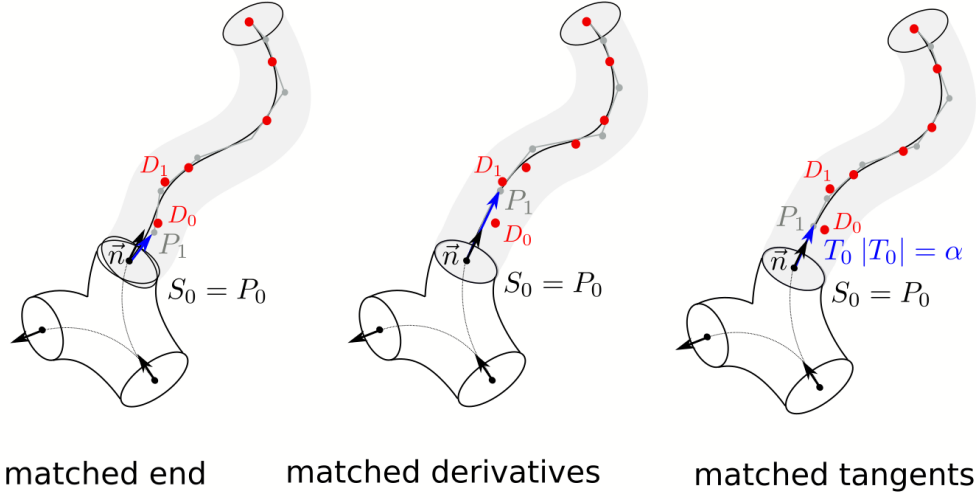


Figure 2.7: Illustration of the end constraints applied in the vessel approximation to maintain the continuity of the model.

For this reason, we propose to propose a weaker constraint that fixes the end tangent while the derivative is free. In this approach, the start and end tangent magnitudes  $\alpha$  and  $\beta$  are included in the parameters to optimize. We consider a spline  $s$  as defined by Equation 2.1, and because we work with clamped curves,  $s(0) = P_0$  and  $s(1) = P_{n-1}$ . Moreover,  $s'(0)$  (respectively  $s'(1)$ ) is in the same direction as vector  $P_1 - P_0$  (respectively  $P_{n-2} - P_{n-1}$ ). If we note  $S_0$  and  $S_{n-1}$  the fixed end-points and  $T_0$  and  $T_{n-1}$  the fixed end tangents, the following new conditions are used during the minimization of Equation 2.2:

$$\left\{ \begin{array}{l} P_0 = S_0 \\ P_{n-1} = S_{n-1} \\ P_1 = P_0 + \alpha T_0 \\ P_{n-1} = P_{n-2} + \beta T_{n-1}, \end{array} \right. \quad (2.10)$$

The details of the system resolution are given in Appendix.

## 2.5 Validation

### 2.5.1 Validation pipeline

The accuracy of the final model largely depends on the spline approximation method as the bifurcation parameters are also extracted using this approximation. To evaluate the proposed approximation method, and compare it to other state-of-the-art methods, we designed the following experiment:

1. High-quality, low-noise centerlines are extracted from a ground truth mesh.
2. The extracted centerline is distorted by adding radius and spatial noise and low sampling of the data.
3. The distorted centerlines are approximated by different approximation methods.
4. The approximation is compared to the original centerline with different criteria (length, curvature...).

This pipeline is illustrated in Figure 2.8.

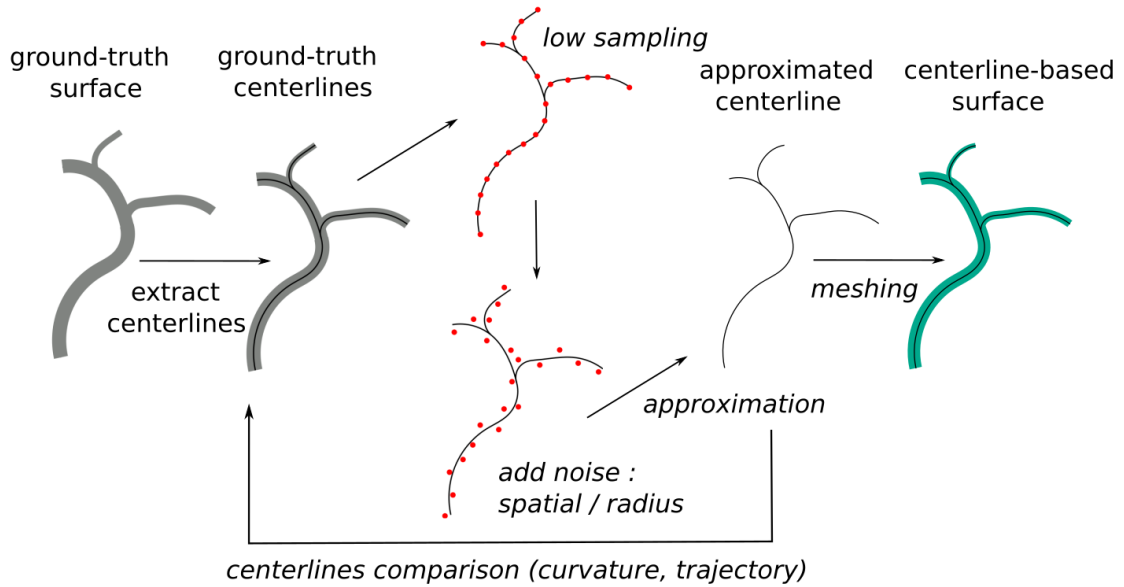


Figure 2.8: Approximation strategy evaluation pipeline.

## 2.5.2 Validation dataset

### 2.5.2.1 Ground-truth centerlines

For this evaluation, we built a dataset of ground truth vessel models. Four surface meshes of cerebral arteries from the Aneurisk database were selected. For each mesh, a single vessel starting from the inlet of the network and ending at an outlet was selected so that it does not include pathologies but goes through bifurcations, where we generally observe high curvature and big radius change. The selected vessel centerlines were extracted with a good resolution and low noise using the VMTK software. The obtained high-quality centerline data points were then approximated by a 4-coordinate spline  $s$  that constitutes the ground truth. The control points were manually added and the accuracy of the fitting of the spatial coordinates, radius and first derivatives was checked visually until the approximation was judged satisfying. The figure 2.9 shows the ground truth centerlines.

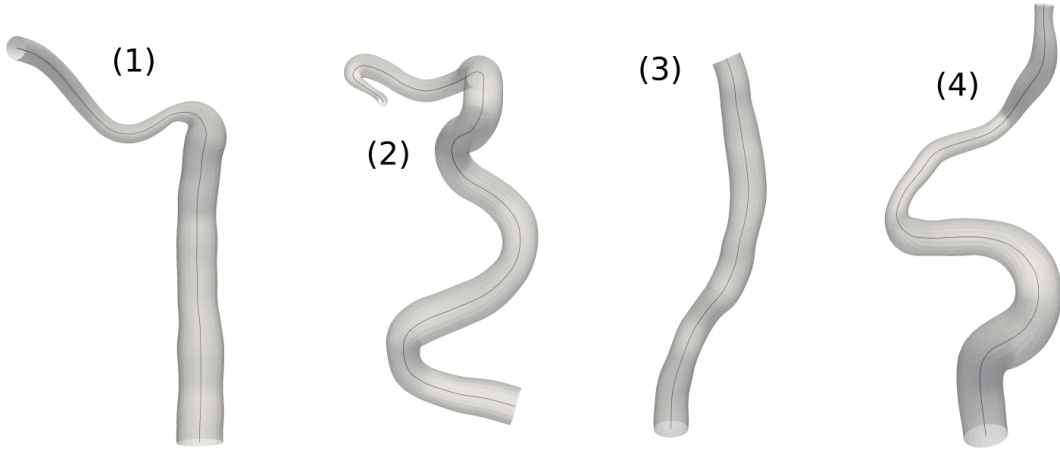


Figure 2.9: Four ground-truth centerlines created for this study.

### 2.5.2.2 Deterioration operations

To evaluate the robustness of our approach, those ground truth data were distorted to mimic defects commonly observed in realistic centerline data; low sampling and noise. Spatial noise and radius noise were applied separately, as they might differ in level. To generate spatial noise, the data points were displaced from their original position. The magnitude of displacement is randomly picked from a zero-centered Gaussian distribution

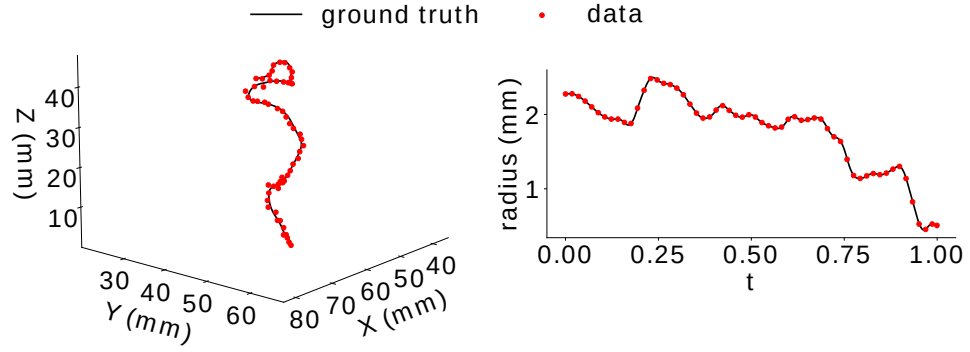
with standard deviation  $\sigma_{spatial}$ . In order not to affect the radius values, the direction of the displacement is normal to the ground truth spline  $s$ . Random radius noise is generated randomly from a zero-centered Gaussian distribution of standard deviation  $\sigma_{radius}$  and added to the ground truth radius. In both cases, the applied standard deviation value is proportional to the point radius, as indicated in Table 2.1, to keep similar levels of noise between big and small vessels. Finally, low sampling is obtained by removing data points along the centerline to reach target point densities. Figure 2.10 shows the different types of deterioration applied to the data.

Table 2.1: Parameters used for the distortion of the ground truth centerlines

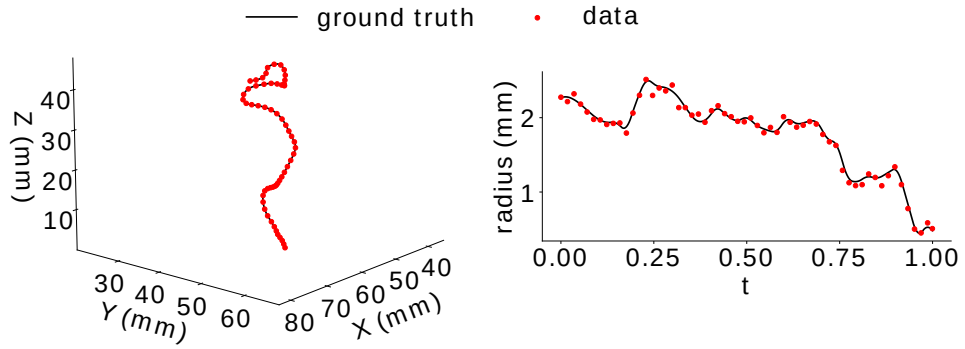
density ( $\text{mm}^{-1}$ )	2	4	10	16	20
$\sigma_{radius}(\text{mm})$	$0.01r$	$0.05r$	$0.1r$	$0.3r$	$0.5r$
$\sigma_{spatial}(\text{mm})$	$0.01r$	$0.05r$	$0.1r$	$0.3r$	$0.5r$

For each density value in Table 2.1, ten combinations of noise parameters are used, spatial and radius noise being added separately. Radius noise is applied to the ground truth data with parameters  $\sigma_{radius}$  as given in Table 2.1 while the spatial noise is set to 0. Then spatial noise is applied with parameters  $\sigma_{spatial}$  as given in Table 2.1 while the radius noise is null. Each noise combination is repeated three times to account for the stochastic effect; we get 30 data per density value, thus 150 in total. This is done for the four vessels of the ground truth dataset, bringing the number of data in the distorted dataset to 600.

(a) Spatial noise addition



(b) Radius noise addition



(c) Resampling

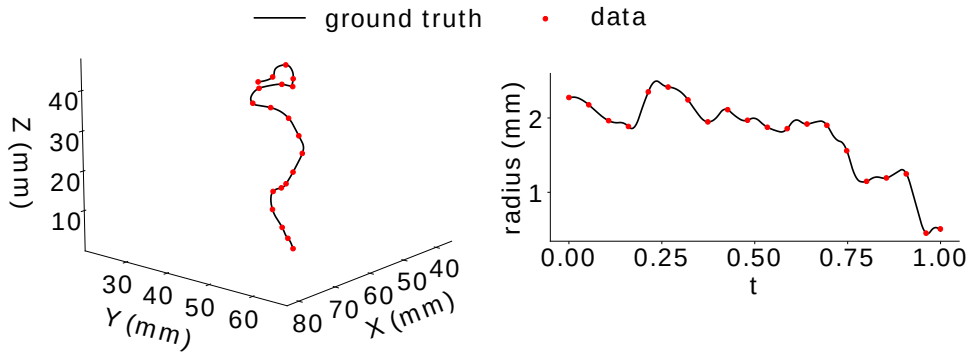


Figure 2.10: Deterioration operations applied to the centerlines. The ground-truth centerline trajectory is represented by a black line, and the data points of the centerline after deterioration are displayed in red. Both the spatial (left) and the radius (right) dimensions are shown.



### 2.5.3 Approximation methods

To demonstrate the robustness and the accuracy of the approximation strategy used to reconstruct the surface of the vessels presented in Section 2.3, we compared it to other explicit centerline-based meshing methods of the literature [43, 25]. Those methods also rely on splines or Bezier curves to approximate the centerline data points and reconstruct the vessel surface. As most of the authors did not provide the details of the fitting method employed, we implemented four commonly used spline-based approximation methods with incremental complexity to emphasize the contributions of the proposed method.

- **Global Non-Penalized (GNP)**: In this basic approach, the control points are optimized without smoothness penalty in the cost function (Equation 2.2 with  $\lambda = 0$ ). The number of control points is set to match the RMSE threshold given in Section 2.3. We call it global because the spatial and radius dimensions are not addressed separately.
- **Global Non-Penalized with Akaike criterion (GNP-AIC)**: Optimizing the number of control points to obtain the desired spline smoothness is a common approximation method in the literature. In this approach, the optimal number of control points minimizes the Akaike information criterion  $AIC_2$ :

$$AIC_2 = m \times \log(SSE) + 8(n + p). \quad (2.11)$$

where  $m$  is the number of data points,  $p$  is the degree of the spline,  $n$  is the number of control points and SSE is the sum squared error from the data points, including their four coordinates. This criterion is based on the same principle as the  $AIC$  criterion of equation 2.3 but is adapted to non-penalized splines.

- **Global Penalized with Akaike criterion (GP-AIC)**: This approach corresponds to the original approximation by penalized splines described in [21]. It uses the same global approach as in GNP, but with a smoothing penalty defined with a parameter  $\lambda \neq 0$  as in Equation 2.2.
- **Spatial coordinates and Radius Penalized with Akaike criterion (SRP-AIC)**: The approximation strategy that we propose in this work penalizes spatial and radius

dimensions separately. The comparison of our strategy with GP-AIC allows us to evaluate the contribution of treating the spatial and radius coordinates individually.

In methods GP-AIC and SRP-AIC, the criterion used to optimize the  $\lambda$  values is another formulation of the Akaike information criterion  $AIC$  of Equation 2.3, adapted to penalized splines, as proposed by [21].

Unlike  $AIC_2$ , it is not employed to choose an optimal number of control points but to select the optimal value for the smoothing parameter  $\lambda$ .

#### 2.5.4 Quality metrics

A total of six quality metrics were selected to evaluate the approximation strategies presented in the previous paragraph. To build an accurate measure of distance between the ground truth spline  $s$  and the approximation spline  $\hat{s}$ , we project one curve on the other. As illustrated in Figure 2.11, two matched sets of time parameters are built. The spline  $s$  is equally sampled with a time vector  $t$ , then projected on  $\hat{s}$  according to the minimum distance to form the matched time vector  $T$ .

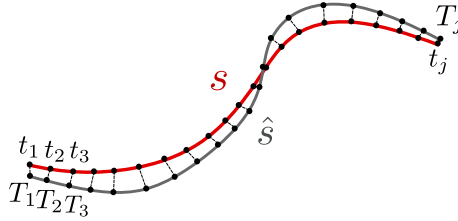


Figure 2.11: Matching time parameters by minimum distance projection from  $s$  onto  $\hat{s}$

Once the projection is performed, the matched values can be compared. We use the root mean squared error (RMSE) as a measure of the closeness of the approximation spline to the ground truth spline. The spatial coordinates and the radius values are treated separately in the evaluation. We note  $RMSE_{radius}$  (respectively  $RMSE_{spatial}$ ) the root mean squared error of the radius (respectively the spatial coordinates). The couple  $(t, T)$  depends on the spline from which  $t$  is sampled. To have a robust comparison between the curves, the projection is computed in both ways (from  $s$  to  $\hat{s}$  and from  $\hat{s}$  to  $s$ ) and the final RMSE value is the average of the RMSE yielded by both projections.

The accuracy of the first derivatives of the model is evaluated by the metrics  $RMSE_{der_{spatial}}$  and  $RMSE_{der_{radius}}$ . As curvature is commonly considered in hemodynamic studies, the

model performance concerning the centerline curvature is also measured, by the metric  $RMSE_{curv}$ . Finally, the length of the vessel affects the delay of blood arrival between the inlet and the outlet of the vascular tree in numerical simulations. Therefore, the difference  $L_{diff}$  of length between the ground truth and the approximated centerline was considered.

The formulas of the quality metrics used for the evaluation of the approximation methods are given hereafter. The ground-truth spline function (respectively its first and second derivatives) is noted  $s$  (respectively  $s'$  and  $s''$ ), and the spline function yielded by the strategy that we want to evaluate (respectively its first and second derivatives) is noted  $\hat{s}$  (respectively  $\hat{s}'$  and  $\hat{s}''$ ).  $s_{xyz}(t)$  (resp.  $\hat{s}_{xyz}(t)$ ) are the 3D coordinates of the point resulting from the evaluation of  $s$  (resp.  $\hat{s}$ ) at a time  $t$ , and  $s_r$  (resp.  $\hat{s}_r$ ) is the radius. We note  $p_{f,g}(t)$  the minimum distance projection of the time vector  $t$  of  $f$  onto  $g$ .  $L_s$  and  $L_{\hat{s}}$  are the length of curves  $s$  and  $\hat{s}$ .  $curv(s)$  and  $curv(\hat{s})$  the curvature of curves  $s$  and  $\hat{s}$  such that

$$curv(s) = \frac{|s' \times s''|}{|s'|^3}.$$

$RMSE_{xyz}(f, g)$  (resp.  $r(f, g)$ ) is the root mean squared error between the functions  $f$  and  $g$ , with regard to the spatial (resp. radius) coordinates:

$$RMSE_{xyz}(f, g) = \sqrt{\frac{\sum_{k=0}^j |f_{xyz}(t_k) - g_{xyz}(p_{f,g}(t_k))|^2}{m}},$$

$$RMSE_r(f, g) = \sqrt{\frac{\sum_{k=0}^j |f_r(t_k) - g_r(p_{f,g}(t_k))|^2}{m}}.$$

From this, the equations of the quality metrics  $RMSE_{spatial}$ ,  $RMSE_{radius}$ ,  $RMSE_{der_{spatial}}$ ,  $RMSE_{der_{radius}}$ ,  $RMSE_{curv}$  and  $L_{diff}$  can be written as:

$$\begin{aligned}
RMSE_{spatial} &= \frac{RMSE_{xyz}(s, \hat{s}) + RMSE_{xyz}(\hat{s}, s)}{2}, \\
RMSE_{radius} &= \frac{RMSE_r(s, \hat{s}) + RMSE_r(\hat{s}, s)}{2}, \\
RMSE_{der_{spatial}} &= \frac{RMSE_{xyz}(s', \hat{s}') + RMSE_{xyz}(\hat{s}', s')}{2}, \\
RMSE_{der_{radius}} &= \frac{RMSE_r(s', \hat{s}') + RMSE_r(\hat{s}', s')}{2}, \\
RMSE_{curv} &= \frac{RMSE(curv(s), curv(\hat{s})) + RMSE(curv(\hat{s}), curv(s))}{2}, \\
L_{diff} &= |L_{ref} - L_{mod}|.
\end{aligned}$$

## 2.5.5 Results

### 2.5.5.1 Smoothing criteria comparison

For the optimization of the smoothing parameters  $\lambda_s$  and  $\lambda_r$  in the centerline approximation process, we compared the five state-of-the-art criteria presented in Section 2.3.3; the Akaike criterion ( $AIC$ ), the corrected Akaike criterion  $AICC$ , the Bayesian information criterion  $SBC$ , the cross-validation  $CV$  and the generalized cross validation  $GCV$ .

Figure 2.12 gives the result of this comparison. The distorted centerlines of the validation dataset are approximated using the five different smoothing criteria. Six quality metrics are aggregated to create two boxplots representing the error with regard to the smoothing criterion used and the input data point density, for the centerlines with spatial noise (respectively radius noise). The  $AIC$  and  $CV$  criteria are shown to be more robust to a low density of centerline data points than the other criteria, and  $AIC$  performs equally well to correct both spatial noise and radius noise. Based on these results, the Akaike criterion  $AIC$  was selected to perform the spline approximation of vessels.

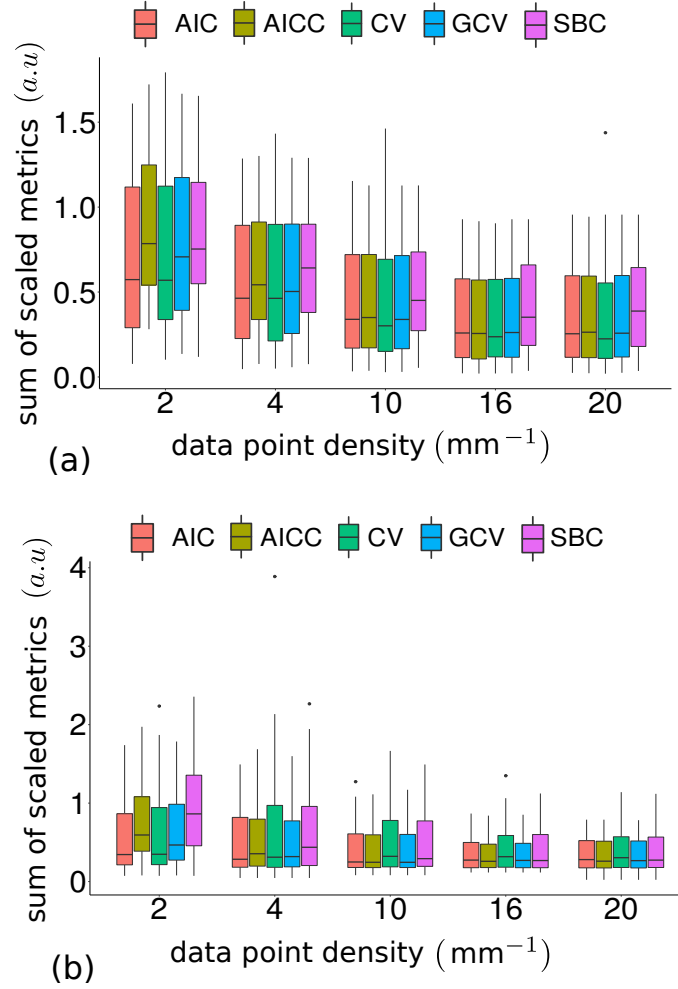


Figure 2.12: Comparison of five different optimization criteria for the smoothing parameter used for the penalized spline approximation. The six evaluation metrics were min-max normalized and summed. (a) shows the performance of the different criteria as a function of the data density for all centerlines with spatial noise. (b) shows the performance of the different criteria as a function of the data density for all centerlines with radius noise. Based on those results, we selected the Akaike criterion AIC.

### 2.5.5.2 Approximation methods comparison

As the spatial and radius distortions are not comparable in nature and magnitude, the evaluation results are presented in two different tables. Table 2.2 (respectively Table 2.3) shows the mean values of the six quality criteria for the four methods after radius noise (respectively spatial noise) addition. As expected, the non-penalized model (GNP) is sensible to the added noise and performs poorly for all radius-related metrics. In Figure 2.13, the radius estimation error is visible on the vessel produced by this method. In the

same way, the spatial-related metrics are impacted when spatial noise is added (Table 2.3). In addition, a tendency to overfit the data is observed in Table 2.2, causing a surprisingly high spatial error. The overfitting and noise problems are partially solved by optimizing the number of control points with the method GNP-AIC. However, this approach still yields a poor approximation of the derivatives: as the number of control points is lower, the space between data points might not be correctly interpolated, which particularly impacts the curvature values.

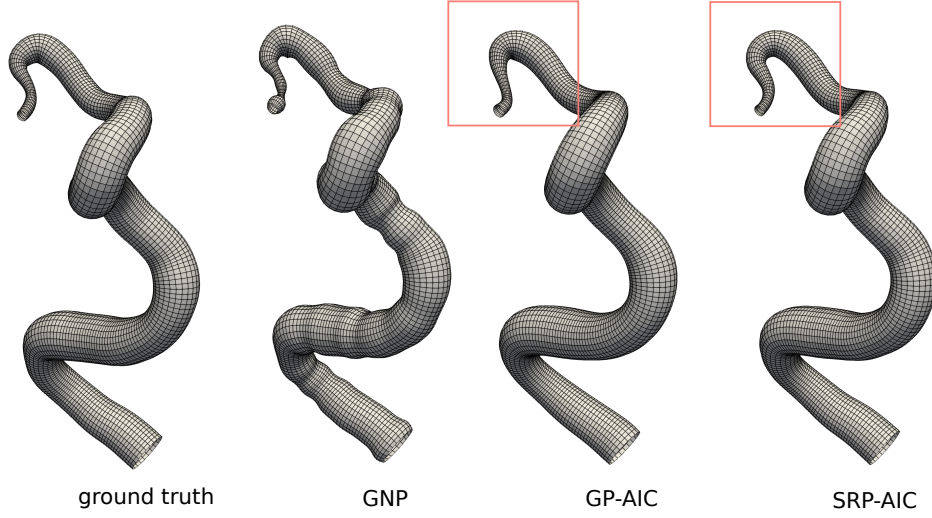


Figure 2.13: Mesh resulting from the approximation of distorted data ( $density = 1mm^{-1}$ ,  $\sigma_{radius} = 0.1$ ) by three of the methods compared in this evaluation.

The penalized approximations GP-AIC and SRP-AIC enabled to drastically improve the estimation of the derivatives and curvature. Finally, the advantage of SRP-AIC over GP-AIC is demonstrated both in the result tables 2.2 and 2.3 and in Figure 2.13. The global smoothing penalty used in GP-AIC forces a trade-off between the radius and spatial accuracy. In Figure 2.13, the radius of the vessel produced by GP-AIC is very similar to the ground truth vessel, but in return, the trajectory of the centerline was too smoothed. On the other hand, both the radius and trajectory of the vessel produced with SRP-AIC are closer to the ground truth. In conclusion, the proposed approximation method shows good robustness to the defects of the input data while enabling to simultaneously and accurately model the vessel centerline and radius. More detailed results are given in Figure 2.14.

Table 2.2: Overall evaluation of the approximation methods: mean values of the quality criteria for all the centerlines distorted by radius noise addition. The cells in gray corresponds to the lowest error for each metric.

	GNP	GNP-AIC	GP-AIC	SRP-AIC
$RMSE_{spatial}$	8.462	0.034	0.053	0.029
$RMSE_{radius}$	17.523	0.095	0.042	0.043
$RMSE_{der_{spatial}}$	0.218	0.118	0.042	0.009
$RMSE_{der_{radius}}$	0.391	0.214	0.032	0.032
$RMSE_{curv}$	1919.428	190.531	0.060	0.035
$L_{diff}$	718.906	0.057	0.207	0.004

Table 2.3: Overall evaluation of the approximation methods: mean values of the quality criteria for all the centerlines distorted by spatial noise addition. The cells in gray corresponds to the lowest error for each metric.

	GNP	GNP-AIC	GP-AIC	SRP-AIC
$RMSE_{spatial}$	0.511	0.152	0.099	0.096
$RMSE_{radius}$	0.008	0.009	0.018	0.007
$RMSE_{der_{spatial}}$	0.314	0.343	0.075	0.076
$RMSE_{der_{radius}}$	0.015	0.019	0.021	0.013
$RMSE_{curv}$	1.524	2.362	0.085	0.091
$L_{diff}$	50.180	15.071	0.252	0.207

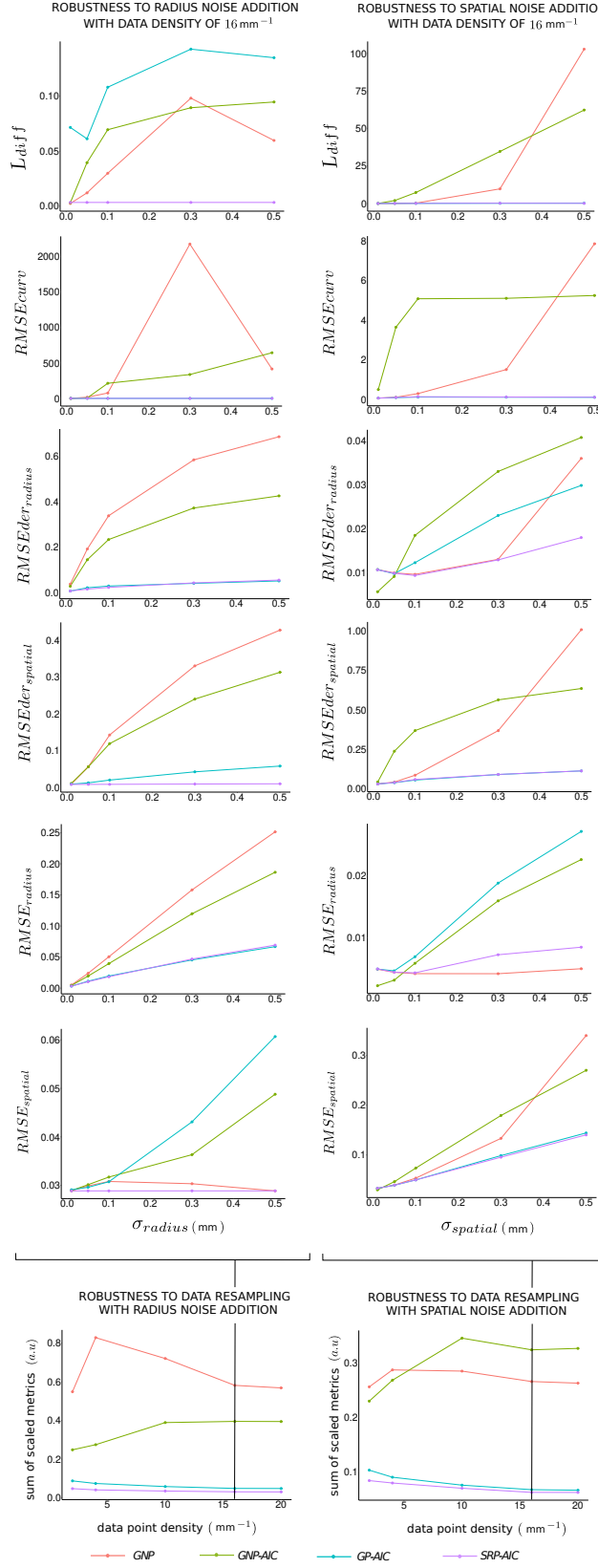


Figure 2.14: In the bottom, the six metrics were min-max normalized, summed up and plotted as a function of data point density, for spatial noise (right), and radius noise (left). For clarity, the plots of the six metrics were represented only for one data density value.



## 2.6 Conclusion

In this Chapter, we proposed a parametric model to reconstruct a continuous vessel surface from vascular centerlines only. Our method overcomes the main limitations of the state-of-the-art, which are the low number of data points and the radius/spatial noise, through two main contributions:

- The proposed vessel approximation method results in a better accuracy of the trajectory, radius and derivatives compared to common approximation methods of the literature, helping to overcome noise and low sampling of the input data. Unlike other methods in the state of the art, the proposed vessel model combines both spatial coordinates and radius in a single function.
- An anatomical model of bifurcation was employed to increase the realism of the bifurcation shape and robustness of the model to the misplacement of bifurcation data points compared to geometrical models. An automatic method is proposed to extract the bifurcation model parameters from centerlines only.

We acknowledge some limitations to the proposed modeling method. In the part of the vessels showing high curvature, the radius of curvature of the spline representing the vessel trajectory may become smaller than the approximation of the vessel radius. This causes an unrealistic fold of the vessel which can hinder the meshing step. To improve this point, a constraint on the spline curvature could be added during the optimization. As for the bifurcation model, some constraints on the position of the cross sections (e.g. max bifurcation angle) could help to reconstruct anatomically realistic even from very distorted input data points. Finally, our vessel model is based on the assumption of circularity of the vessel cross sections. If this assumption might be acceptable in arteries, it is a limitation for the modeling of other closed anatomical structures such as veins, whose cross section is often elliptic, or vascular pathologies such as stenosis or aneurysms.

---

# Meshing

## 3.1 Introduction

For CFD applications, the vascular geometry of interest is typically represented by a surface mesh, which is then divided into cells to form a volume mesh. The geometry of the surface mesh, but also the properties of the volume mesh, especially the type of cells (tetrahedral, hexahedral...) impacts the simulation computation and results. More specifically, Vinchurkar et al. [85], De Santis et al. [18] and Ghaffari et al. [25] compared the performances of hexahedral and tetrahedral meshes for different models (airways, coronary tree, and cerebral arteries) and applications. Those studies demonstrated that hexahedral meshes in general, and more specifically structured hexahedral meshes, converge faster and require fewer cells for the same accuracy of the results. De Santis et al. [18] and Ghaffari et al. [25] reported that 6 times fewer cells (resp. 10 times) and 14 times (resp. 27 times) less computational time were required. Finally, Vinchurkar et al. [85] insisted on the importance of having hexahedral flow-oriented cells for near-wall measurements (e.g. particle deposition, wall shear stress). The reason for this difference is thought to be two-fold :

1. The cells of hexahedral structured mesh are aligned in the flow direction which reduces the computational error due to numerical diffusion. On the other hand, unstructured meshes with randomly oriented cells are reported to introduce numerical diffusion in the solution.

2. As the vessels are tubular, the flow does not show strong variation in the longitudinal direction, but mainly in the radial direction. Therefore, the ideal mesh for numerical simulation has a low longitudinal cell density but a high radial cell density, This can be achieved easily with hexahedral meshes, but not with tetrahedral meshes.

The structure of hexahedral meshes is based on flow-oriented cross-sections. The advantages of this type of structure are not limited to CFD; it simplifies the boundary layer creation, bridges the gap between representation and physical simulation, and provides a basis for NURBS modeling [95] and isogeometric analysis. Hexahedral meshing, and more specifically structured hexahedral meshing, is however limited by a far more complex generation process than standard tetrahedral meshes [85]. In this chapter, we present a method we developed to create a structured hexahedral volume mesh with flow-oriented cells using the vessel and bifurcation models presented in the previous chapter.

## 3.2 State-of-the-art

In the literature, some methods have been proposed to mesh a triangular surface representing a vascular network into a fully hexahedral volume mesh. These methods rely on the information carried by the skeleton of the shape to find a decomposition of the surface that allows for the creation of a mesh with hexahedral cells. Some of the methods falling into this category were designed specifically for the meshing of vascular networks. In this specific case, the decomposition task is simplified as it is focused on the branching parts; the bifurcations are divided into three branches. A variety of approaches were proposed to obtain a robust decomposition. De Santis et al. introduced a semi-automatic method based on a user-defined bifurcation coordinate system [16], and Verma et al. defined the decomposition by solving Laplace's equation in the domain [82]. In this decomposition scheme, the quad surface produced is decomposed of a series of cross-sections along the vessels. This particular structure allows for the creation of a hexahedral volume mesh from the quad surface in a very simple way, by adjusting a grid template to the successive cross-sections. This enables the creation of a structured hexahedral mesh and makes it easier to parametrize the volume mesh for applications such as CFD. The size of the cells in the boundary layer, for instance, can be adjusted. However, if it is valid

for bifurcations, it is not easily generalized to any branching pattern (e.g. non-planar  $n$ -furcations, bifurcations close to each other).

The meshing can be generalized to complex branching shapes [4, 17, 90] or even any shape that can be represented by a centerline [50, 86], by allowing more complex decompositions of the surface. Automatic decomposition methods are based on Voronoi diagram [4], resolution of Laplace's equation [82] or random-walk algorithm [90]. The hexahedral meshing can then be created through various techniques; Copper scheme in the work of [4], and Laplacian-based harmonic functions combined with Catmull-Clark subdivision in [90]. De Santis et al. [17] proposed to generate a block-structure representation of the vascular network, which is then projected and refined to match the original surface. The method is limited by the optimization of the position of the initial blocks, which requires manual intervention. Viville et al. [86] proposed a robust decomposition algorithm based on the partition of a sphere, followed by optimization steps to match the original shape while keeping the quality of the hexahedral cells. This method was applied to the medical domain by meshing cerebral vascular networks and airways, as shown in Figure 3.1. It makes it possible to mesh complex branching patterns efficiently and was even extended to shapes with large planar surfaces.

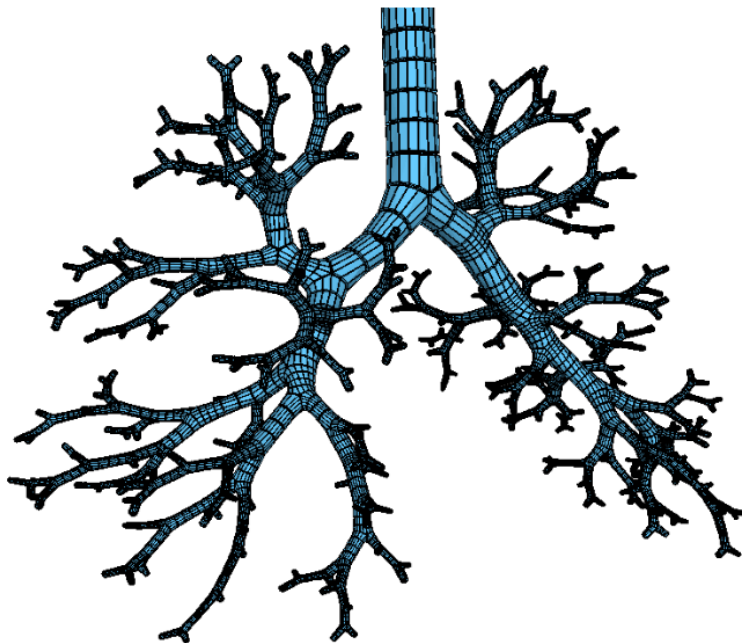


Figure 3.1: Hexahedral mesh of airways generated by the method of Viville et al. [86]

The main limitation of the application of the methods described above to our framework is that they rely on a triangular surface mesh as input for the extraction of a high-resolution skeleton and for the branch decomposition and meshing steps. Several meshing methods were proposed to create a surface or hexahedral volume mesh from centerlines only. This type of method often relies on modeling to compensate for the lack of geometrical information about the surface. In these methods, the model and the mesh are strongly linked; the parameters of the model often serve as parametrization for the meshing, and meshing methods can affect the shape of the domain. The majority of the proposed methods result in the creation of a triangular surface [28, 31, 43, 1] and would need to employ one of the meshing methods described above to obtain a hexahedral mesh. De Santis et al. [18] and Ghaffari et al. [25] proposed methods to create a hexahedral mesh directly from the centerlines, without resorting to triangular surface meshes. In [18], the authors proposed to select the most relevant cross-sections from the centerlines to build the structure of the hexahedral mesh. These cross-sections are used to define a branch decomposition scheme. In [25], the decomposition serves directly as a basis for the modeling of the bifurcations. The generation of a hexahedral mesh is simplified by the tubular assumption on which centerlines rely. The main limitations of this meshing procedure are the accuracy of the centerline data point position and radius, and the realism of the model proposed for the reconstruction of the vascular geometry.

In this work, we decided to follow the latter approach. If this meshing method is not easily generalized to any branching pattern, it produces structured meshes with parametrizable cross-sections directly from centerlines alone. We developed original methods to adapt this approach to the parametric model presented in the previous chapter. First, we compute a decomposition of the bifurcations into three branches. The decision not to rely on geometrical models of bifurcation in the modeling step complexifies the decomposition and the meshing, but in return, the shape of the surface does not depend on the meshing. Based on this decomposition, a surface mesh with quadrilateral faces is created. This surface mesh is finally converted into a hexahedral volume mesh by fitting an O-grid pattern to the cross-sections of the surface mesh.

### 3.3 Bifurcations

The vessels and bifurcations of the vascular network are meshed separately. In this section, we focus on the meshing of bifurcations parts. The method employed to mesh the vessels is presented in the next section.

#### 3.3.1 Decomposition

The bifurcations are the most challenging parts to mesh with hexahedral elements. The meshing approach proposed relies on a decomposition scheme that splits the bifurcation into three geometrical branches; one inlet branch and two outlet branches. This method allows us to recover the meshing advantages offered by the geometrical bifurcation models of other state-of-the-art methods, while keeping the anatomical realism of the anatomical bifurcation model used in this work. Figure 3.2 (b) gives an example of branch splitting using three separation planes. Antiga et al. [5] proposed a bifurcation decomposition scheme based on the Voronoi diagram of a surface mesh. The proposed scheme is robust to variations in input geometry and has been successfully used for hexahedral meshing [5]. However, it was not originally designed to obtain high-quality meshes but to offer a robust mapping of bifurcations. Moreover, it requires a surface mesh to be computed and its transposition to centerline data is not straightforward. Based on this work, we introduced a decomposition scheme that relies on the spline and bifurcation models described above.

In this decomposition method, three separation planes are defined by a set of five points; the apex point  $AP$ , which is already a parameter of the model, two center points  $CT_0$  and  $CT_1$  and two separation points  $SP_1$  and  $SP_2$ . As illustrated in Figure 3.2 (a), we first define the geometric center of the bifurcation  $X$ , as the barycenter of  $AP$ ,  $p_{m_1}$  and  $p_{m_2}$ , where  $p_{m_{1,2}}$  are the projection of the key points  $m_{1,2}$  located at the intersection of one centerline with the surface of the other vessel. The separation points  $SP_{1,2}$  are obtained by projecting  $X$  on the surface in the opposite direction from  $AP$ .

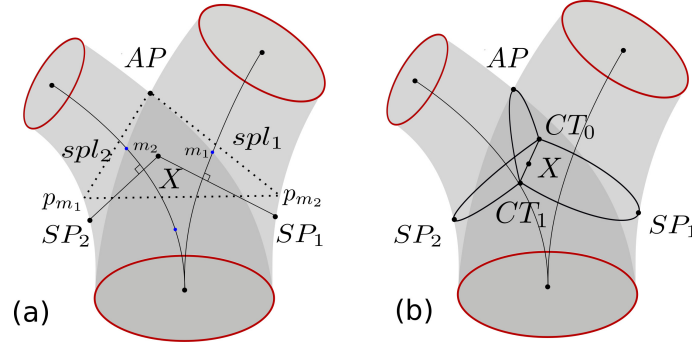


Figure 3.2: Geometric decomposition of the bifurcation model. In (b), the end cross-sections are represented in red and the separations planes in black.

Finally, the position of center points  $CT_0$  and  $CT_1$  is obtained by projecting  $X$  on the surface of the vessels. The direction of projection is normal to the plane defined by the three points  $AP$ ,  $SP_1$  and  $SP_2$ . The separation points  $AP$ ,  $SP_1$  and  $SP_2$  are finally connected to the center points  $CT_0$ ,  $CT_1$  by arcs, which delineate a geometrical frontier between the branches of the bifurcation (see Figure 3.2 (b)), providing the desired branch decomposition. This decomposition method enables us to handle large radius differences between the daughter vessels, as the barycenter  $X$  is naturally closer to the vessel with the smallest radius, which relaxes the angles between the separation planes and improves the quality of the resulting mesh.

### 3.3.2 Surface meshing

In this step, the surface mesh of the bifurcation is created using the separation planes defined in the previous section. First, we create an initial mesh grid that connects the end cross-sections to the separation planes with a set of successive sections, as illustrated in Figure 3.4. Each section of the mesh has a number  $N$  of nodes where  $N$  can be any multiple of 4. Figure 3.3 illustrates the initial mesh creation process. We first compute the  $N$  nodes of the end sections  $C_0$ ,  $C_1$  and  $C_2$ . A normalized reference vector  $ref_C$  which minimizes the rotation with the separation points  $SP_{1,2}$  is defined for each end cross-section  $C$ . The nodes of the end sections are placed on the outline of the cross-section with evenly spaced angles starting by  $ref_C$  and rotated counterclockwise. The nodes of the separation planes are positioned with equally sampled angles along the arcs connecting the separation point  $AP$ ,  $SP_1$  and  $SP_2$  to both center points  $CT_0$  and  $CT_1$ .

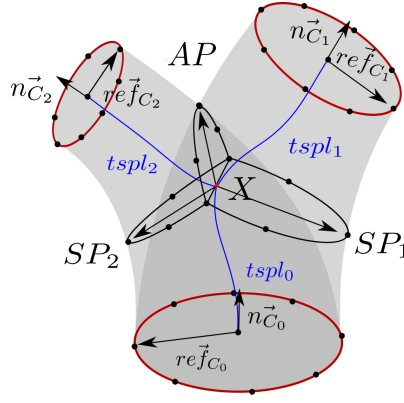


Figure 3.3: Computation of the nodes (black dots) of the end cross-sections and the separation planes for  $N = 8$  and splines  $tspl_{1,2,3}$ .

Once the nodes of the end sections and the nodes of the separation half-sections are computed, they are connected to form a surface mesh. The nodes of  $C_0$  are connected to the nodes of the half-sections defined by  $SP_1$  and  $SP_2$ , and the nodes of  $C_{1,2}$  are connected respectively to the nodes of the half-sections  $SP_{1,2}$  and  $AP$ .

We first define an initialization of the 3D trajectory that connects two nodes, as shown in the left column of Figure 3.4. This initialization is an approximation that is used to control the topology and geometry of the final mesh grid, but it does not necessarily lie on the exact surface of the bifurcation at this stage. The initial trajectories are evenly sampled with  $n$  nodes, where  $n$  determines the number of cross-sections to compute along a given branch. This number is proportional to the radius of the end section of the branch, by a coefficient  $d$  which can be adjusted to obtain the intended density of faces in the mesh.

The nodes are then projected radially to the surface of the two vessels, as illustrated in the right column of Figure 3.4. The direction of the projection is important to maintain the quality of the faces of the initial grid after projection. Ideally, the nodes of the initial trajectory must be displaced only radially from the center of the branch vessel. However, the shape splines  $spl_1$  and  $spl_2$  do not constitute a good approximation of the centerline of the three geometric branches. For this reason, we create another set of splines  $tspl_{1,2,3}$  connecting the center of each end section to the center  $X$  of the bifurcation, represented in blue in Figure 3.3. The nodes are projected to the surface of the bifurcation model according to the normal of this new set of splines.



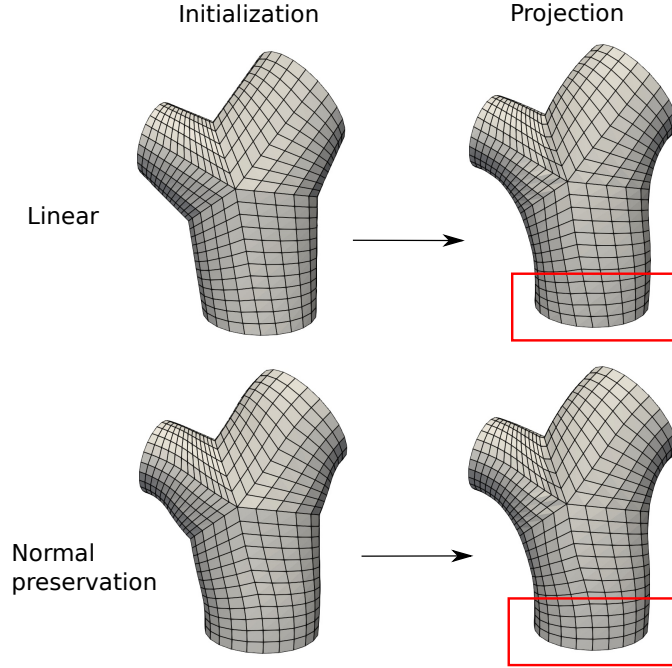


Figure 3.4: Initial surface mesh and mesh after projection for the two types of initialization considered. The red squares emphasize the impact of the two types of initialization on the final mesh.

The properties of the resulting mesh depend on the initial trajectory approximation. Figure 3.4 illustrates the meshes obtained after projection considering two types of initialization. The first row shows the simple case where the nodes of the end sections are linearly connected to the nodes of the separation geometry. In the second row, connection trajectories are computed so that the normal of the end sections is preserved in the output surface mesh. If both approximations allow to preserve the topology of the grid and the quality of the faces after projection, the initial trajectories with normal preservation are closer to the actual surface of the bifurcation, causing less displacement of the nodes during projection. Moreover, the preservation of the normal of the end sections facilitates the inclusion of the bifurcation mesh in larger arterial networks; the connecting curves can be smoothly extended to downstream vessels. In the rest of this work, we use the normal preserving initialization.

### 3.3.3 Relaxation

The projection step of the meshing method results in an uneven sampling of the nodes along the trajectories and can lead to faces with heterogeneous sizes or important skew-

ness. Moreover, a rupture of continuity is observed when the curves cross the separation between two branches. Those unwanted features are corrected by relaxation of the nodes of the surface mesh. Mesh smoothing methods are an easy way to reduce the skewness of faces but it triggers important deformations of the general shape of the model. To avoid deformations, [83] proposed to combine smoothing with a back projection on the surface. Following this approach, an iteration of Laplacian smooth (relaxation factor of 0.8) is first applied to the bifurcation mesh, then the nodes are projected back to the original surface. To prevent cross-sections from intersecting, the projection is made in the direction of the line connecting the center of the cross-section to the node to project. This process can be repeated until the relaxation is satisfying. Figure 3.5 displays a bifurcation mesh after 1 and 5 relaxation iterations. The faces are colored according to their geometric quality, measured by the scaled Jacobian. We observe that while the shape of the model is preserved, the quality of the faces near the separation plane is improved, and the trajectories smoothly cross the separation planes. Based on the average quality of the faces, we estimate that 5 relaxation iterations are sufficient.

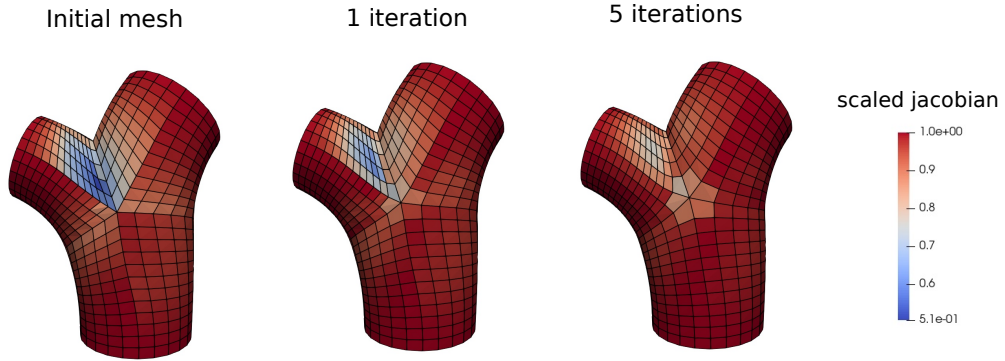


Figure 3.5: Relaxation of the bifurcation mesh; original mesh, and mesh after 1 and 5 relaxation iterations. The scaled Jacobian is used to measure the quality of the cells between  $-1$  (poor quality) and  $1$  (high quality).

### 3.3.4 Apex smoothing

The last step of the bifurcation meshing is the smoothing of the apical region. The model presents an unwanted sharp angle where the two vessels merge. The curvature in the apex regions impacts the pressure and velocity fields obtained by numerical simulation, as shown by [29]. Conventional mesh smoothing methods (e.g. Laplacian, Taubin smoothing)

are fast and can produce smooth meshes with high-quality faces. However, as they are global methods, they struggle to generate important local deformations. [93] proposed to smooth the apex region by projecting the nodes on a sphere of a given radius, rolling on the surface. This method is accurate, but it is computationally expensive and might not preserve the quality of the cells in the case of hexahedral meshes. Taking advantage of the topology of the proposed surface mesh, we reduced this complex 3-dimensional problem to a 2-dimensional problem.

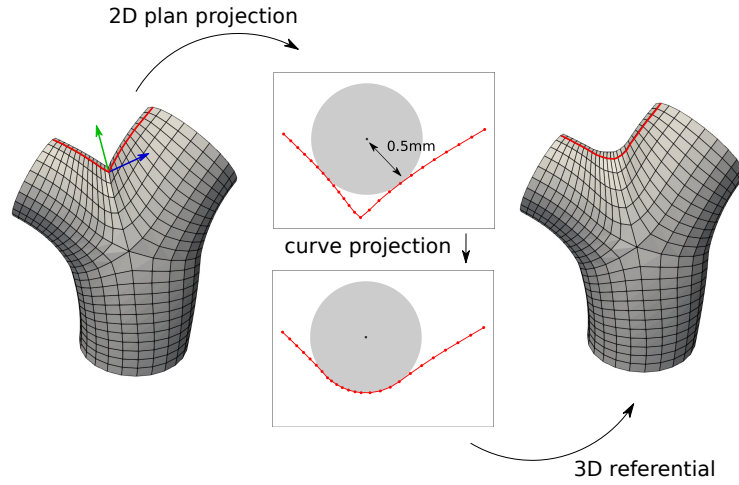


Figure 3.6: Apex smoothing pipeline.

Figure 3.6 illustrates the proposed smoothing method. The 3D polylines connecting two nodes of the end sections of the bifurcation are extracted (e.g. the curve in red on Figure 3.6). They are then projected on the 2D plane defined by the normal of the mesh at the separation point and the normal of the separation plane (resp. green and blue arrows on Figure 3.6). A circle whose radius corresponds to the desired apex radius of curvature is rolled along the 2D curves. The position of the circle where it is in contact with a further part of the curve is computed analytically. The points located under the circle are moved to the outline while preserving their original sampling. Finally, the new coordinates of the points are projected back on the original 3D referential to form the output surface mesh.

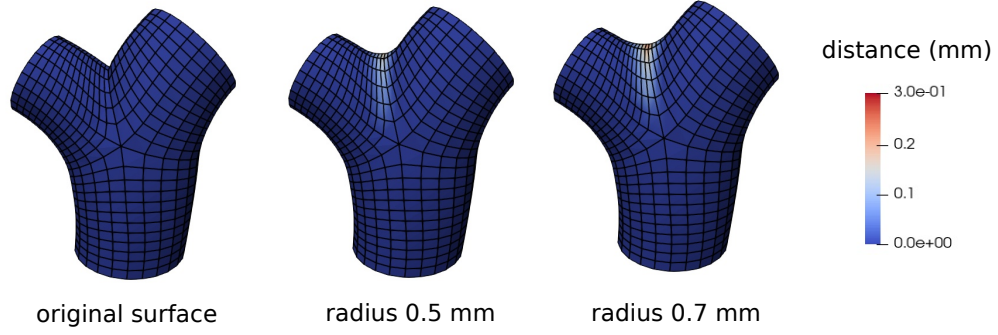


Figure 3.7: Apex smoothing with different radii of curvature. The colormap encodes the local distance to the original mesh, on the left.

The described smoothing method enables to control the direction of projection and the sampling of the projected nodes. Therefore, the quality of the faces is preserved. As shown in Figure 3.7, the smoothing is very local and does not affect the shape of the vessels outside of the apical region.

### 3.3.5 Extension to $n$ -furcations

In the same way that the model was extended to  $n$ -furcations, the meshing method used for bifurcations can be applied to  $n$ -furcations with  $n \geq 3$ , as shown in Figure 3.8.

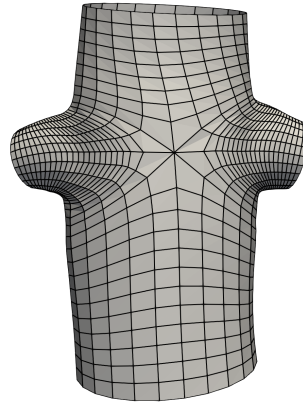


Figure 3.8: Mesh of a trifurcation.

However, the validity of the decomposition scheme described before is limited to the planar  $n$ -furcations, i.e.  $n$ -furcations whose branches lie approximately in the same plane. If the structured nature of the mesh created offers some advantages for numerical simu-

lation, it complicates the mesh creation. The conditions imposed on the topology of the faces make it hard to generalize to complex branching patterns. Meshing this type of pattern would require to adapt a hexahedral meshing method such as the method proposed by Viville et al. [86] to the vascular model we proposed.

### 3.4 Vessels

To mesh the surface of a vessel, the spline model is evaluated at a set of time values equally sampled in the  $[0, 1]$  interval. Those values are used to set the center position and radius of the cross-sections along the vessel (i.e. the longitudinal resolution of the mesh). The density of cross-sections (number of sections per mm) is proportional to the mean radius of the vessel, with a proportional coefficient  $d$  which is set by the user. From each center position,  $N$  nodes are radially projected on the model surface to form a circular cross-section. The projection vector is swept along the centerline and avoids twisting between the sections. The successive sections are connected to form the mesh faces. In the case of vessels connecting one bifurcation to another, a rotation is applied to the referential along the vessels so that the referential of the inlet section of the vessel matches the referential of the inlet section of the next bifurcation, as illustrated in Figure 3.9.

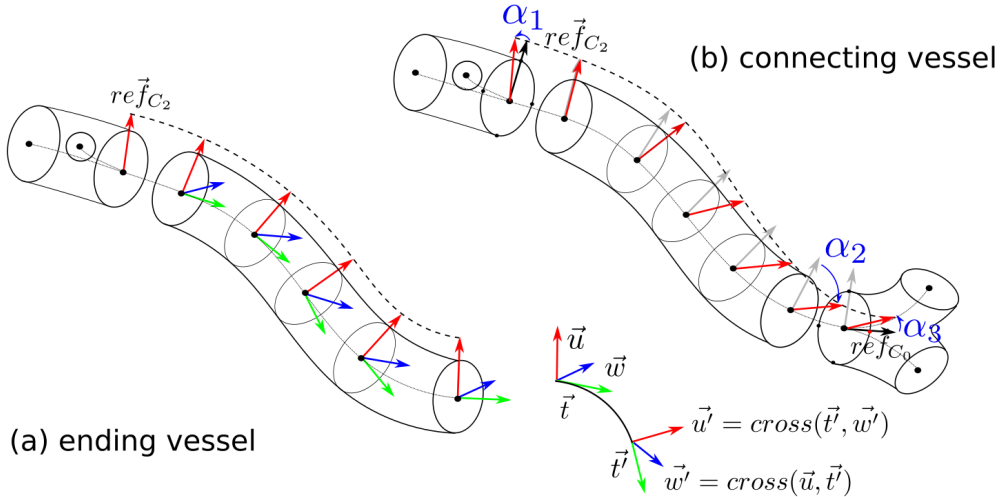


Figure 3.9: Rotation applied to the mesh referential to connect bifurcation to ends (a) and bifurcations to bifurcations (b).

### 3.5 Volume meshing

The volume of the vessel is meshed following the method of [25]. From each cross-section of the surface mesh, a structured O-grid pattern is created. It has 3 different areas, including the boundary layers, intermediary layers and central block. The relative size  $\alpha$ ,  $\beta$ ,  $\gamma$  of the areas, the number  $N_\alpha$  of boundary layers and the number  $N_\beta$  of intermediary layers can be adjusted. The separation planes of the n-furcations are handled by combining  $n + 1$  halves grids. The successive O-grid patterns are connected to form the hexahedral cells of the volume mesh, as shown in Figure 3.10.

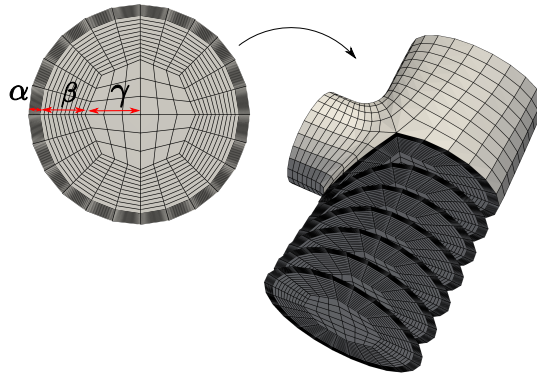


Figure 3.10: O-grid pattern and volume meshing method.

### 3.6 Validation

#### 3.6.1 Meshing success rate

To demonstrate further the possible applications of our method, we applied it to 60 patients of the BraVa dataset. The meshes produced for 8 of the patients are given in Figure 3.11. This dataset is considered challenging for several reasons. As the centerlines were extracted manually by medical doctors, they are noisy and have a low sampling. The superimposition of the centerline data points on the magnetic resonance angiography image in Figure 3.13 shows the high level of noise encountered in the input data, both in the radius estimation and the spatial positions. Besides, by computing the ratio of the number of data points on the total length of the connecting polyline, we estimated the average point density in the database to be  $0.45 \text{ mm}^{-1}$ , which is very low.

We evaluate the percentage of successfully meshed vessels and bifurcations separately,

as the meshing method is different. The vessels or bifurcations with at least one cell with a negative scaled Jacobian score are considered failed. With this strict definition, a total of 83% of the bifurcations and 92% of the vessels were successfully meshed.

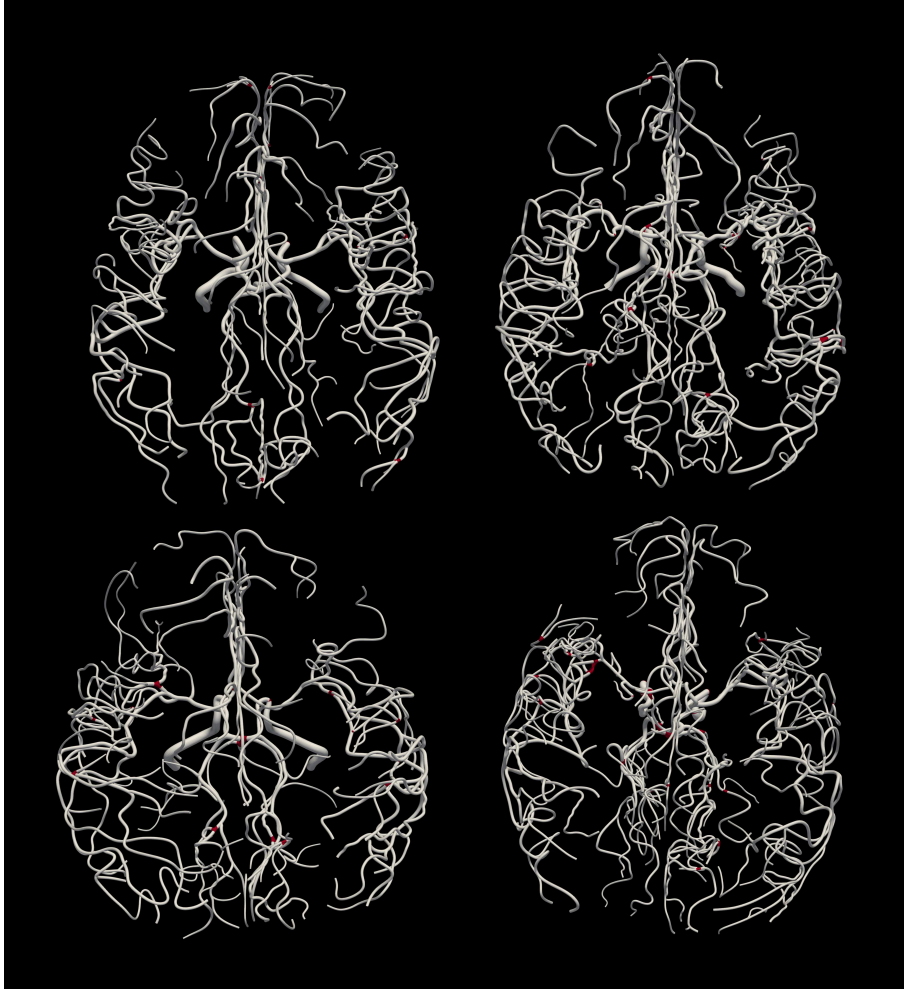


Figure 3.11: Top view of 4 meshes among the 60 generated from the patients of the BraVa database. The bifurcations where the meshing algorithm has failed (i.e. at least one of the cells has a negative Jacobian) are represented in red. The cross-sections of the vessels with cells of negative Jacobian values are also represented in red.

The main reason for the failure of the vessel mesh is a too-high curvature - mainly in the arteries with high tortuosity such as the internal carotid arteries - caused by a sharp angle in the input centerline. The main causes of failure for the bifurcations were very low bifurcation angles and misplacement of bifurcation points in the input data. This last case is illustrated in the image (b) of Figure 3.13. We can see that the bifurcation point in the centerline data was positioned too far downstream in the main vessel, causing one of



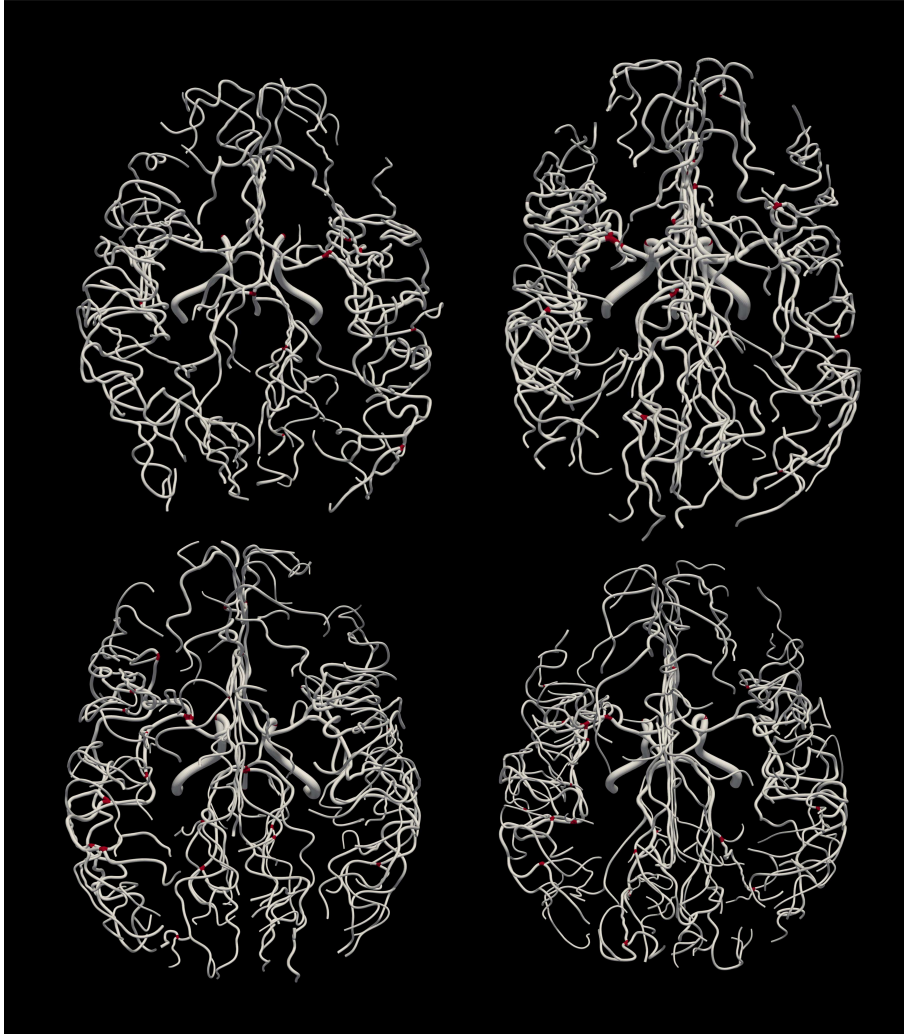


Figure 3.12: Top view of 4 meshes among the 60 generated from the patients of the BraVa database. The bifurcations where the meshing algorithm has failed (i.e. at least one of the cells has a negative Jacobian) are represented in red. The cross-sections of the vessels with cells of negative Jacobian values are also represented in red.

the daughter vessels to go backward from the direction of the flow with a sharp angle. As we use an oriented bifurcation model, it failed to correctly represent the geometry. The image (a) of Figure 3.13, on the other hand, illustrates a successful reconstruction of the trajectory of the vessel. Although the input centerline was very imprecise both in the radius estimation and point positions, we were able to produce a smooth model, closest to the vessel geometry as given by the medical image. Moreover, as shown in the insert of Figure 3.13 (a), even challenging topologies (e.g. short connecting segments between bifurcations) can be successfully meshed with hexahedral elements.



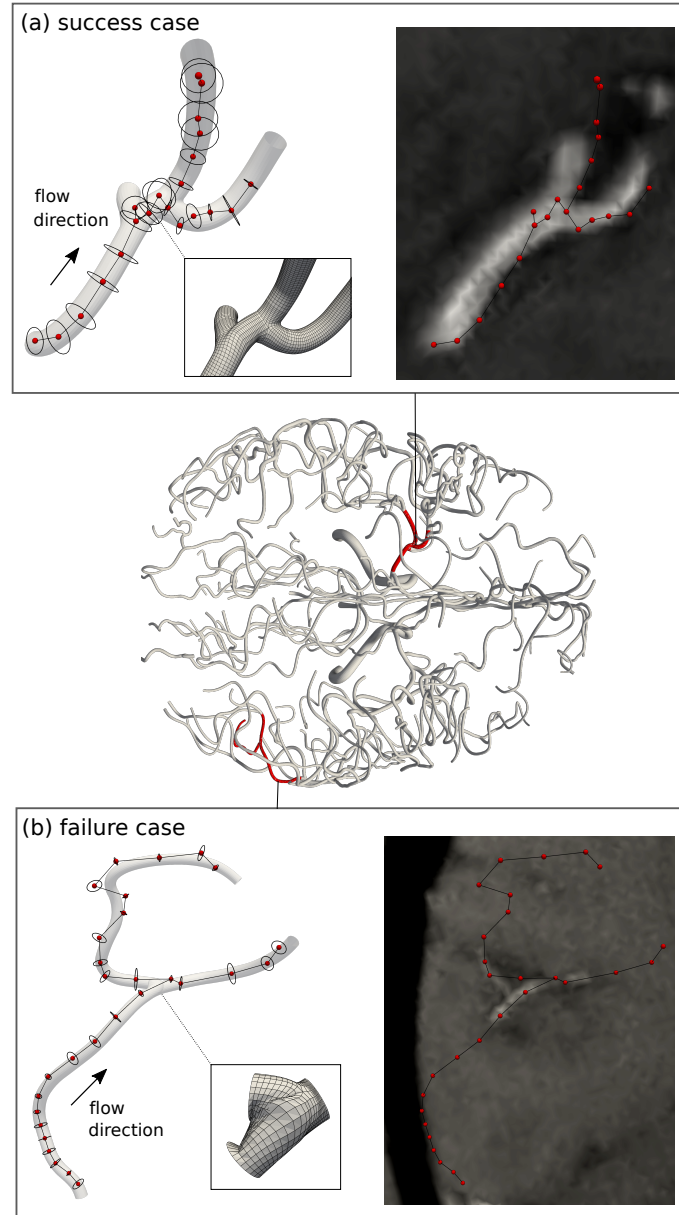


Figure 3.13: Example of success and failure of our method for one patient of the BraVa database. The whole-brain mesh is represented in the middle with a focus on two parts of the network. For each focus, the original centerline data points are represented by red dots (center) and black circles (radius). The mesh obtained is superimposed on the data points. On the right image, the original centerline data points are overlayed on the original MRA image.

### 3.6.2 Mesh quality

In CFD, the accuracy and stability of the simulation are affected by the quality of the mesh. To evaluate this quality, we computed the scaled Jacobian of the cells in the meshes

generated with the proposed method; this quality metric was detailed in Chapter 1. The scaled Jacobian ranges from  $-1$  (worst quality) and  $1$  (best quality). Negative values indicate invalid cells. The volume meshes for 60 patients from the BraVa database were generated, with the following parameters;  $N = 24$ ,  $d = 0.2$ ,  $\alpha = 0.2$ ,  $\beta = 0.3$ ,  $\gamma = 0.5$ ,  $N_\alpha = 10$ ,  $N_\beta = 10$ . The cells of the bifurcations and vessels are evaluated separately. Failed bifurcations and vessels were excluded from the study. The histograms of scaled Jacobian for the 60 patients are given in Figure 3.14, together with an example branch.

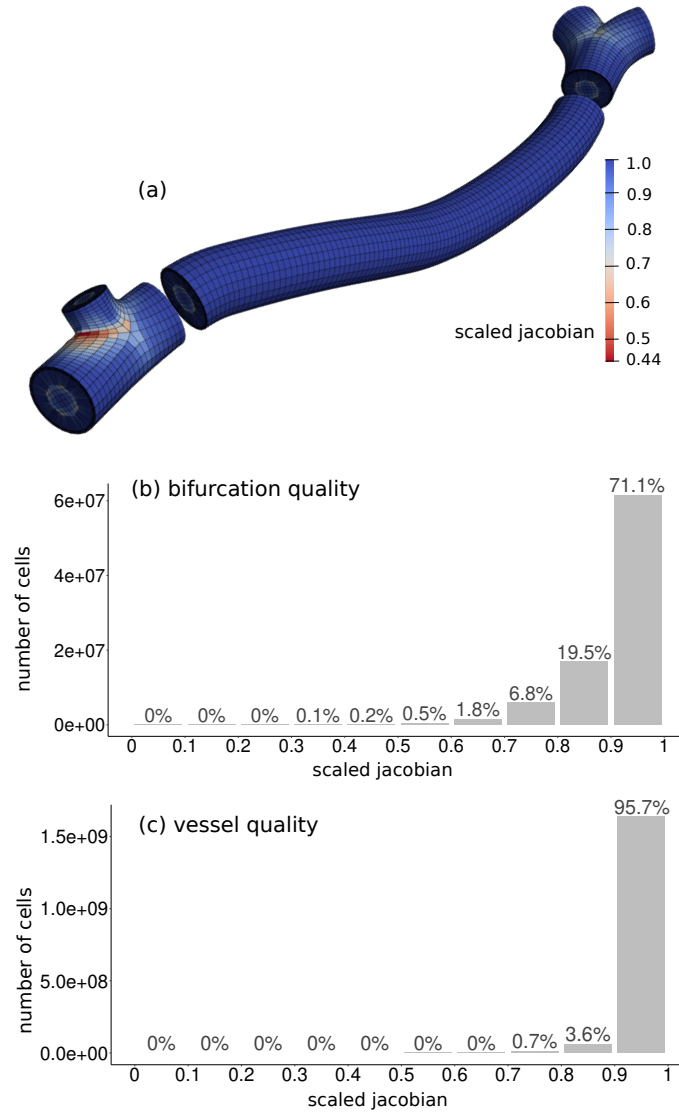


Figure 3.14: Distribution of the scaled Jacobian values of the mesh cells. Histogram (b) represents the quality of bifurcation cells and histogram (c) the quality of the cells of the vessels. Image (a) illustrates the location of high and low-quality cells within a mesh.

As shown in the image (a) of Figure 3.14, the bifurcations are the most challenging structures to mesh. The lower-quality cells are mainly localized in the bifurcation separation planes. Nevertheless, we achieved a very good overall quality for bifurcation cells, with 71% of the cells with a scaled Jacobian value higher than 0.9. The vessel cells have even better quality, with 95.7% of the cells having a scaled Jacobian higher than 0.9. In terms of mesh quality, our method improves the state of the art. Indeed, only 49% of the cells have a scaled Jacobian above 0.9 on average on the distributions given for three large cerebral networks in [25]. This proportion goes up to 62% of the cells of the abdominal aortic artery geometry meshed by the method of [90]. Finally, in [17], between 65% and 82% - depending on the case and the cell density - of the cells of the aortic arch meshed have a scaled Jacobian value between 0.8 and 1. Quantitatively, our method gives better results, especially given that the histograms for the other methods were computed on all the cells, both bifurcations, and vessels. However, we bear in mind that the study of [17] and [90] focuses on arterial geometries that differ from our study.

### 3.6.3 Computational time

The computational time of the modeling and meshing steps for five patients of the BraVa database was computed. The results for three of them are given in Table 3.1. The average time for modeling a large cerebral vascular network is about 16 minutes. The time for the volume meshing step is given in Table 3.1 for different cell densities. The average meshing time goes from 24.6 minutes for a coarse mesh to 49.7 minutes for a fine mesh. We want to stress that this study was performed on large networks, with a high number of bifurcations (around 100) and vessels (around 200). The meshing time increases with the number of bifurcations and vessels, while the modeling time is affected by the number of data points.

Besides, a large part of the meshing time corresponds to the computation of the surface nodes; on average 17.4 minutes for a coarse mesh and 34.8 minutes for a fine mesh. The volume mesh is generated directly from the nodes of the surface mesh without recomputing them. Finally, meshing can be run in parallel, by splitting the network into parts to be meshed on different CPUs. Using 12 CPUs, we were able to reduce the meshing computational times given in Table 3.1 by a factor of 5.

Table 3.1: Computational time required to model and mesh large vascular networks from the BraVa dataset.

id	furcation (#)	vessel (#)	data point (#)	modeling time (min)	cells (#)	meshing time (min)
P1	96	194	2816	11.3	1389k	20.4
					1853k	25.7
					2316k	31.2
					2779k	38.4
P2	101	203	3531	18.3	1916k	27.5
					2555k	38.2
					3193k	49.1
					3832k	67.5
P3	107	216	3474	16.8	1737k	26.3
					2316k	36.1
					2895k	44.2
					3474k	55.9

### 3.7 Conclusion

In this chapter, we proposed an automatic parametric method to create a structured hexahedral mesh of large cerebral vascular networks in a reasonable time. The mesh is created from the surface information given by the vascular model presented previously in Chapter 2. The produced mesh has high-quality flow-oriented cells, which makes it suited for CFD simulations. Moreover, this method gives more control over the distribution and density of the cells than the commonly used tetrahedral meshing. The main contribution of this work is the method proposed to mesh the bifurcations. Contrarily to other methods of the state-of-the-art, our meshing method for bifurcation does not require any surface mesh and relies on an anatomical model of bifurcation, which improves the anatomical realism of the bifurcation shape compared to the commonly used geometric models. Additionally, we included relaxation and smoothing steps to improve the quality of the cells without deforming the model shape. However, there are still some limitations to this meshing method as its success depends on the modeling step. Indeed, misplaced centerline data points close to the bifurcation can cause to inaccurate bifurcation model parameters. The position of the surface of the bifurcation is not well defined and this leads to the failure of the meshing step. For this reason, we believe that improving the modeling step by adding more constraints based on anatomical prior - as mentioned in the Conclusion section of

Chapter 2 - would solve the failure of the meshing step. Another main limitation of the proposed meshing method is that it is not compatible with complex branching patterns (e.g. non-planar  $n$ -furcations). If this type of branching is rare in the cerebral network, it restrains its application to other vascular networks such as the aortic system. The generalization of the proposed method to complex branching patterns would need a further generalization effort to maintain the structured nature of the mesh, or a complete shift in the meshing approach, by adapting general methods like the one proposed by Viville et al. [86]. Finally, the centerline-based meshing method proposed must be evaluated not only in terms of mesh quality or success ratio, but also against other state-of-the-art methods, including segmentation-based methods. In the next chapter, we perform some additional experiments to emphasize the advantages of the proposed method in the current state-of-the-art in vascular meshing.

---

# Comparison with segmentation-based methods

## 4.1 Introduction

Centerline-based meshing has the potential to offer several advantages in terms of meshing large vascular networks and usage of existing databases. However, segmentation-based meshing remains the most used method to create meshes for CFD. In Chapter 2 and Chapter 3, we introduced a method to obtain a hexahedral mesh from centerlines alone, using a parametric model to reconstruct a smooth and continuous vessel surface. In this chapter, we evaluate the centerline-based meshing method proposed by comparing it to recent segmentation-based methods of the literature. With this comparison, our goal is to verify the hypothesis that led us to the development of a centerline-based meshing method in the first place, by answering three questions : (1) Do centerline-based methods enable to reconstruct meshes for more complex and larger cerebral arterial networks than state-of-the-art segmentation methods? (2) How can this type of method address the current shortcomings of segmentation-based methods? (3) What are the specific advantages of using a hexahedral volume mesh over a tetrahedral mesh for CFD simulations?

Recently, a lot of effort has been put into developing new vessel segmentation algorithms (e.g. convolutional neural network-based methods), leading to a sharp increase in performances, as smaller and smaller arteries can be successfully segmented. However, in the literature, most authors evaluate the segmentation result by comparing it to a ground-truth segmentation using image-based metrics such as the Dice score. For numeri-

cal simulation, it is not the accuracy of the segmentation but the accuracy of the mesh that is essential. Indeed, the smoothness, the geometry, and the topology of the mesh surface can hinder the convergence of the numerical simulations and have a significant impact on the results, especially when it comes to near-wall measurements (e.g. wall shear stress) [91, 63]. In this work, we argue that even a segmentation that is very close to the ground truth might result in inaccuracies in the mesh it entails and that centerline-based methods can help to overcome these shortcomings. Besides, once a segmentation-based surface mesh is reconstructed, the most common method to discretize it into volume cells for CFD is to use tetrahedral elements, for their ability to mesh automatically any shape. However, several studies suggest that other types of mesh, especially the structured hexahedral mesh, offer many advantages for numerical simulations [85, 18, 25].

To support these assertions, we conducted two additional experiments whose results are presented in this chapter. In Section 4.3, we compare the surface mesh reconstructed by centerline-based and segmentation-based methods. For this, we re-implemented and trained two state-of-the-art deep neural networks to segment the vessels from MRA volumes. The segmentation-based meshes produced are quantitatively and qualitatively compared to the centerline-based meshes created with our method. In Section 4.4, we compare the CFD-related performance of the tetrahedral and hexahedral meshes. The performance of the different types of mesh is evaluated in terms of CFD simulation cost, convergence, and accuracy.

## **4.2 State-of-the-art**

Segmentation of medical images is the most common method to obtain patient-specific meshes for CFD. In recent years, deep learning-based models have led to significant advances in vascular segmentation. More specifically, convolutional neural networks (CNNs) have achieved very good performances [39, 80]. The popular U-net architecture [69] has been successfully applied to the segmentation of intracranial vessels [51, 65]. Hilbert et al. [34] proposed an extended U-net architecture using context aggregation and deep supervision for brain vessel segmentation. Besides, the attention mechanisms have been used to help the network to better learn global dependencies and increase the receptive field [48, 60, 62].

For medical applications such as CFD, more than the segmentation itself, the smoothness and the topological accuracy of the mesh it entails are critical. However, in the literature, there was very little focus on the conversion of the segmented volumes to mesh. Recently, Wickramasinghe et al. [88] and Kong et al. [44] introduced novel neural network architectures to go directly from a 3D volume to a 3D surface. Despite those recent advances, the meshing largely relies on algorithms such as the Marching Cubes to produce a surface mesh with tetrahedral elements, followed by a smoothing step [87, 58]. However, this type of segmentation-based meshes commonly suffers from topological inaccuracies (e.g. merging or disconnected vessels, bulges, missing vessels) and requires burdensome manual post-processing [27], as we demonstrated in the present chapter. Such problems are not correctly captured by the image-based metrics (e.g. Dice score) used to evaluate the segmentation methods. To overcome those challenges, the centerline representation of vascular networks has recently gained interest.

Some recent segmentation approaches propose integrating the vessel centerline information to build more topology-oriented metrics. Keshwani et al. [42] proposed to segment the vascular network from its skeleton by learning a connectivity metric between centervoxels. Besides, Shit et al. [76] introduced a novel topology-preserving loss for the training of neural networks, which relies on the centerlines of the predicted segmentation. The information provided by the centerlines allowed the neural network to improve the topology correctness of the segmentations. In this context, we believe that the use of centerlines in the meshing process can offer many advantages for CFD applications.

## 4.3 Surface mesh comparison

In this section, we compare qualitatively and quantitatively the vessel surface mesh produced by state-of-the-art deep learning segmentation-based meshing and the proposed centerline-based method.

### 4.3.1 Comparison pipeline

Centerlines can be extracted either from the grayscale image directly or from a segmented image. In this way, centerline-based meshing can be used either as a substitute or a complement to segmentation. In our comparison study, we investigated both approaches.



We took advantage of the BraVa database, which provides both raw MRA images and the associated manually-extracted centerlines to design a pipeline with two different evaluations, as illustrated in Figure 4.1; a quantitative and a qualitative approach. The MRA images are first segmented by state-of-the-art segmentation methods [51, 80]. The implementation details are given in the next section. A surface mesh is then produced from the segmented images by the marching cube algorithm. From this segmentation-based mesh, two different evaluations are proposed.

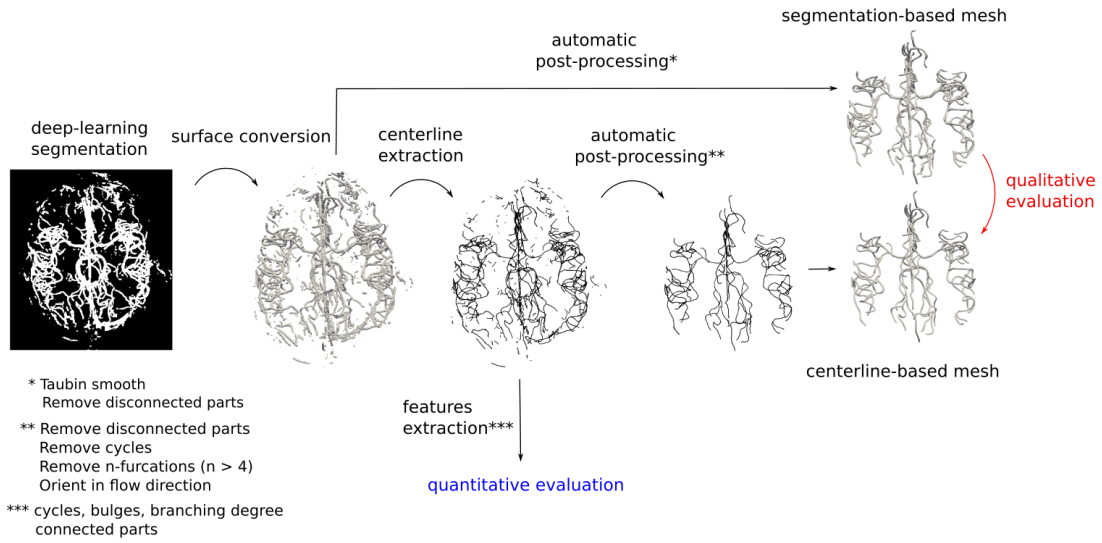


Figure 4.1: Pipeline used to compare segmentation-based meshing and centerline-based meshing.

In the quantitative evaluation part, centerlines are extracted from the segmentation-based mesh to analyze its geometry and topology based on a set of quantitative features (cycles, branching patterns...). The topological and geometrical features of the segmentation-based centerlines are then compared to the features extracted from the expert centerlines provided in the BraVa database.

In the qualitative evaluation part, the centerlines extracted from the segmentation-based mesh are post-treated so that they can be used as input for our centerline-based meshing method. The detail of the post-treatments applies in Section 4.3.4. From these post-treated centerlines, a mesh is created by centerline-based meshing methods. Two centerline-based methods are used to create meshes from those post-processed centerlines and the expert centerline; our explicit meshing method and the method of [1], which

is based on implicit structures and for which we used the Blender plug-in provided by the authors, in metaballs reconstruction mode. Parallely, the segmentation-based mesh itself is post-treated by the automatic methods commonly applied in the literature. It is first smoothed by a Taubin filter and the small components of this mesh are removed to keep only the largest connected parts by a connectivity analysis. The post-treated centerline-based meshes and the segmentation-based meshes are then visually compared. The segmentation-based mesh is also compared visually to the centerline-based meshes created from the BraVa expert centerlines directly.

### 4.3.2 Datasets and model training

In this section, we detail the image datasets used for the evaluation and the implementation of the segmentation methods. Two databases of whole-brain MRA images for healthy patients are used for this evaluation: The BraVa database gives access to 62 expert centerlines and 50 MRA, and the TubeTK database [10] is composed of 34 MRA and the 34 in-house expert segmentations associated. These databases were chosen because they include the same type of images while offering complementary information (resp. expert centerlines and expert segmentation).

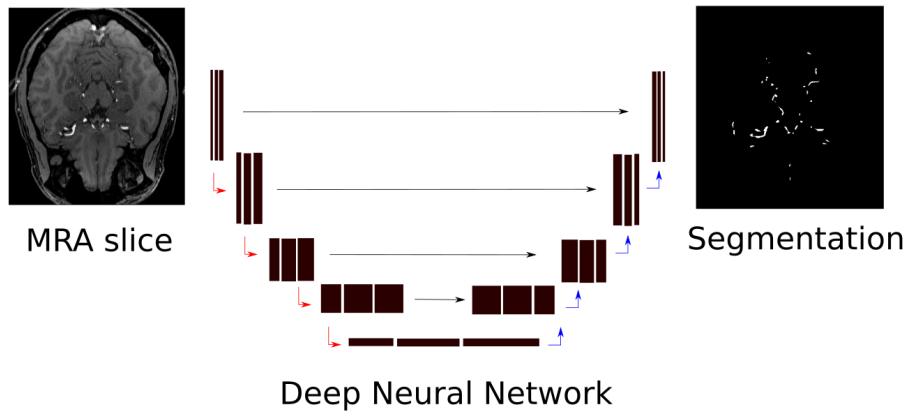


Figure 4.2: Principle of convolutional neural network-based image segmentation.

For the vessels segmentation, two state-of-the-art methods were implemented, both based on neural networks: DeepVesselNet [80] and U-net [51] (see Figure 4.2). This re-implementation work was done by Pierre Rougé, a Ph.D. student of the laboratory CRE-ATIS. The DeepVesselNet architecture provided by the authors was used (<https://github.com/giesekow/deep>

and the U-net neural network was re-implemented. The details of these methods can be found in their respective original articles. Both models were trained on the expert segmentations of the TubeTK database. 27 images were included in the training set and 7 in the test set. The loss function used during training was a combination of Dice loss and cross-entropy loss. Extracted patches of size  $96 \times 96 \times 96$  were used during the training phase. The stochastic gradient descent algorithm was used for the optimization, with a learning rate of 0.01 for U-net and 0.001 for DeepVesselNet. The batch size was set to 5 for U-net and 10 for DeepVesselNet due to memory constraints. These hyperparameters were set empirically by testing a large selection of values for each hyperparameter. The U-net was trained for 200 epochs and DeepVesselNet for 300 epochs.

For this comparison study, we considered the segmentations produced for the 7 images of the test set of TubeTK and the 50 MRA of the BraVa database -segmented with the models trained on TubeTK-. Table 4.1 summarizes the data type and the number of patients of the different datasets created for this comparison study.

Table 4.1: Description of the different datasets used in our comparison study. The name of the original database (BraVa or TubeTK), the method used, the nature of the data, and the number of patients are given.

Database	Method	Data	nPatients
BraVa	DeepVesselNet	Segmentation	50
BraVa	Expert	Centerlines	62
BraVa	Unet	Segmentation	50
TubeTK	DeepVesselNet	Segmentation	7
TubeTK	Expert	Segmentation	34
TubeTK	Unet	Segmentation	7

### 4.3.3 Quantitative evaluation

#### 4.3.3.1 Topological and geometrical features

A total of 9 topological and geometrical features were computed to perform the quantitative evaluation of the segmentation-based centerlines compared to the expert centerlines.  $nBulges$  corresponds to the number of bulges in the geometry. It is obtained by counting the number of ending segments smaller than the vessel diameter.  $nBranch$  is the number of branches in the entire network.  $nCC$  is the number of connected components and  $nBranchMaxCC$  is the size of the largest connected component, evaluated by its number of branches. This metric highlights the disconnected vessels and small isolated parts in

the mesh.  $nCycle$  is the number of cycles of the network. The only cycle in the cerebral vascular system is the circle of Willis, so the number of cycles should be either 1 (complete circle of Willis) or 0 (incomplete circle of Willis). Finally, the branching topology of the network is analyzed via the number of bifurcations  $nBif$ , the number of trifurcations or more  $nTrif+$  and the minimal (resp. maximal) furcation degree  $minDeg$  (resp.  $maxDeg$ ), i.e. the number of in and out branches (bifurcations = 3). These metrics are reported in Table 4.2 for the different datasets considered.

#### 4.3.3.2 Results

The results of the quantitative evaluation are given in Table 4.2, where the value obtained for the metrics proposed are reported for the different datasets.

Table 4.2: topological and geometrical features of the segmentation-based meshes for the different datasets. For each dataset, the median value between all patients is given.

Database	Method	nBulges	nBranch	nCC	nBranchMaxCC	nCycle
BraVa	DeepVesselNet	43	369	451	59	26
BraVa	Expert	0	205	1	205	0
BraVa	Unet	38	504	380	200	44
TubeTK	DeepVesselNet	52	552	557	212	46
TubeTK	Expert	12	551	26	508	98
TubeTK	Unet	34	626	300	446	85

Database	Method	nBif	nTrif+	minDeg	maxDeg
BraVa	DeepVesselNet	76	15	0	5
BraVa	Expert	102	1	1	4
BraVa	Unet	136	30	0	6
TubeTK	DeepVesselNet	140	28	0	6
TubeTK	Expert	230	48	0	6
TubeTK	Unet	215	46	0	6

We observe in Table 4.2 that the expert centerlines (in gray) do not have any small ending segments ( $nBulges = 0$ ). The network forms a single connected component, there are no isolated vessels and no cycles. In the BraVa expert centerlines, the posterior communicant arteries were willingly not included in the centerlines to obtain a tree structure, which explains the absence of cycles in the networks ( $nCycle = 0$ ). Besides, the branchings are mainly bifurcations, as expected in the cerebral vascular system where trifurcations are rare. The expert centerlines show no branching with a degree superior to 4 (= trifurcations).

On the other side, segmentation-based meshes present a lot of bulges ( $> 12$ ) and cycles ( $> 26$ ), mainly because closed vessels are merged in the resulting mesh. The number of trifurcations and higher degree branching is high ( $> 15$ ), and furcations with up to 6 branches were observed. These metrics bring to light some inaccuracies in the topology of the meshes produced by segmentation, which will affect the mesh geometry and therefore the CFD simulation results. It is interesting to see that such problems - disconnected or merged vessels, bulges - are observed even in the meshes based on the ground truth segmentation made by medical doctors (TubeTK/Expert row in Table 4.2). They are not only caused by the segmentation method but also by the nature of the meshing process itself, as it relies on the segmentation of low-resolution images. Moreover, no distinction can be made between veins and arteries in the segmentation process, which might cause peculiar topology in the network. To run numerical simulations in such segmentation-based meshes, important post-processing is required to isolate the arterial system and reconnect or separate vessels. The topological problems highlighted here are illustrated by enhanced visualizations of the meshes (Figures 4.3 and 4.4) in the next section.

#### 4.3.4 Visual evaluation

##### 4.3.4.1 Centerline extraction and post-treatment

In this paragraph, we give more details about the post-treatment that the segmentation-based centerline underwent to match the input requirements of our centerline-based meshing software. All those post-processing were automatic. The small ending segments (bulges) were first identified and removed. Then, the cycles were removed by computing a spanning tree which maximizes the length and minimizes the angle between branches. To allow for the separation of merged branches, a small clique is added at the  $n$ -furcation ( $n > 3$ ) points before the computation of the spanning tree. Finally, the edges connecting the data points are oriented in the flow direction.

##### 4.3.4.2 Results

In this part, the results from our method are visually compared against centerline-based meshes produced by the method of Abdellah et al. [1] and the segmentation-based meshes produced by Unet [51] and DeepVesselNet [80]. A BraVa patient was selected for

this visual evaluation to have access to the expert centerlines associated. We selected a segmentation made with Unet as it provided better results than DeepVesselNet on this database. Figure 4.3 shows the whole brain meshes obtained by different methods, and Figure 4.4 shows enhanced visualization of some relevant parts. As shown in Figure 4.3 (a), the deep learning segmentation algorithm demonstrates a good ability to segment a large part of the vascular network, including small segments. However, the algorithm does not guarantee the connectivity of the network, and some post-processing filtering is needed to remove the small isolated parts (Fig. 4.3 (b)). The centerline-based methods (images (d) and (e)) were able to produce meshes with a topology similar to the segmentation-based mesh from which the centerlines were automatically extracted. We can see in this figure that the mesh produced by our method is smoother than the other meshes; the geometrical quality of the meshes obtained by different methods will be further discussed below. As illustrated in the image (f), the manually extracted expert centerline allowed to reconstruct a larger arterial network with smaller vessels than the centerlines based on the segmentation results. Note that, overall, the radius of the expert centerlines is smaller, due to the extraction method [52].

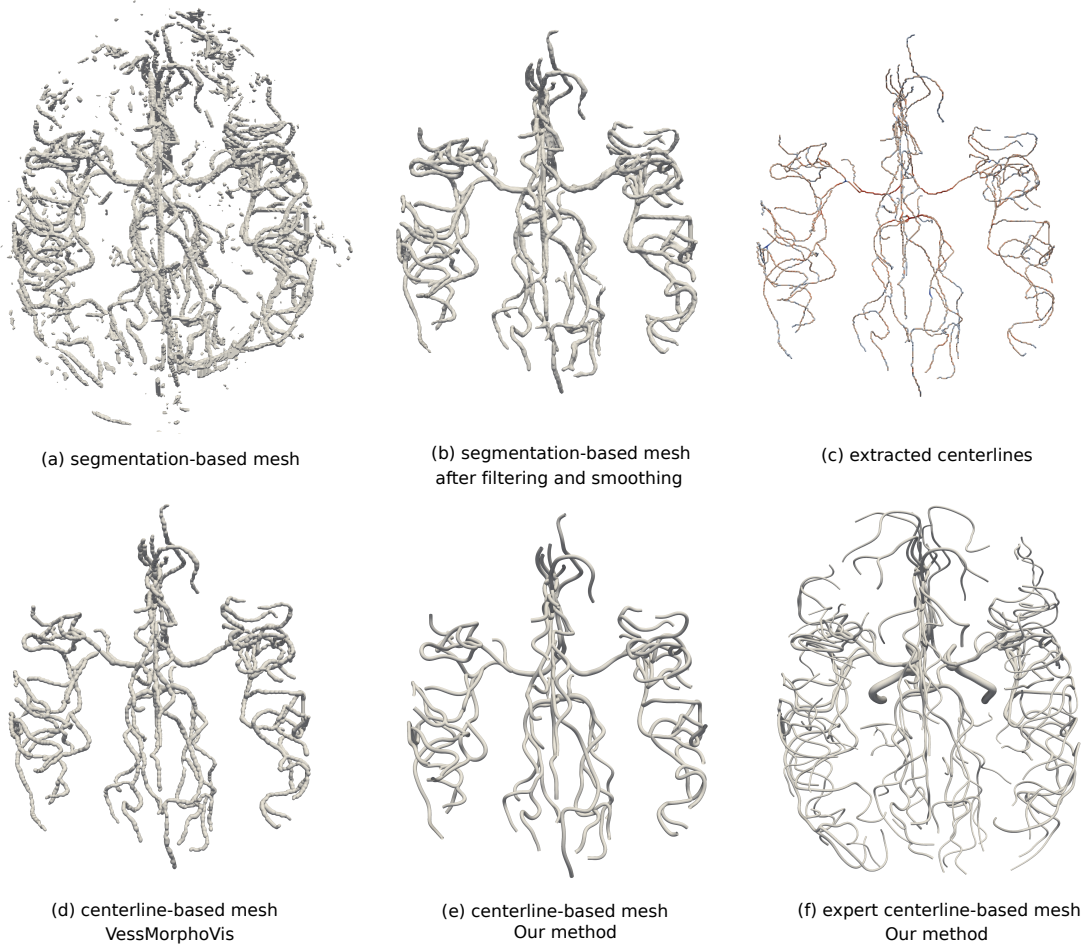


Figure 4.3: Meshes produced with segmentation-based and centerline-based methods for a patient of the BraVa database. From left to right and top to bottom: original mesh created from the segmentation with Unet, the same mesh after filtering of the smallest components and smoothing; centerlines extracted from the mesh after post-processing; mesh produced by VessMorphoVis from these centerlines; mesh produced by our method from these centerlines; mesh produced by our method from manually extracted expert centerlines.

In View 1 of Figure 4.4, we observe that our method, which relies on the vessel tubularity assumption, efficiently cleaned the vascular network from the bulges observed in the segmentation-based mesh. The radius and trajectory smoothing allows for reconstructing the disconnected parts in a very natural way. View 2 highlights the merging vessels and cycles observed in segmentation-based meshes. Our automatic post-treatment of the centerlines allowed removing unwanted cycles in the network. The implicit method VessMorphoVis [1] offers good flexibility to mesh complex geometry such as pathology or complex branching patterns. However, the surface of the vessels appears bumpy and

irregular. Moreover, as shown in both views of Figure 4.4, this method appears to be sensitive to noise on the centerline geometry and radius. As it is robust to noise and data sparsity, our method improves the smoothness and realism of the vascular geometry.

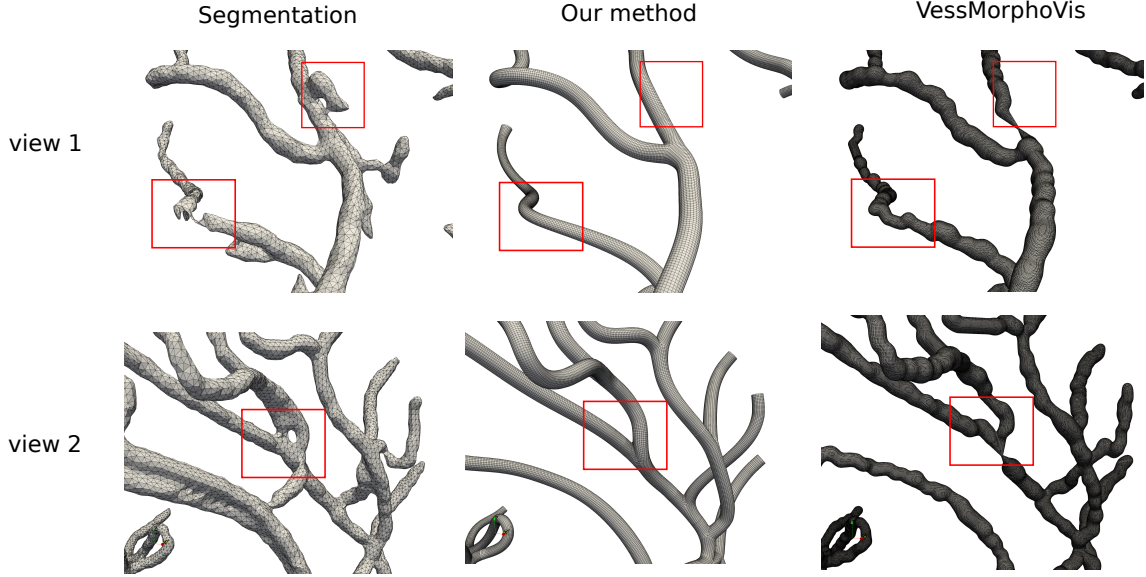


Figure 4.4: Enhanced visualization of the segmentation-based mesh, the centerline-based mesh obtained with our method and the centerline-based mesh obtained with VessMorphoVis. The red squares highlight the parts where the meshes show important differences.

## 4.4 Volume mesh comparison

In this section, we compare the CFD-related performance of the different types of volume meshes. For the same surface geometry, we compare the hexahedral mesh created with our method to the tetrahedral mesh generally used by segmentation-based methods. The performance of the meshes is evaluated in terms of CFD simulation cost, convergence, and accuracy.

### 4.4.1 Tube model

As a first step, the comparison was conducted in a straight tube model. This setting allows a comparison to the analytical form of the sectional velocity profile given by the Poiseuille equation, and therefore the evaluation of the accuracy of the results.



#### 4.4.1.1 Geometry

We created a tubular model which mimics the middle carotid artery, where cerebrovascular pathologies often occur. The diameter of the tube was set to  $D = 2.5\text{mm}$  [66]. To ensure to get a fully developed flow in our experiment, as detailed in the next section, we set the tube length to  $L = 200\text{mm}$ . The surface mesh of our tube model ( $D = 2.5\text{mm}$ ,  $L = 200\text{mm}$ ) was produced by our meshing method with a very fine face density, to capture correctly the cylindric shape. From this surface, two types of volume meshes were produced; tetrahedral and hexahedral meshes. For each type, meshes with different cell densities - from coarse to fine - were created.

#### 4.4.1.2 Tetrahedral meshing

To create the tetrahedral meshes, the initial surface composed of rectangular cells was first triangularized and remeshed to mimic the triangular, unorganized faces of segmentation-based meshes. The mesh was then discretized with tetrahedral cells using the software TetGen® [32], a state-of-the-art tetrahedral meshing software often used to produce the volume mesh in blood flow studies [79, 74]. Five meshes of various cell densities were created by adjusting the parameter controlling the maximal size of the cells. Some of the tetrahedral meshes created are shown in Figure 4.5, and their properties are given in Table 4.3.

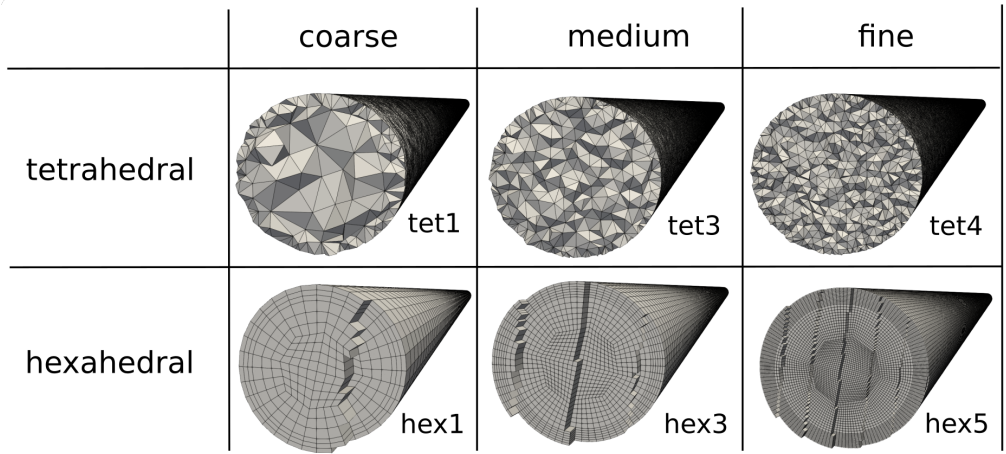


Figure 4.5: Meshes produced for the CFD study of the tube model. Tetrahedral and hexahedral meshes of different densities are represented. The mesh id as given in Table 4.3 is written under each image. The meshes were cut using an oblique plane to reveal the inside cells.

#### 4.4.1.3 Hexahedral meshing

To create the hexahedral meshes, we employed the meshing method proposed in the present chapter. In our hexahedral meshing method, the cell density can be adjusted by 6 different parameters; the number of nodes in the circumference of the cross sections  $N$ , the radius-dependent longitudinal density  $d$ , the thickness ratio of the boundary and intermediary layers areas  $\alpha$  and  $\beta$ , the number of boundary layers  $N_\alpha$  and intermediary layers  $N_\beta$ . However, the variation of each parameter separately modifies the cell shape, which might have a different impact on the simulation results. For this reason, those meshing parameters were set so that the edge length ratio of the hexahedral cells is maintained as we increase the cell density. The geometry of the hexahedral cells can be characterized by three characteristic lengths; the longitudinal length  $\lambda_l$ , the circumferential length  $\lambda_c$  and the radial length  $\lambda_r$ , whose respective equations are

$$\lambda_l = \frac{D \times d}{2}, \quad (4.1)$$

$$\lambda_c = \frac{\pi D}{N}, \quad (4.2)$$

$$\lambda_r = \frac{\alpha \times D}{2(N_\alpha + 1)}. \quad (4.3)$$

To simplify the estimation of the meshing parameters, the thickness ratio of the different layer areas were fixed respectively to  $\alpha = 0.2$  and  $\beta = 0.4$ . As the number of nodes in the circumference of the cross sections  $N$  is a parameter that has the strongest setting constraints, as it must be set as a multiple of 8,  $N$  was fixed first. The other meshing parameters  $d$ ,  $N_\alpha$  and  $N_\beta$  are then estimated from the equations above so that the ratio of the cell's characteristic lengths  $\lambda_l$ ,  $\lambda_c$ ,  $\lambda_r$  matches a target ratio which characterizes the cell shape. In this study, the cell ratio  $(\lambda_l, \lambda_c, \lambda_r)$  was set to  $(2, 1, 0.6)$  for the cells of the intermediate layers and to  $(2, 1, 0.3)$  for the cells of the boundary layers. A total of five meshes of various densities were produced by selecting the following values for the parameter  $N$ ;  $(32, 48, 64, 96, 128)$ . Some of the hexahedral meshes created are shown in Figure 4.5.

## 4.4.1.4 Mesh quality

Table 4.3: Table of the properties of the tetrahedral and hexahedral mesh generated. The final residuals after convergence of the simulation are also reported in this table.

id	type	cells (#)	$SJ_{min}$	$SJ_{mean}$	$SJ_{max}$	final residuals
tet1	tet	894871	0.009	0.397	0.982	$10^{-6}$
tet2	tet	1489391	0.011	0.429	0.998	$10^{-6}$
tet3	tet	2158463	0.008	0.454	0.999	$10^{-5}$
tet4	tet	5228811	0.006	0.540	0.999	$10^{-5}$
tet5	tet	9082376	0.006	0.540	0.999	$10^{-5}$
hex1	hex	104448	0.707	0.960	0.995	$10^{-6}$
hex2	hex	381888	0.707	0.969	0.999	$10^{-6}$
hex3	hex	834560	0.707	0.971	0.999	$10^{-6}$
hex4	hex	3052608	0.707	0.976	0.999	$10^{-6}$
hex5	hex	7093760	0.707	0.976	0.999	$10^{-6}$

Table 4.3 summarizes the different meshes created and their characteristics. The meshes cover a similar range of the number of cells, from 104k to 9082k cells, allowing for comparison. The quality of the internal cells, measured by the scaled Jacobian value, is higher for hexahedral cells ( $> 0.7$ ) than for tetrahedral cells (mean of 0.47). The meshing time is also lower for hexahedral meshes, especially for meshes with a high cell density, as shown in Figure 4.6.

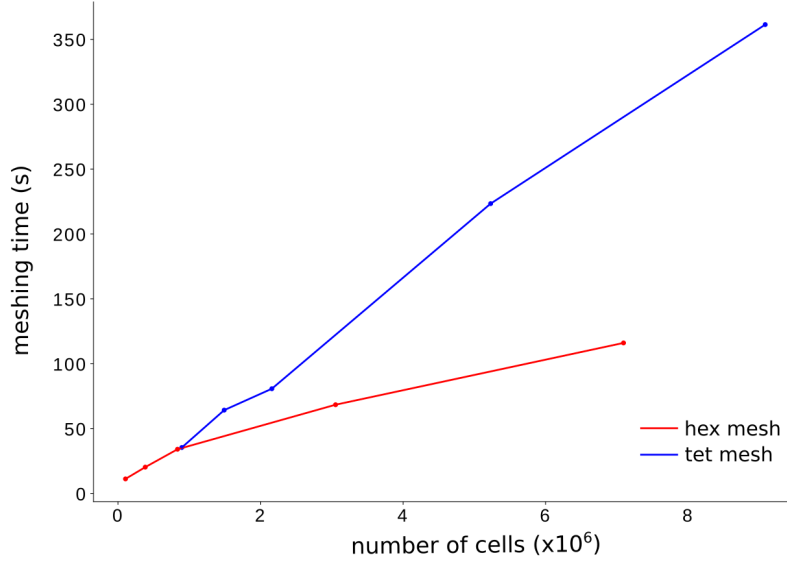


Figure 4.6: Meshing time as a function of the number of cells for tetrahedral and hexahedral meshes.

#### 4.4.1.5 CFD simulations

The following properties of blood were considered; a density of  $\rho = 1053\text{kg.m}^{-3}$ , a dynamic viscosity  $\mu = 0.0035\text{kg.m}^{-1}.\text{s}^{-1}$ , and a kinematic viscosity of  $\nu = \frac{\mu}{\rho} = 3.324 \times 10^{-6}\text{m}^2.\text{s}^{-1}$ . The blood was considered Newtonian and the inlet velocity was set to  $0.2\text{m.s}^{-1}$  [49]. In these experimental conditions, the Reynolds number computed with the formula given in Chapter 1, Equation 1.2 is  $Re = 150.43$ . The minimal length  $L_d$  to consider to obtain a fully developed flow can be estimated by the following formula;

$$L_d = 0.06 \times Re \times D. \quad (4.4)$$

In our flow settings, we have  $L_d = 22.56\text{mm}$ . The length of the tubular model was set to  $L = 200\text{mm}$ , which ensures that the flow is fully developed. The flow was assumed laminar, which is justified by a Reynold number of 150.4. The outlet boundary condition was set to zero pressure. The residuals value for convergence was fixed to  $10^{-6}$ . The CFD simulations were run using ANSYS Fluent (ANSYS Inc., USA). To estimate the accuracy of the result of the CFD simulation, the computed velocity sectional profile is compared to the analytic form of the velocity profile for an incompressible viscous fluid in a pipe, given by the Poiseuille equation 4.5.

$$u(r) = 2U(1 - \frac{r^2}{R^2}), \quad (4.5)$$

where  $U$  is the mean velocity and  $R$  the radius of the pipe. In our settings  $U = 0.2$  and  $R = 1.25$ .

#### 4.4.1.6 Mesh independence

The mesh independence test was performed by reporting the maximal velocity value in three cross sections located downstream in the tube model. We choose to use the maximal velocity as it is a more sensible metric than the average velocity. Figure 4.7 shows the maximal velocity as a function of the number of cells for the tetrahedral and hexahedral meshes. The maximal velocity as given by the analytic equation was also reported. We can observe that the mesh independence -i.e. the stability of the value of interest- was reached faster using hexahedral meshes than tetrahedral meshes, for a more accurate velocity value.

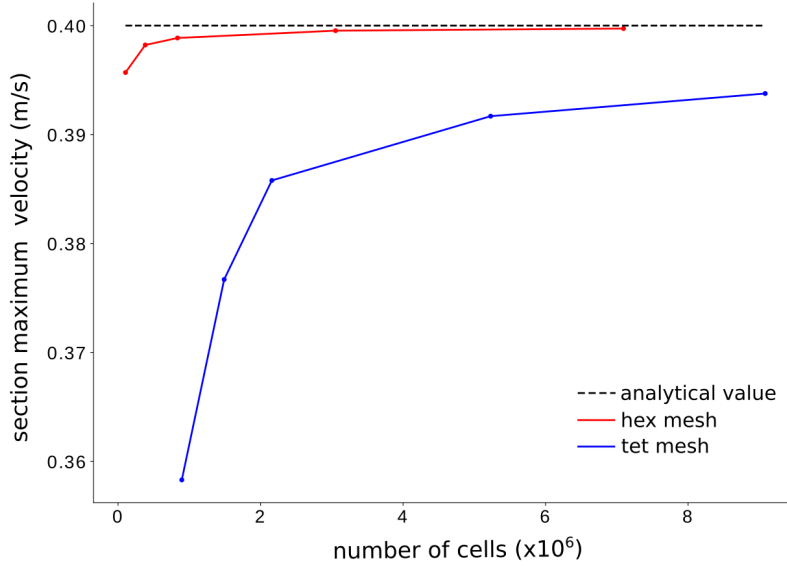


Figure 4.7: Graph of the sectional maximal velocity as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. The maximal velocity was averaged on three cross-sections along the tube model. The analytical value expected is shown by the black dotted line.

Figures 4.8 and 4.9 provide respectively the number of iterations to reach convergence and the simulation time as a function of the number of cells for the different meshes included in this study. The minimal value of residuals to consider the convergence was set to  $10^{-6}$  in this experiment. For some of the tetrahedral meshes (tet3, tet4 and tet5), this convergence criterion was never matched, the residuals remaining stable around  $10^{-5}$ , as shown in Table 4.3. In this case, the simulation was stopped when the lowest residuals were first observed, and the iteration number and simulation time were reported. This convergence issue confirmed that hexahedral meshes enable not only faster but better convergence than tetrahedral meshes.

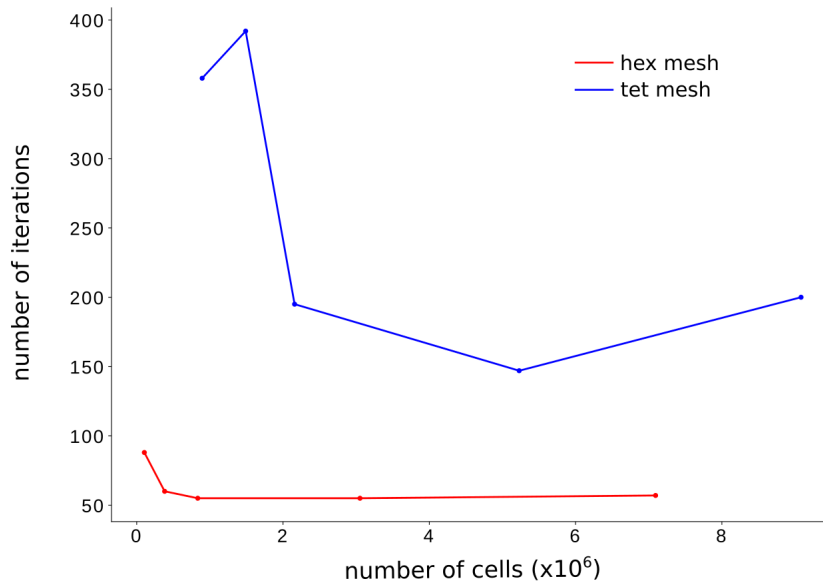


Figure 4.8: Number of iterations until convergence as a function of the number of cells for tetrahedral and hexahedral meshes.

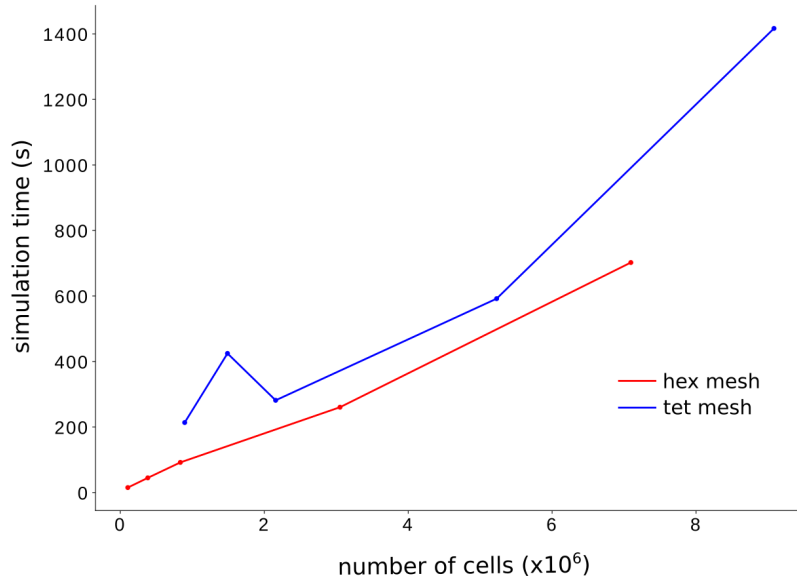


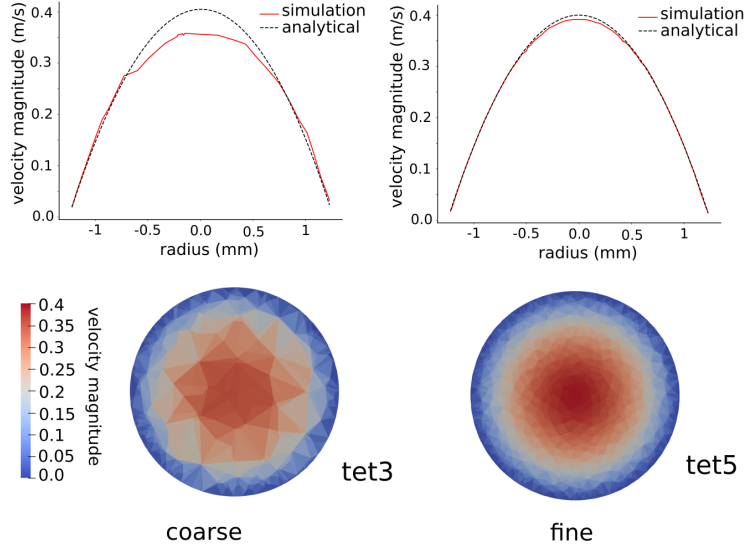
Figure 4.9: Simulation time as a function of the number of cells for tetrahedral and hexahedral meshes.

Besides, Figure 4.8 shows fewer iterations were necessary to obtain convergence of the results with hexahedral meshes. The simulation time was also reduced, which adds to the fact that fewer cells are required to reach accurate results with hexahedral meshes (as shown in Figure 4.7), reducing the computational cost even more. These results are consistent with the conclusions given in the works of [85], [18], and [25], demonstrating the advantages of hexahedral meshes over tetrahedral meshes for CFD simulations.

#### 4.4.1.7 Simulation accuracy

In Figure 4.10, we provide some visual results of the cross-sectional velocity fields. The velocity profile computed by CFD is compared to the analytical profile given by Equation 4.5. From Figure 4.10, we observe that the 2D velocity fields appear smoother for the hexahedral meshes -even the coarse mesh- than for the tetrahedral meshes. Besides, the velocity profile matches the analytical solution better for hexahedral meshes. The use of tetrahedral meshes resulted in an underestimation of the velocity magnitude at the vessel center. A coarse hexahedral mesh resulted in a more accurate solution than a fine tetrahedral mesh.

(a) tetrahedral



(b) hexahedral

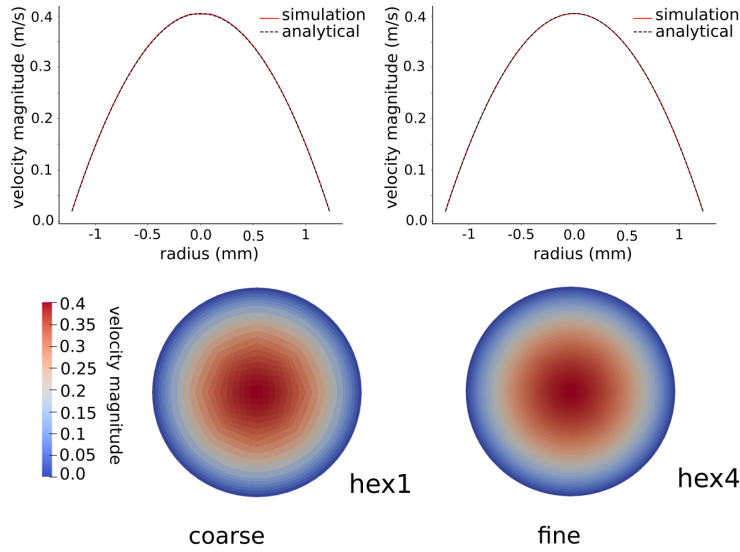


Figure 4.10: Velocity field and velocity profile for a cross-section of the tube model for coarse and fine tetrahedral (a) and hexahedral meshes (b). The analytical velocity profile is plotted alongside the simulation velocity profile. The meshes labeled coarse and fine were selected so that the number of cells is similar between hexahedral and tetrahedral meshes, to facilitate the comparison.

The numerical error of the numerical solution  $\hat{u}$  against the analytical solution  $u$  was evaluated quantitatively with two metrics; the average relative error  $RE_{mean}$ , as defined



by Equations 4.6 and 4.7

$$RE_{mean} = \frac{1}{n} \sum_{r=1}^n \left( \frac{u_r - \hat{u}_r}{u_r} \right), \quad (4.6)$$

as well as the relative error to the peak velocity  $RE_{max}$  reached for  $r = 0$ .

$$RE_{max} = \frac{u_0 - \hat{u}_0}{u_0} \quad (4.7)$$

The  $RE_{mean}$  and  $RE_{max}$  error values for the different types of meshes are reported in Figure 4.11. From this Figure, we can conclude that CFD simulation in hexahedral meshes led to more accurate results than in tetrahedral meshes.

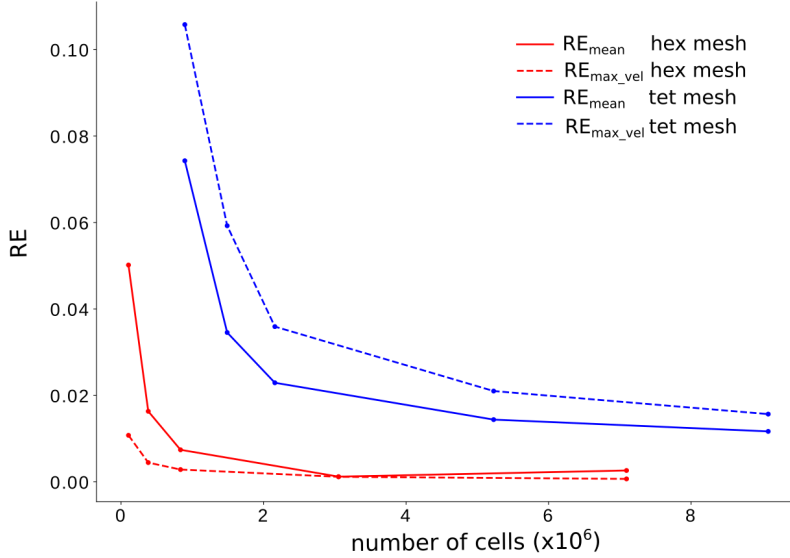


Figure 4.11:  $RE_{mean}$  and  $RE_{max}$  relative error values as a function of the number of cells for tetrahedral and hexahedral meshes.

#### 4.4.2 Bifurcation model

In this section, we reproduced the tube experiment presented before in a more realistic model of vascular bifurcation.

##### 4.4.2.1 Geometry

The patient-specific geometry of the middle carotid artery (MCA) bifurcation was extracted from the BraVa centerlines for one of the patients. A fine surface mesh was

produced, and flow extensions were automatically added using our meshing method. The mesh is shown in Figure 4.12.

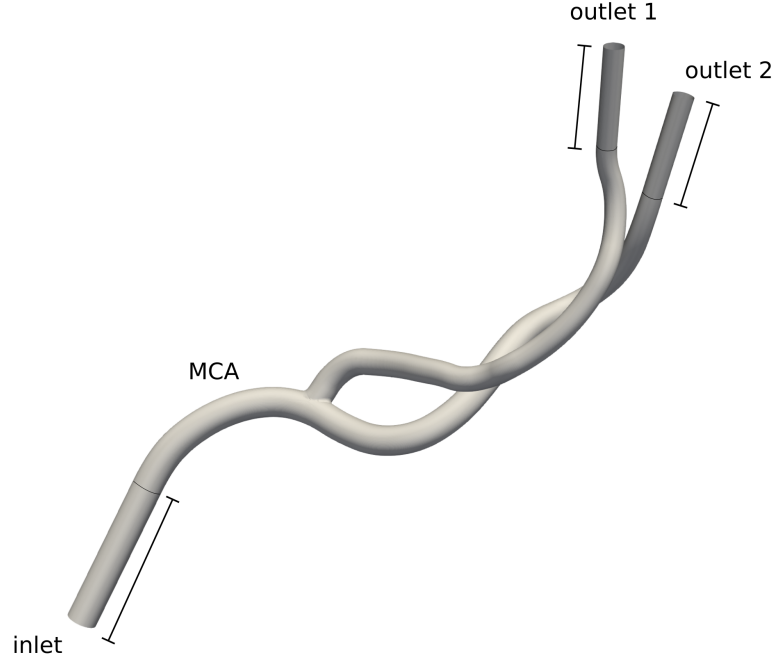


Figure 4.12: The bifurcation geometry used for the CFD simulations, with flow extensions.

#### 4.4.2.2 Meshing

Tetrahedral and hexahedral volume meshes were then created from this surface by the same method as described in the tube model in the previous section. The properties of the generated meshes are given in Table 4.4.

Table 4.4: Table of the properties of the tetrahedral and hexahedral mesh generated.

id	type	cells (#)	$SJ_{min}$	$SJ_{mean}$	$SJ_{max}$	final residuals
tet1	tet	455732	0.015	0.381	0.989	$10^{-6}$
tet2	tet	693877	0.007	0.409	0.986	$10^{-5}$
tet3	tet	932456	0.007	0.433	0.997	$10^{-5}$
tet4	tet	2093564	0.007	0.519	0.998	$10^{-6}$
tet5	tet	3469647	0.007	0.519	0.998	$10^{-5}$
hex1	hex	84736	0.479	0.955	0.995	$10^{-6}$
hex2	hex	308880	0.513	0.966	0.998	$10^{-6}$
hex3	hex	671744	0.513	0.967	0.998	$10^{-6}$
hex4	hex	2453568	0.531	0.972	0.999	$10^{-6}$
hex5	hex	5696768	0.536	0.973	0.999	$10^{-6}$

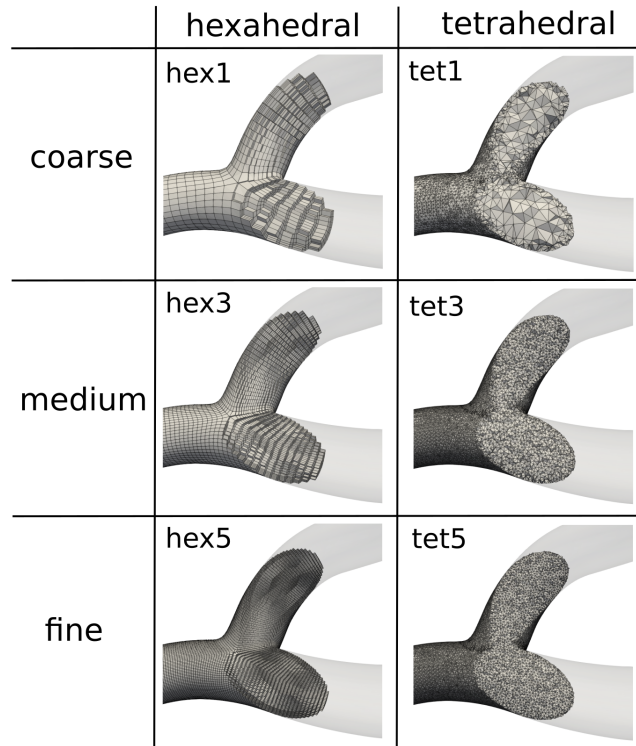


Figure 4.13: Meshes produced for the CFD study of the bifurcation model. Tetrahedral and hexahedral meshes of different densities are represented. The mesh id as given in Table 4.4 is written under each image. The meshes were cut using an oblique plane to reveal the inside cells.

As shown in Table 4.4, the mesh quality of tetrahedral elements is similar to the previous experiment (mean of 0.452). The presence of a bifurcation causes a drop in the quality of the cells in the hexahedral meshes ( $> 0.48$ ), but the average quality remains very high (mean of 0.967). Figure 4.14 shows the meshing time for the different meshes.

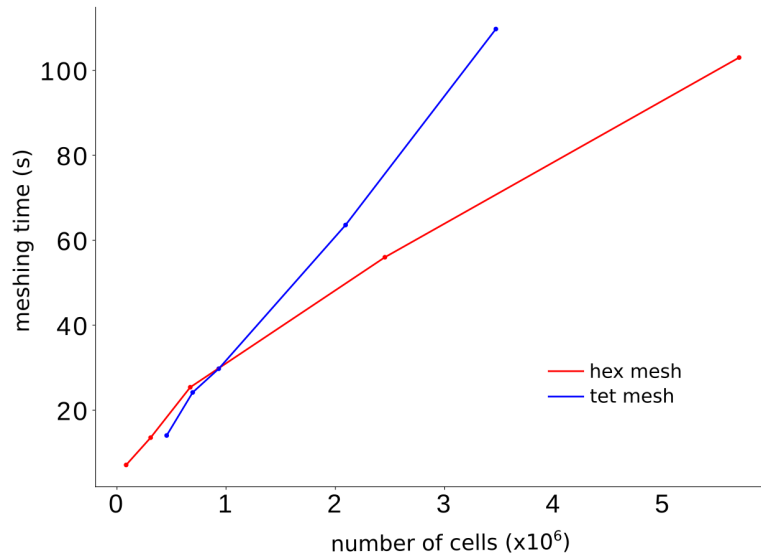


Figure 4.14: Meshing time as a function of the number of cells for tetrahedral and hexahedral meshes.

#### 4.4.2.3 Mesh independence

The CFD simulations were run with the same parameters as described previously. Figures 4.15 and 4.16 provide respectively the number of iterations to reach convergence and the simulation time as a function of the number of cells for the different meshes. The results are consistent with the results of the tube experiment. The hexahedral meshes show a faster convergence with fewer iterations than tetrahedral cells

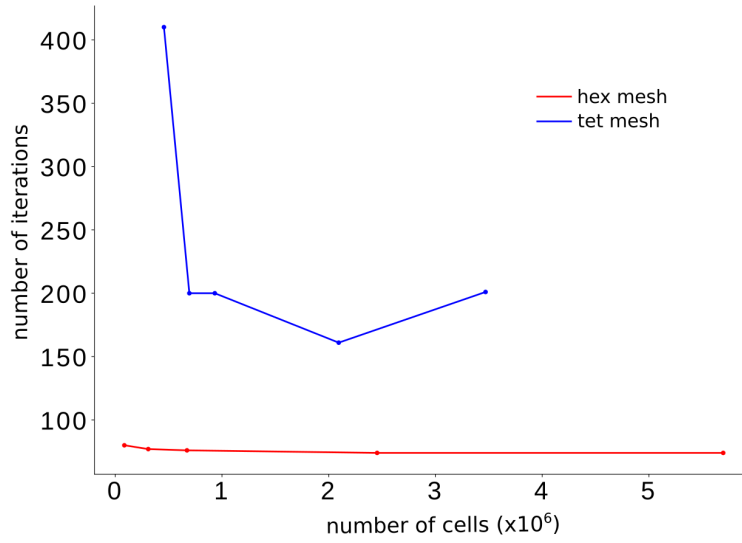


Figure 4.15: Number of iterations until convergence as a function of the number of cells for tetrahedral and hexahedral meshes.

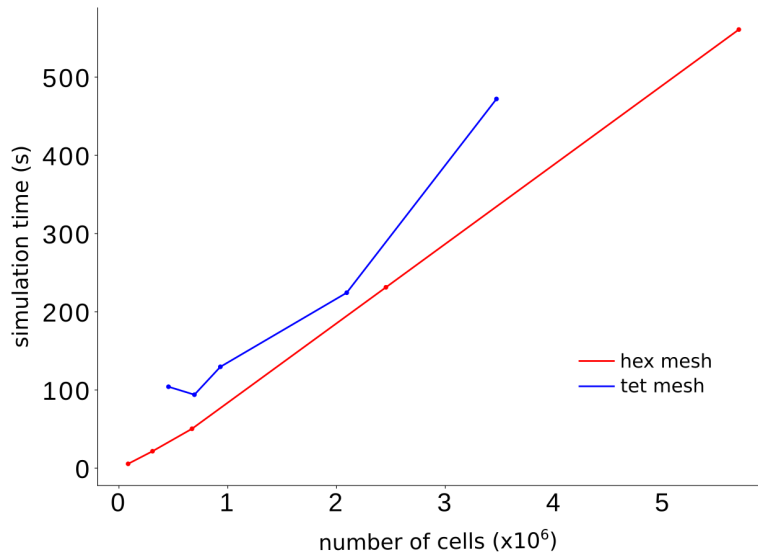


Figure 4.16: Simulation time as a function of the number of cells for tetrahedral and hexahedral meshes.

Figure 4.17 and 4.19 shows the results of the mesh independence experiment. In Figure 4.17, the results are consistent with the conclusion of the tube experiment, showing a faster convergence for the hexahedral meshes, which reach mesh independence with fewer cells.

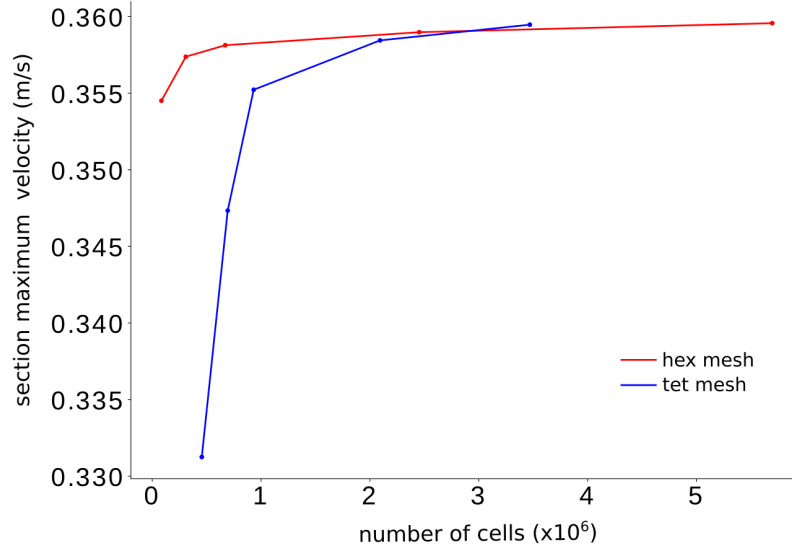


Figure 4.17: Graph of the sectional maximal velocity as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. The maximal velocity was averaged on cross-sections uniformly sampled along the vascular network.

#### 4.4.2.4 Wall shear stress

The wall shear stress (WSS) is a measure frequently used in CFD studies of hemodynamics. The bifurcation model offers the possibility to compare the mean wall shear stress for the different types of mesh. The results of the CFD simulation were post-treated automatically to keep only the surface cells and remove the flow extensions. The wall shear stress field for hexahedral and tetrahedral mesh is represented in Figure 4.18. The mean wall shear stress is reported in Figure 4.19.

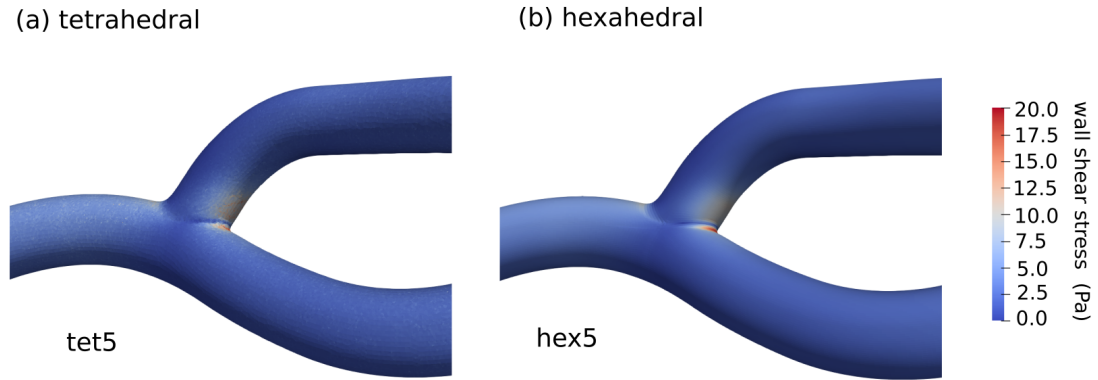


Figure 4.18: Wall shear stress values computed by CFD for hexahedral and tetrahedral meshes.

Figure 4.19 shows that the hexahedral meshes offer a more stable value of wall shear stress compared to tetrahedral meshes. Besides, the mean wall shear stress value reached with hexahedral meshes is higher than the value reached by tetrahedral meshes. If we can not state the accuracy of one value or the other by lack of ground-truth reference, this observation is consistent with the conclusions of previous works and shows the importance of the mesh for near-wall measurements. Moreover, the study of Lewandowska et al. [47] in a carotid bifurcation model demonstrated that the use of an unstructured tetrahedral mesh with prism inflation layer led to an overestimation of the WSS compared to real-life CT examination results. This study suggests that the hexahedral structured mesh gave the most accurate results both in terms of WSS and flow patterns.

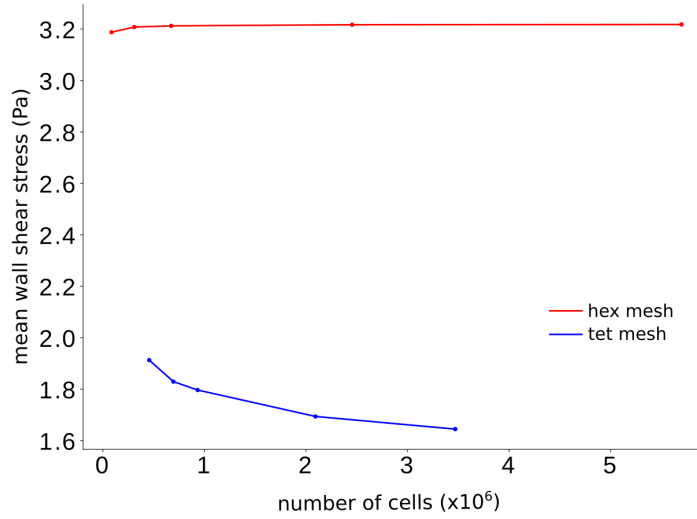


Figure 4.19: Graph of the mean wall shear stress as a function of the number of cells in the mesh for both tetrahedral meshes and hexahedral meshes. The wall shear stress was averaged on the surface of the mesh, excluding the flow extensions.

## 4.5 Conclusion

In this chapter, we compare the centerline-based meshing method proposed in our previous work to both centerline-based and segmentation-based meshing methods of the state-of-the-art. The goal is to evaluate the performance of our method and identify its advantages over existing approaches. We conducted two experiments to compare the methods. The first experiment evaluated the validity of the surface mesh produced by both methods. The second experiment compared the performance of tetrahedral and hexahedral volume meshing in terms of CFD simulation cost and accuracy. The experiments led us to the following conclusions:

In this chapter, we compared the centerline-based meshing method proposed to both centerline-based and segmentation-based meshing methods of the state-of-the-art. Two experiments were conducted, the first one to evaluate the validity of the surface mesh produced by both methods, and the second one to compare the performance of tetrahedron and hexahedron-based volume meshing in terms of CFD simulation cost and accuracy. Those experiments led us to the following conclusions:

1. The state-of-the-art deep learning-based segmentation, as well as the expert segmen-



tation resulted in topological and geometrical inaccuracies in the final mesh (bumps, disconnected or merging vessels, cycles). This led us to conclude that these problems are not only caused by the segmentation method but also by the nature of the meshing process itself, as it relies on the segmentation of low-resolution images.

2. With our method, we were able to create a mesh from the centerlines extracted from the segmentation itself. In this way, centerline-based methods can be used not as a substitute but also as a complement to segmentation.
3. The anatomical a priori of our model allowed us to correct some of the problems encountered in segmentation-based meshes (bumps, disconnected vessels, cycles) to produce meshes that are smoother, more anatomically realistic, and therefore more suitable for CFD.
4. The hexahedral volume mesh produced with the proposed method enables reducing the computation time, and improving the convergence and the accuracy of the CFD simulations compared to the commonly used tetrahedral mesh. The mesh type was shown to be particularly critical for near-wall measurements like wall shear stress.

Overall, those results demonstrate the ability of our algorithm to produce high-quality meshes not only from manually extracted centerlines but also to integrate fully-automated pipelines. It shows that some of the current shortcomings of segmentation-based mesh can be addressed by our centerline-based method, and that hexahedral meshes offer many advantages for the CFD simulations compared to tetrahedral meshes. Our comparison study, however, did not include hybrid meshes with a hexahedral/prismatic boundary layer added on a tetrahedral-based mesh, although this type of mesh is often used in the literature and is thought to lead to better results than fully tetrahedral meshes [85]. The evaluation of this type of mesh would be future work. Besides, the study was realized on a tubular surface, which is advantageous for the creation of a hexahedral mesh with high-quality cells of similar size. A model with non-circular sections would have led to a deformation of the hexahedral cells which may have impacted the results of the study. Finally, we would like to emphasize that the centerline-based methods offer more editing flexibility than segmentation-based methods, where the editing must be done manually by modifying the final mesh. Centerline-based methods allow for a faster and more efficient

manual correction of the defects commonly observed, either by automatic algorithms relying on topological properties encoded in the centerlines, or manually by editing only a few data points. Therefore, we believe that allowing the users to modify the centerlines via a user-friendly visual interface would enable to take advantage of this editing flexibility. For this reason, we developed a user interface that complements the proposed framework, presented in the next chapter.

---

# Editing framework

## 5.1 Introduction

Centerlines greatly facilitate the post-processing of vascular 3D models due to their ability to represent vascular networks with a low number of data points and encode network topology. Our centerline-based meshing framework takes advantage of these characteristics and offers new editing possibilities. To further simplify post-processing, we developed a user interface with a 3D viewer allowing for easy visualization and user interaction. This software is also a way to open the use of the meshing method developed in this work to non-expert users, including medical doctors. This software includes visualization, centerline edition, modeling and hexahedral meshing functionalities within a single framework. Unlike existing software, it allows the user to manually interact and correct the algorithm outputs at every step of the modeling based on the information contained in medical images.

## 5.2 State-of-the-art

In this section, we present the available softwares dedicated to the visualization and edition of vascular networks. Two types of methods are commonly used to generate patient-specific meshes of vascular networks: the segmentation of medical images and the centerline extraction. There was a renewed interest in the centerline-based representation of vascular networks, either as an additional contribution to the segmentation or meshing, or as a self-reliant way to model vascular networks. As opposed to image segmentation,

centerlines advantageously incorporate topological information and facilitate manual extraction and manipulation with its reduced number of parameters. It opens the way to numerical experiments in always larger and more complex networks. This interest is attested by the release of new softwares focusing on the visualization of vascular networks from centerlines in recent years.

Longair et al. developed the ImageJ plugin Simple Neurite Tracer [52]. The application was originally designed for the semi-automated tracing of neurons but can be applied to other tubular structures. The user places centerline points on the axial and sagittal slices of a 3D image. The path between points and the radius are automatically computed. The VesselVio application [11] enables the automatic extraction of centerlines from pre-binarized medical images and the computation of various quantitative features of the vascular network. Both applications focus on the manual or automatic extraction of centerlines and their visualization, which corresponds to a step before the application we propose. The edition and the generation of a CFD mesh is not the purpose of these works. Besides, Abdellah et al. proposed the VessMorphoVis Blender plugin [1], whose purpose is close to ours. This application enables to visualize, analyze and automatically repair the vascular skeletons. A basic polyline mesh as well as a more realistic polygonal mesh based on metaballs can be generated automatically from the centerlines. This useful plugin differs from our application in several ways. The interface is not intended for centerline editing based on the medical images and the proposed meshing method is mainly dedicated to visualization.

None of these applications allow to produce the mesh discretization of vascular networks required for CFD. This task is commonly performed with general softwares able to discretize any surface mesh with tetrahedral elements [23]. However, discretization methods specifically designed for vascular networks can produce more adapted and efficient meshes, with flow-oriented, high-quality hexahedral elements. Few softwares are dedicated to the volume meshing of the vasculature. The VMTK software [38] performs centerline extraction from a surface mesh, and uses the centerline and surface to generate a tetrahedral volume mesh with a hexahedral boundary layer. The interface PyFormex [18] is dedicated to the semi-automatic hexahedral re-meshing of a surface mesh with a single vascular bifurcation. In both cases, the input required is a surface mesh, which is challenging to obtain for large vascular networks. Our application aims at filling the gap

between the visualization and meshing softwares.

## 5.3 Software overview

### 5.3.1 Input data

The input of the software is the centerlines of the vascular network to edit and mesh. The input formats supported include the SWC format proposed by Cannon et al. [12] and VTK polyLines structure as used by the centerline extraction software VMTK ([www.vmtk.org](http://www.vmtk.org)) [38]. These formats are the most commonly used in the literature and they allow reading data from the large open databases of vascular centerlines Aneurisk [3] - in VMTK format - and BraVa [89] - in SWC format - for instance. In addition, we created a new input format dedicated to the creation of idealized vascular networks with straight vessels and uniform radii. This type of vascular model is used in many studies investigating the impact of vascular geometry on blood flow [7, 9]. In a text file, each line contains a list of six values separated by a space, representing one branch. These values are respectively the id, the branch length (mm), the branch radius (mm), the number of data points, the angle formed with the previous branch (*degree*), and the id of the previous branch.

### 5.3.2 Main functionalities

The proposed software enables the user to visualize and edit the centerline manually based on the medical image. The main geometric and topological features (e.g. branch angle, branch configuration) of the network can be modified effortlessly. The modeling and meshing methods presented in the previous chapters can be run directly from the software, and their output can be visualized and edited. Four different representation modes (data, topology, model and mesh) corresponding to different steps toward the final mesh, can be displayed. These representations are detailed in the following sections and illustrated in Figure 5.1, as well as the functionalities mentioned in this overview.

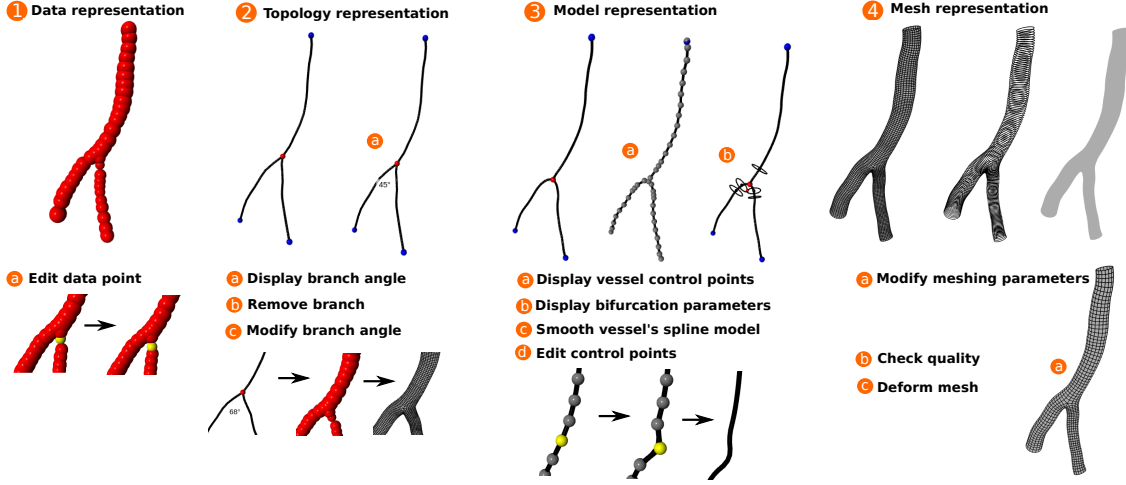


Figure 5.1: Overview of functionalities of the software. The numbers 1 to 4 correspond to the different representation modes of the interface (data, topology, model, mesh). For each mode, the related functionalities are listed and illustrated with displays of the software. For space reasons, all the functionalities could not be illustrated; please refer to Figure 5.7 for more images.

### 5.3.3 Interface layout

The interface was built using the 3D programming python package vpython ([www.vpython.org](http://www.vpython.org)). It was chosen over other well-known softwares optimized to handle large meshes, such as Blender ([www.blender.org](http://www.blender.org)) or Paraview ([www.paraview.org](http://www.paraview.org)), for its simplicity of use and user-interaction-oriented functions. It is more suited for manual editing than visualization-oriented tools such as Paraview or pyvista ([www.pyvista.org](http://www.pyvista.org)), while providing a more simple, user-friendly layout than Blender. As illustrated in Figure 5.2, the designed interface integrates the following components:

- The 3D viewer is dedicated to the display of the objects to visualize, here the vascular networks. It handles both the basic user interactions such as rotation and zoom and the original edition functions developed in this work.
- Important information, software handling guidance, and algorithm output messages are displayed in the message bar.
- The edit menu activates the edition mode for one of the four vascular network representations: data, topology, model or mesh.

- The image import and management bar allow the user to import a medical image and visualize it in the 3D view through the dedicated widgets.
- An new input centerline can be imported through the import field.
- The export field allows the edited networks to be saved in various formats.
- The four visualization panes can be activated independently by checking the dedicated box. They control the display of the different vascular network representation modes (data, topology, model, and mesh) on the 3D viewer. The representation modes are detailed in the following sections. If activated simultaneously, the representations are superimposed on the viewer. Each pane gathers several widgets which control the functionalities associated with a given representation mode.

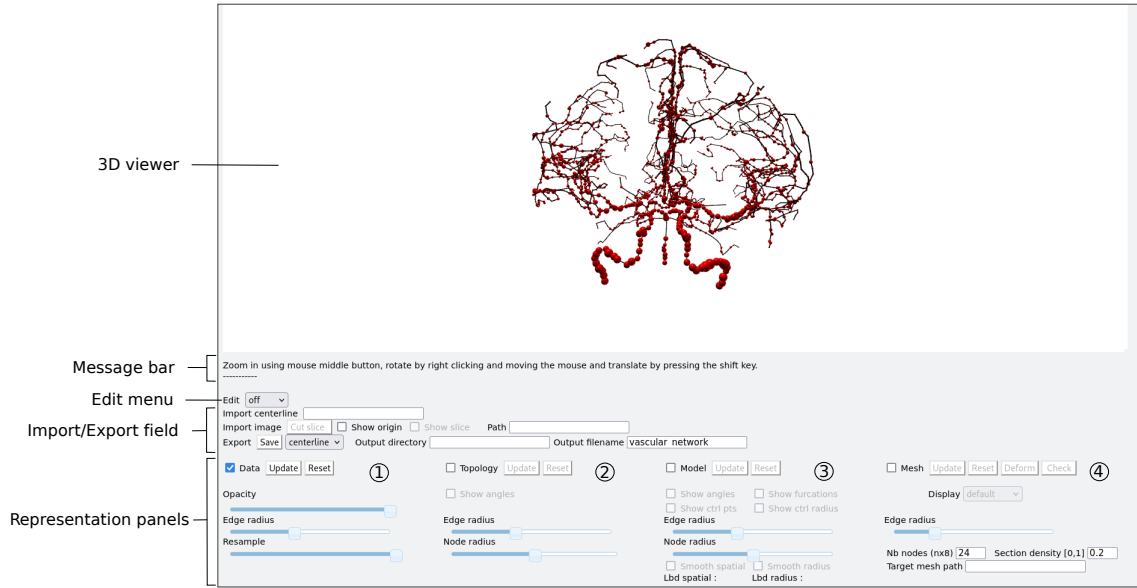


Figure 5.2: The interface layout. The numbers 1 to 4 correspond to the different representation modes of the interface (data, topology, model, mesh).

### 5.3.4 Spatial graph encoding

Graph-based representations are well-suited to encode and analyze vascular networks. Whereas this type of structure provides interesting tools to study vascular connectivity, [6] demonstrated the importance of encoding not only the topology but also the 3D geometry of the network in a spatial graph. Following this idea, we represent a vascular network

with four spatial graphs. A *data graph* gathers the information of the input centerline; data points are represented by nodes with 3D coordinates and radius as attributes. They are connected with edges oriented in the blood flow direction. A second graph, called *topological graph* integrates both geometrical and topological information. The particular data points (furcations, inlet, or outlet) only are set as nodes and connected by directed edges. The data points of the paths connecting those nodes are stored as edge attribute. The parameters of the vessel and bifurcation models are gathered in a *model graph*. Finally, the position of the cross-sections and the nodes of the mesh are stored in the *crsec graph*. The graphs and their attributes are represented in Figure 5.3. This storage method facilitates data manipulation and editing as it allows the use of graph theory-based algorithms such as depth-first search or neighbor identification. The meshing of cycles (e.g. the circle of Willis) was made possible by the addition of sink nodes -nodes with no out edges- in the graph.

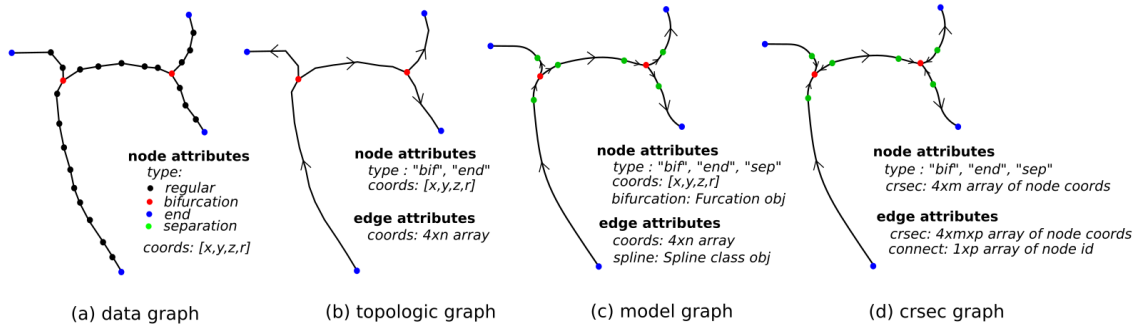


Figure 5.3: The 4 graphs used to encode the vascular network, with the edge and node attributes.

## 5.4 Editing functionalities

### 5.4.1 Centerlines

#### 5.4.1.1 Visualization

The input centerline data before treatment can be displayed under two representation modes; the data mode and topology mode. In data mode, each centerline data point  $(x, y, z, r)$  is represented on the 3D view by a red sphere of center  $(x, y, z)$  and radius  $r$ . The spheres are connected with black lines, according to the topology of the network.



This mode enables to visualize the information provided by the input centerlines in a single representation. However, the underlying topology of the vascular network may be hidden by the data point spheres, especially for centerlines with high sampling. To facilitate the visualization, another representation called topology mode is proposed.

In topology mode, only the key data points (furcation, inlet, and outlet) are displayed. They are represented with spheres of uniform radius. A single color is associated with each type of key point. The connecting vessels are represented by black polylines. The opacity and the size of the spheres and polylines can be adjusted using the dedicated sliders, allowing different representations to be overlapped easily. The data and topology representation modes are illustrated in columns 1 and 2 of Figure 5.1.

#### **5.4.1.2 Data point editing**

The centerline data points' position and radius can be edited directly from the 3D view. To prevent accidental modification of the centerlines, the editing mode must be activated beforehand. Centerline editing is performed from the data representation mode. The data points can be individually selected, dragged to a new position or removed (Figure 5.1 image 1.a). The radius of the data point can also be adjusted after selection by keyboard control. New data points or new branches can also be manually created. The topology of the network can be modified by removing or creating new edges between data points. Finally, a slider allows for data resampling, to facilitate the visualization of centerlines with high density of data points. We acknowledge that it can be hard to visualize and manipulate large networks. For this reason, we implemented a function allowing the user to select the vessels of interest and crop the network accordingly, like in Figure 5.7. It is therefore possible to focus only on a small part of the network and easily go back to the visualization of the entire network.

#### **5.4.1.3 Branch removal**

The topology of the vascular network differs between individuals. For instance, the circle of Willis has various configurations, which impacts the hemodynamics [2]. To facilitate the study of the influence of the network topology, a trimming functionality is included in our software. To preserve the connectivity of the network, all the downstream vessels arising from the branch to remove are also suppressed. The identification of downstream

vessels is performed efficiently using a depth-first search in the graph associated with the topology representation mode.

#### 5.4.1.4 Branch angle

There are many studies investigating the impact of the angle formed by two branches on blood flow [7, 68, 75]. The creation of the meshes in those studies mostly relies on manual design using 3D modeling or CAD softwares. In this work, we propose to simplify this task by allowing the user to modify effortlessly the branch angles of the vascular network. Additionally, the branching angles can be computed and displayed on the 3D view. In topology mode, it is computed as the angle formed by the vectors connecting the bifurcation point to a point located one diameter downstream in each daughter branch. The angle obtained by this method may be affected by the quality of the initial centerlines. Angles can be retrieved more accurately from the model representation, in which case the computation method differs. Images 2.a and 2.c of Figure 5.1 give an example of angle display and modification.

### 5.4.2 Model

#### 5.4.2.1 Visualization

The vessel and bifurcation models obtained by the method described in Chapter 2 can be displayed via the model representation mode. By default, the bifurcation barycenter as well as the inlet and outlet points are represented by spheres of different colors, and the vessels' spline model by black curves. The control points of the splines and the cross sections composing the bifurcation models can optionally be displayed on the viewer. The different model representations are illustrated in column 3 of Figure 5.1.

#### 5.4.2.2 Vessel trajectory edition

The surface of the mesh produced by our framework is defined entirely by this parametric model. For this reason, we give the possibility for the user to directly edit the model through the interface. It is an alternative to the editing of input data points that is less time-consuming and gives more direct control of the final surface. The control points are represented on the 3D viewer by spheres that can be modified (position and radius) in

the same way any regular data point can be moved. As the control points are modified, the spline trajectory is updated, as shown in the image (3)(d) of Figure 5.1. Besides, the smoothing parameters of the splines used to model vessels ( $\lambda_s$  and  $\lambda_r$ ) can be manually modified to control the vessel's smoothness. Unlike control point editing, smoothing affects all the control points of a spline simultaneously, modifying a vessel segment globally. The impact of the changes on the curve trajectory and control points is displayed in real-time, as well as the current parameter value.

### 5.4.3 Mesh

#### 5.4.3.1 Visualization

The structured hexahedral meshing method described in Chapter 3 can be run directly from the software. On the 3D view, four modes are available to display the resulting mesh, as shown in column 4 of Figure 5.1; the *default* mode (surface and edges), *solid* mode (surface only), *wireframe* (edges only) *section* (cross sections only). The framework used for the creation of the interface, vpython, was selected as a compromise between efficiency and simplicity of use. It is not optimized for the display and manipulation of large meshes as other software such as Paraview or Blender can be. This can cause the interface to become slow or unresponsive when working with a large vascular network in mesh representation. In our experience, only meshes with less than 50k faces can be displayed smoothly. To circumvent this limitation, we applied the same idea as for the visualization of the 3D image: to show only what is needed. Through an edge selection menu, the mesh can be displayed one part at a time, following the current visualization needs.

#### 5.4.3.2 Quality check

High curvature or data point misplacement can cause low-quality cells in the output mesh, hindering the use of the mesh for numerical simulation purposes (cf. Chapter 3). In such a case, local editing of the centerline is required. To facilitate the localization of the meshing failures, we implemented a function that highlights the low-quality area by adding a red coloration on the surface mesh, as shown in Figure 5.7. We used the positivity of the cells' scaled Jacobian as a quality metric. The surface map is thresholded

to color the invalid faces in red. As those faces can be very small or hidden, the entire section is colored if the invalid cell is located in a vessel and all the faces if it is located in a bifurcation.

## 5.5 Medical image superimposition

To ensure that the centerline network is as close as possible to the real patient anatomy, a medical image (in NIFTI format) can be imported into the viewer. As there is no registration step, the user must ensure beforehand that the image real coordinate in mm, as written in the image header, matches the initial centerline data point coordinates. The visualization of large 3D images such as magnetic resonance angiography (MRA) is challenging. To enhance the relevant information and accelerate the rendering, we propose to display local 2D slices of the image. The center of the slice is defined by a user-set origin cursor, the normal being the current scene viewpoint angle. Ray tracing is employed to extract a slice according to a given normal from the 3D image. For clarity and computational time requirements, only one slice at a time can be displayed. By computing the slides sequentially along the vasculature, the input centerline can easily be compared to the vessel position on the image, enabling to manually correct different parts of the network. The image slice can also be juxtaposed with other representation modes such as the model or the mesh. An example of medical image superimposition is provided in Figure 5.7.

## 5.6 Pathology modeling

### 5.6.1 Aneurism

The proposed model is based on the assumption that vessel cross-sections are circular, which is limiting when dealing with pathological vessels. A way to address this limitation is to deform the cross-sections to match a target surface as post-processing. If the user input data is a surface mesh, we propose the following alternative use of our meshing framework:

1. Extract the centerline from the surface mesh (using VMTK software for example).

2. Create a tubular mesh from the centerline using the proposed method.
3. Deform the tubular mesh to match the original surface.

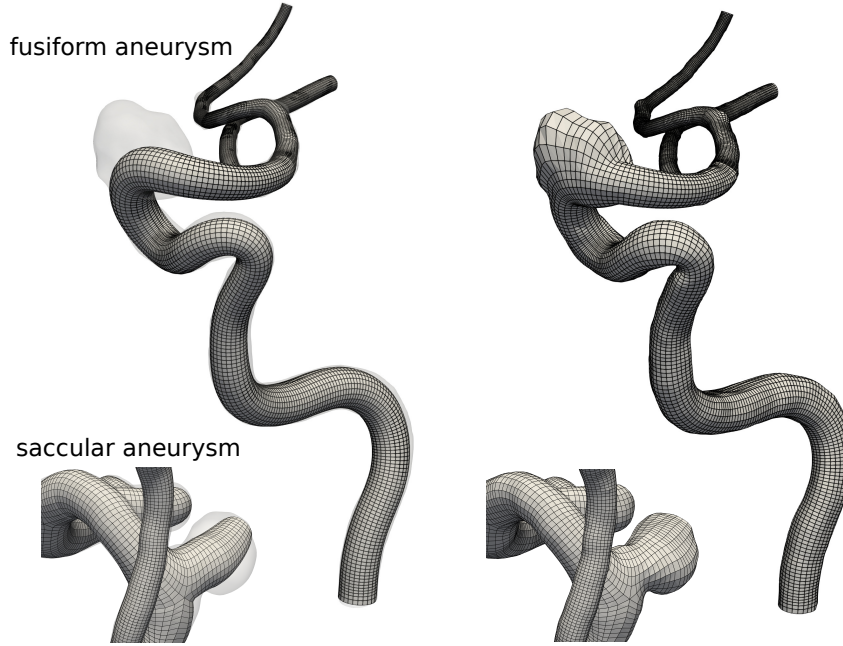


Figure 5.4: Structured hexahedral meshing of cerebral arteries with a fusiform or saccular aneurysm by deformation. On the left, the tubular mesh, obtained by our framework, is superimposed on the target surface. On the right, the mesh after projection is shown.

Figure 5.4 illustrates an example of this pipeline to mesh arteries with aneurysms. In the deformation step, the nodes are individually projected onto the surface of the target mesh. To prevent the sections from intersecting, the nodes are projected radially from the section center. As shown in Figure 5.4, saccular aneurysms are initially modeled as bifurcating vessels and then deformed. Because the shape of the O-grid pattern depends on the position of the section nodes (cf. Chapter 3), the deformation of the surface mesh is smoothly conveyed to the cells inside the mesh, as illustrated by Figure 5.5.

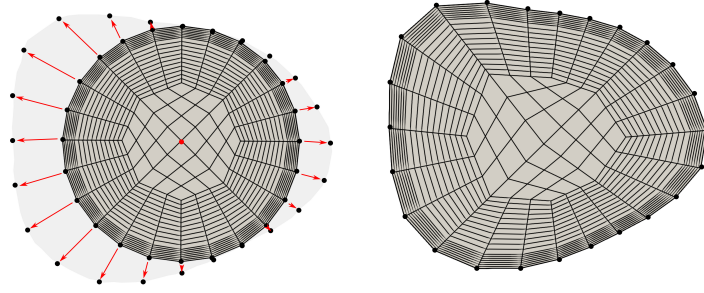


Figure 5.5: Cross-section pattern before and after deformation. The corresponding slice of the target surface mesh is represented in grey.

This pipeline is not limited to pathological vessels. It can be extended to remesh any vessel surface mesh with hexahedral cells.

### 5.6.2 Stenosis

In the case of pathologies such as stenosis, we designed a method to automatically match a user-defined template to a target part of the vasculature. With this method, the design of the stenosis shape is simplified and the same stenosis can be applied to different vessels.

A template of the stenosis shape is first designed by the user directly from the software interface. A cylinder is used as a template shape. The stenosis shape can be formed by reshaping successive cross-sections of the cylinder, as shown in Figure 5.6, Image 1. Each cross-section is represented by a polyline whose points can be moved one by one by dragging. Once the user input data is saved, a bicubic interpolation is applied to obtain a smooth 3D model from a small number of cross-sections, which reduces the editing time needed to create the model. Figure 5.6, image 2, shows the superimposition of the stenosis 3D surface obtained by interpolation and original cross-sections defined by the user. Finally, by using the curvature and radius information provided by our spline-based vessel model, the template can be matched to any vessel in the vascular network of interest. The same template can be used with different cell densities, as shown in Figure 5.6, image 3.

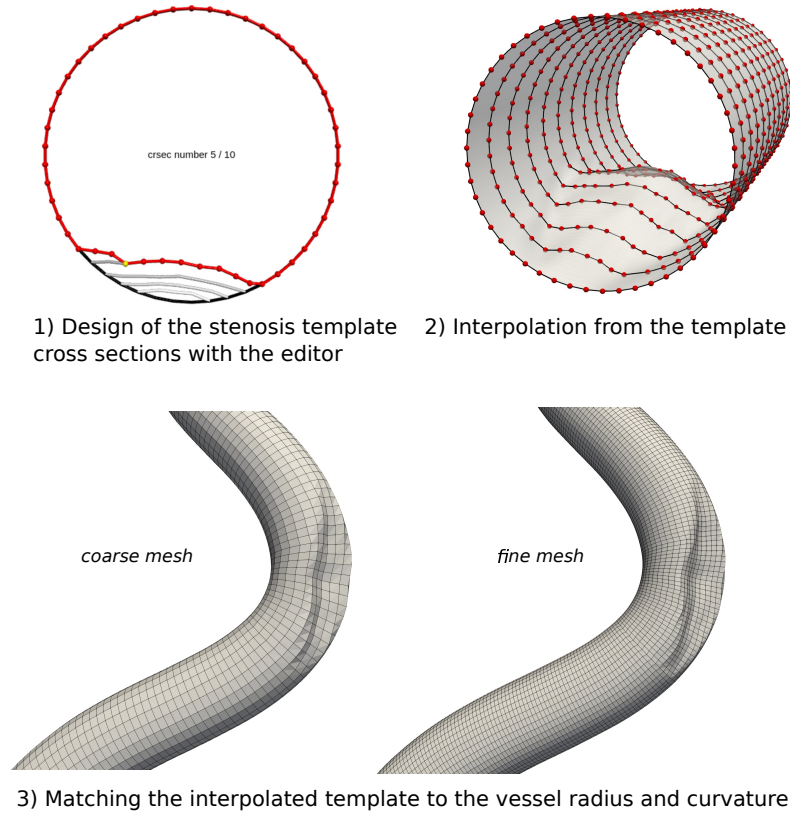


Figure 5.6: Design of a stenosis in a vascular network.

## 5.7 Applications

### 5.7.1 Open database of cerebral arterial network meshes

To our knowledge, there are no databases of large cerebral arterial network meshes available publicly. However, such a database could find many interesting applications such as conducting catheter/stent deployment simulations, studying the impact of vascular pathologies on the local blood flow but also on the brain tissue blood supply, studying phenomena such as the delay/dispersion of imaging contrast agent, or providing a ground-truth for vessel segmentation methods. Therefore, we propose to create the first open database of cerebral arterial network surface and volume meshes from the centerlines of the BraVa database using the automatic meshing method and the editing software developed in this thesis.

As discussed in Chapter 3, the main limitation to the use of the centerlines of the BraVa database for CFD is the defect caused by the manual extraction (low number of data points,

misplacement) and by the automatic algorithms (branching point misplacement, radius noise). As discussed in Chapter 3, if our meshing method was designed to be robust to centerline defects, it remains affected by the quality of the input data. In this context, the presented software finds its purpose. Using the editing functions provided, we corrected manually and locally the centerlines of some patients of the BraVa database, to create a database of ready-for-CFD, high-quality meshes of large arterial networks.

#### **5.7.1.1 Editing method**

The manipulations performed with the software to complete this task are reported and shown in Figure 5.7. The parts of the centerlines with defects are first isolated from the network by the cropping function. In case A of Figure 5.7, the misplacement of data points near the bifurcation led to the generation of invalid cells in the mesh. The low-quality parts were highlighted in red by the mesh-checking functionality provided in the software. It corroborates the defects identified by overlapping the medical image on different views. Once the visualization step (A) is achieved, the centerline data points can be corrected (B). The points were moved manually to match the medical image and their radius was adjusted consequently.



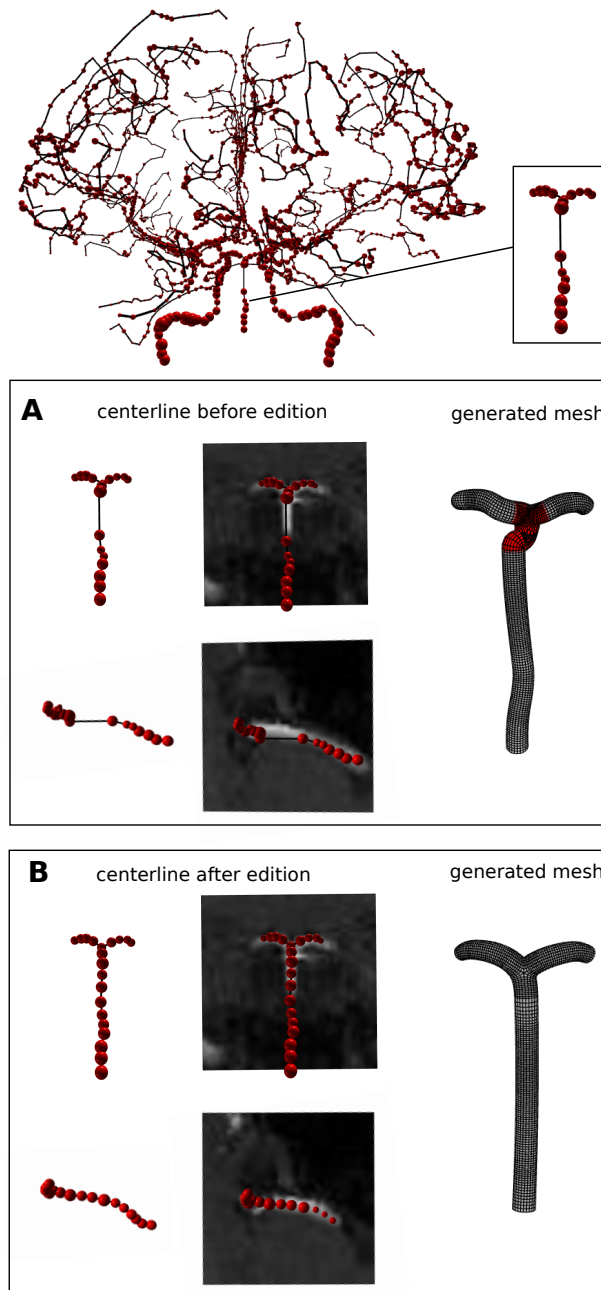
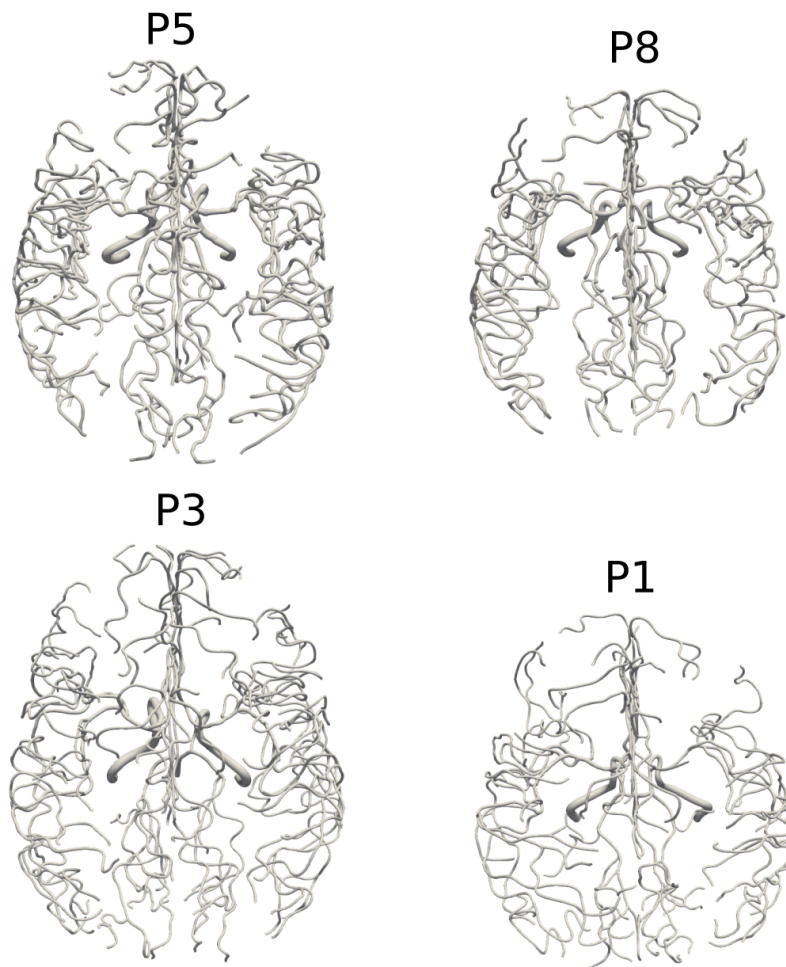


Figure 5.7: Software use case for the manual correction of a vascular network of the BraVa database. In this example, we focused on editing the basilar artery for a patient of the BraVa database. Panel A shows several views of the input centerline and the superimposition of the magnetic resonance angiography image. The mesh generated by the software for the centerline is shown on the right. The checking function was used to color in red the low-quality cells. Panel B shows the same centerline after editing the data points to match the medical image. The mesh produced from the corrected centerlines -on the right- has no low-quality cells. The images in the figure are unchanged displays provided by the software.

Overall, the editing task only took a few minutes. Panel B of Figure 5.7 shows the centerline after correction and the new mesh generated. This task was repeated in the location of mesh failures until the mesh reached the quality required for CFD simulations.

#### 5.7.1.2 Database

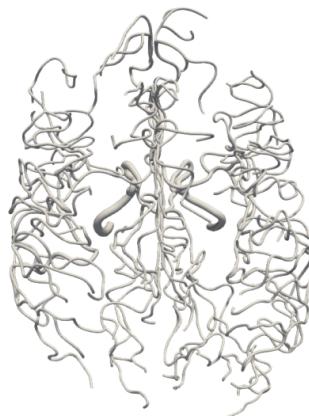
A total of 10 patients were manually corrected by the method described above and included in our database of meshes. All the produced meshes show a scaled Jacobian value  $> 0.2$  and are suited for CFD. The meshes will be further analyzed and released publicly in future work. Figure 5.8 shows the meshes included in our database.



P9



P11



P18



P17



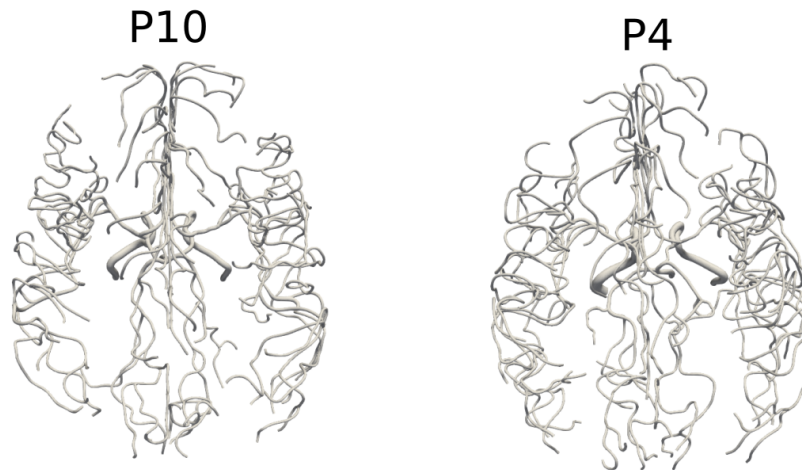


Figure 5.8: Database of 10 meshes of the large cerebral arterial network produced for the centerline of the BraVa database using the meshing method and software developed in this thesis.

### 5.7.2 3D printing

In this section, we show one of the possible applicative use of the proposed database. We fabricated a plastic model of the whole brain cerebral vascular network by 3D printing one of the meshes of the database for visualization or educational purpose. As the mesh includes very small parts and complex shapes, it could not be printed at one time. The mesh was divided into 12 parts that were rescaled by a factor 1.5 and then 3D printed individually. All the parts were then glued together and fixed to a socle, as shown in Figure 5.9.

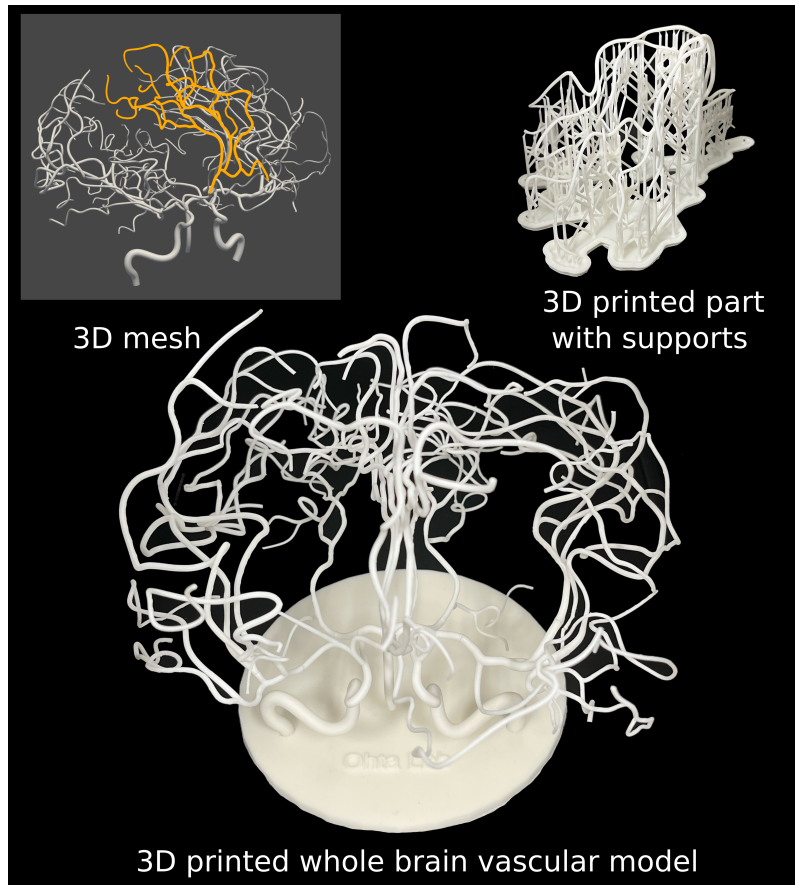


Figure 5.9: 3D printing of a whole brain arterial network model.

### 5.7.3 CFD in a stenotic vessel case

In this section, we demonstrate the applicability of the meshing method proposed for CFD simulations. We reconstructed a patient-specific mesh of the middle carotid artery (MCA) and downstream vessel in a case where the segmentation failed to produce a valid mesh. Using our framework, fluid extensions were automatically added to the inlet and outlet of the mesh and a stenosis with a user-defined shape was automatically added to the MCA. The stenosis was designed to induce a reduction of 50% of the vessel diameter. Figure 5.10 illustrates the modeling and meshing process used to reconstruct the meshes used in the CFD study presented in the main article.

Simple boundary conditions were used for this application case, as our goal is not to provide an analysis of this case study but simply to demonstrate the applicability of our method for the study of cerebrovascular pathologies by CFD. Blood is considered a

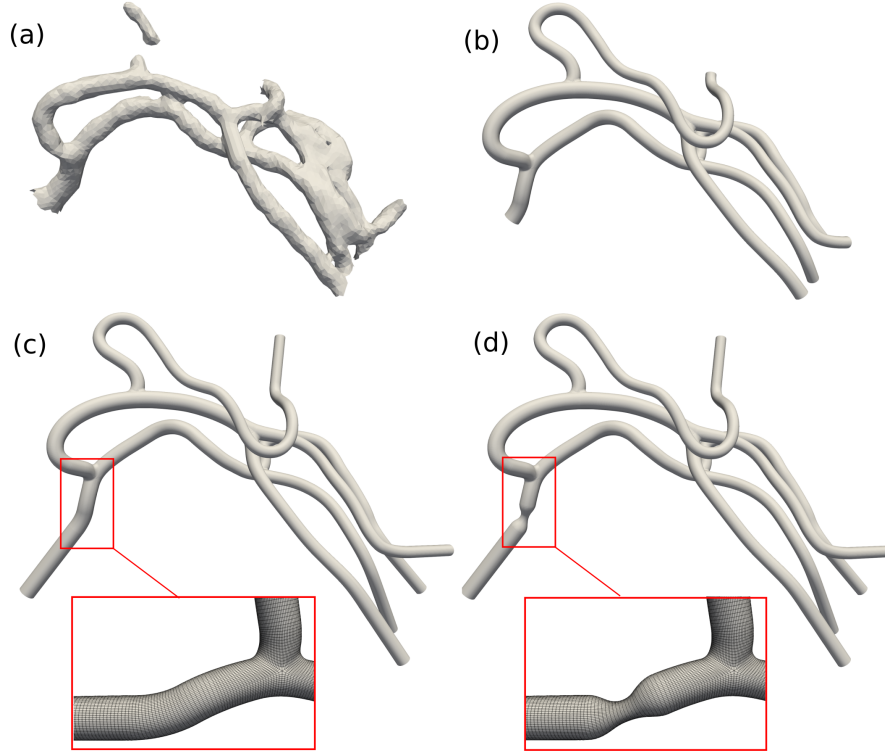


Figure 5.10: Comparison of the segmentation-based and centerline-based mesh. (a) Segmentation-based mesh produced by Unet for the arterial network of interest, i.e. the right middle cerebral artery (MCA) territory. (b) Mesh produced for the same arterial territory as (a). This mesh was reconstructed with our method from the centerlines of the database BraVa. (c) Mesh after automatically adding extensions to the inlet and outlets. (d) Mesh after automatically adding a 50% stenosis in the MCA. In (c) and (d), the hexahedral mesh created in the stenosis region is highlighted.

Newtonian fluid ( $\rho = 1053\text{kg.m}^{-3}$ ,  $\mu = 0.0035\text{kg.m}^{-1}.\text{s}^{-1}$ ), and the flow is assumed steady and laminar. The inlet boundary condition is set to a velocity of  $0.2\text{m.s}^{-1}$  [49], and the outlet is set to zero pressure. The simulation converged with a residual value of  $10^{-6}$  in 50 iterations (resp. 78 iterations) and 4 minutes (resp. 6 minutes) for the healthy case and the pathologic case respectively. The velocity streamlines and the velocity fields computed by CFD for the healthy and pathologic cases are presented in Figure 5.11. With this experiment, we showed the potential of our meshing method to easily design and conduct blood flow studies by CFD. The editing flexibility of our framework allows to study the effect of hemodynamic pathologies or topological changes compared to a reference geometry like in Figure 5.11. The advantages of our framework are not limited

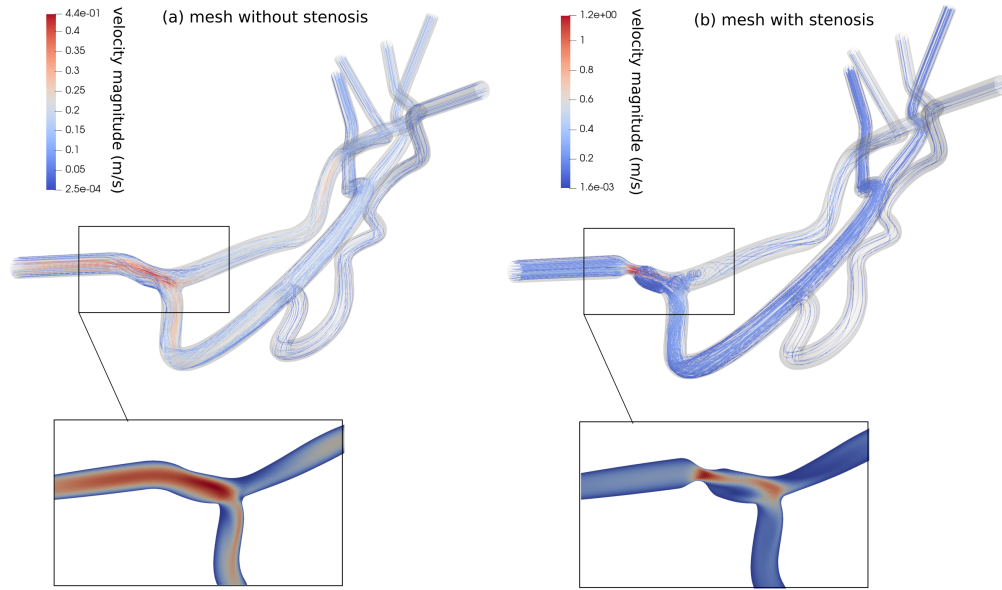


Figure 5.11: CFD simulation results for the mesh without stenosis (a) and with stenosis (b). In both cases, the velocity streamlines were rendered and the velocity field is shown by a cut on the stenosis region.

to the modeling and meshing steps, as it also facilitates the analysis of the results (e.g. extraction of cross sections, extraction of velocity values along the centerline), as shown in Figure 5.12.

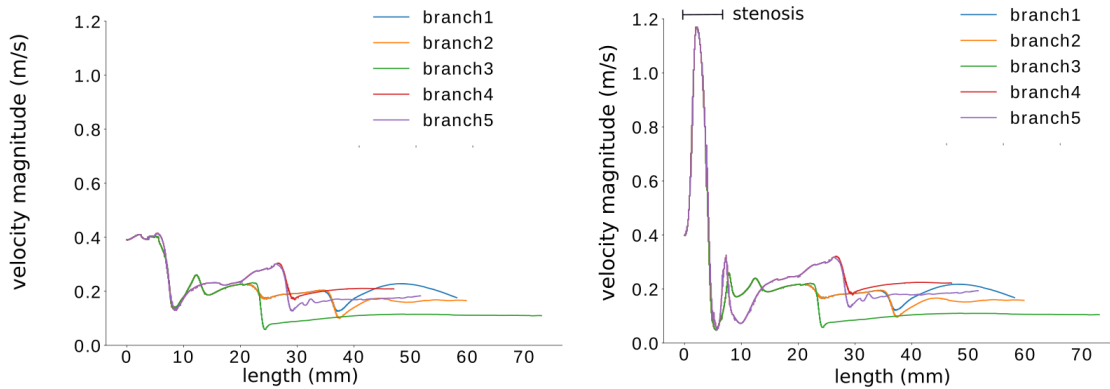


Figure 5.12: Velocity along the centerline for the different branch of the MCA territory model in the healthy case (a) and the stenosis case (b).



## 5.8 Conclusion

In the present chapter, we presented editing possibilities provided by our code. The possibilities offered include, but are not limited to, the visualization and manual modification of input centerlines, the removal of branches of arterial territories, and the modification of branch angles or vessel trajectories. The editing of vascular networks from centerlines to mesh was facilitated by a user interface that was developed during this thesis. The interface was designed so that it is simple and intuitive to use. We believe that it can be useful to researchers in various fields including medicine and biomechanics. It was already tested by a panel of students with medical, biomechanical and informatics background for different types of studies. The provided feedback led to the integration of functionalities adapted to the user needs such as the branching angle computation and display. Using this software, we created a database of large arterial cerebral vascular networks ready for CFD that we believe will be useful to the scientific community. Indeed, such a database can be used to evaluate medical devices (e.g. stent deployment), to conduct hemodynamic studies of vascular pathologies in realistic vascular networks, to provide ground-truth data for deep learning models or realistic boundary conditions for CFD simulations (e.g. using 1D models). Nevertheless, we acknowledge some limitations to this work. The interface is subject to the limitations of the automatic method used for modeling and meshing, such as difficulties with high curvature areas or mesh failures. However, it was precisely designed to overcome them by allowing the user to manually correct the output of the algorithm. Additionally, the mesh deformation function is limited, as it requires a target surface. Paradoxically, the easiest way to obtain this target surface is image segmentation, which is precisely the task we are trying to bypass in this framework. Moreover, some types of deformations can badly impact the quality and properties of the initial tubular mesh by introducing intersecting cross-sections.



---

## Conclusion

In this work, we addressed the problem of the reconstruction and meshing of large vascular networks from noisy, sparse centerlines. In Chapter 2 and Chapter 3, we propose a method to automatically create a hexahedral structured mesh suitable for CFD from centerlines alone. It opens the way to CFD simulations in large vascular networks manually or semi-automatically extracted by medical doctors, with minimal manual intervention. An original approximation method unifying the spatial and radius information in a single function is proposed to model the vessels. The use of a bifurcation model based on anatomical parameters is associated with new hexahedral meshing and smoothing techniques to produce bifurcations with a realistic shape and high-quality cells in a reasonable time. In Chapter 4, we evaluated the proposed method against the commonly used segmentation-based meshing method. The results of our comparison experiments showed that the centerline-based method proposed helps to overcome the shortcomings of segmentation-based meshes (e.g. topological and geometrical inaccuracies) and facilitates the creation of meshes for CFD, and that the hexahedral mesh enables to improve the accuracy and reduce the cost of numerical simulations, which can have an impact on clinical studies [47].

If this work is a first step towards building a complete toolkit for hemodynamics studies, it could be extended and improved in many ways. The pipeline was mainly tested on cerebral vascular networks so far, and non-planar  $n$ -furcations ( $n > 3$ ) that are common in other vessels (e.g. aorta) were not addressed. In addition, the robustness of the modeling method could be further improved as it causes the meshing step to fail in some cases. For this, we would like to integrate more constraints on the bifurcation and

vessel models such as a maximum curvature or maximum vessel angle. Besides, we want to emphasize that our objective in this work was not to improve the performance of the segmentation or centerline extraction algorithms but to acknowledge the limitations of the realistic data and generate meshes as close as possible to the real anatomy from flawed centerlines and existing databases. Hence, the accuracy of the reconstruction depends on the accuracy of the input centerlines and might not be in total accordance with the real patient anatomy as shown in medical images. The issue of the agreement between the medical image and the centerline points position, was partially addressed by developing an editing software, as presented in Chapter 5. The functionalities of this software include the different automatic methods presented in this thesis (modeling, meshing) and enable to modify effortlessly the geometry (e.g. centerline data point modification) and the topology (e.g. branch removal, branch angle editing) at every step of the meshing process, based on the medical image in a 3D viewer. We used this tool to manually correct the defects of the meshes for BraVa patients and release the first open database of ready-for-CFD large cerebral arterial network meshes.

In future work, we would like to investigate some applications for this database. In particular, our framework gives interesting opportunities for the generation of realistic cerebral vascular networks using graph deep learning generative models as used in other fields [57]. Besides, it could be used to generate synthetic MRA images with more realistic vessel shapes than existing generation methods [30]. The ground truth provided would not be limited to the segmentation itself, but also include 3D mesh and topology information, offering new opportunities for the training of deep-learning algorithms.

---

# Appendix

Here we detail the resolution of the system used to perform penalized spline approximation with fixed end points tangents, as mentioned in Section 2.3.4 of the main article. We want to approximate a dataset of a set  $m$  points with  $x$  coordinates,  $\{D_0, D_1, \dots, D_{m-1}\}$  using P-splines. A spline  $C$  is defined as

$$C(t) = \sum_{i=0}^{n-1} N_{i,p}(t)P_i,$$

for  $t \in [0, 1]$ , where  $N_{i,p}$  is the  $i$ th basis spline function of order  $p$  and  $\{P_0, P_1, \dots, P_{n-1}\}$  the  $n$  control points of the spline.

The cost function used for penalized spline approximation is:

$$f(P_0, \dots, P_{n-1}) = \sum_{k=0}^m |D_k - C(t_k)|^2 + \lambda \sum_{j=2}^n (P_j - 2P_{j-1} + P_{j-2})^2,$$

where  $t_k$  is a parametrization chosen following the data, for instance using the chord length method. The first term of the sum  $S_1$  minimizes the distance to the data, while the second term  $S_2$  is a smoothing term. Both are balanced by the value  $\lambda$ .

In this problem, we want to fix both the end points and the end tangents, the magnitude of tangents being free. As we work with clamped curves,  $C(0) = P_0$  and  $C(1) = P_{n-1}$ . Moreover,  $C'(0)$  (respectively  $C'(1)$ ) is in the same direction as vector  $P_1 - P_0$  (respectively  $P_{n-2} - P_{n-1}$ ). If we note  $D_0$  and  $D_{m-1}$  the fixed end points and  $T_0$  and  $T_1$  the fixed end tangents, the following new conditions are defined:

$$\begin{cases} P_0 = D_0 \\ P_{n-1} = D_{m-1} \\ P_1 = P_0 + \alpha T_0 \\ P_{n-1} = P_{n-2} + \beta T_1, \end{cases} \quad (6.1)$$

where  $\alpha$  and  $\beta$ , the end tangent magnitude, are additional parameters to optimize. Let us find the partial derivatives of the terms  $S_1$  and  $S_2$ .

## Partial derivatives of the term $S_1$

The constraint system 6.1 gives

$$S_1 = \sum_{k=0}^m |D_k - N_{0,p}(t_k)D_0 - N_{1,p}(t_k)(D_0 + \alpha T_1) - N_{n-2,p}(t_k)(D_{m-1} + \beta T_2) - N_{n-1,p}(t_k)D_{m-1} - \sum_{i=2}^{n-3} N_{i,p}(t_k)P_i|^2.$$

With  $q_1 = (N_{0,p}(t_k) + N_{1,p}(t_k))D_0 + (N_{n-1,p}(t_k) + N_{n-2,p}(t_k))D_{m-1}$ ,

$$\begin{aligned} S_1 &= \sum_{k=0}^m [(D_k - q_1 - N_{1,p}(t_k)\alpha T_1 - N_{n-2,p}(t_k)\beta T_2) \cdot (D_k - q_1 - N_{1,p}(t_k)\alpha T_1 - N_{n-2,p}(t_k)\beta T_2) \\ &\quad - 2 \sum_{i=2}^{n-3} N_{i,p}(t_k)P_i \cdot (D_k - q_1 - N_{1,p}(t_k)\alpha T_1 - N_{n-2,p}(t_k)\beta T_2) + (\sum_{i=2}^{n-3} N_{i,p}(t_k)P_i) \cdot (\sum_{i=2}^{n-3} N_{i,p}(t_k)P_i)] \end{aligned}$$

Let us find the partial derivatives of  $S_1$  according to parameters  $\alpha$ ,  $\beta$ ,  $P$  to optimize. For all  $g \in \{2, n-3\}$ ,

$$\frac{\partial S_1}{\partial P_g} = 2 \sum_{k=0}^m [-N_{g,p}(t_k)(D_k - q_1 - N_{1,p}\alpha T_1 - N_{n-2,p}\beta T_2) + N_{g,p}(t_k) \sum_{i=2}^{n-3} N_{i,p}(t_k)P_i].$$

$$\frac{\partial S_1}{\partial P_g} = 2 \sum_{k=0}^m [-N_{g,p}(t_k)(D_k - q_1) + N_{g,p}(N_{1,p}\alpha T_1 + N_{n-2,p}\beta T_2) + N_{g,p}(t_k) \sum_{i=2}^{n-3} N_{i,p}(t_k)P_i].$$

In addition, we have

$$\frac{\partial S_1}{\partial \alpha} = \sum_{k=0}^m [-2N_{1,p}(t_k)T_1(D_k - q_1 - N_{1,p}(t_k)\alpha T_1 - N_{n-2,p}(t_k)\beta T_2) + 2 \sum_{i=2}^{n-3} N_{i,p}(t_k)N_{1,p}(t_k)P_i T_1],$$

$$\frac{\partial S_1}{\partial \beta} = \sum_{k=0}^m [-2N_{n-2,p}(t_k)T_2(D_k - q_1 - N_{1,p}(t_k)\alpha T_1 - N_{n-2,p}(t_k)\beta T_2) + 2 \sum_{i=2}^{n-3} N_{i,p}(t_k)N_{n-2,p}(t_k)P_i T_2].$$

Let us define the following matrices:

$$N_{4m \times 4(n-4)+2} = \begin{pmatrix} N_{1,p}(t_0)T_{1x} & N_{2,p}(t_0) & 0 & \dots & 0 & 0 & N_{n-2,p}(t_0)T_{2x} \\ N_{1,p}(t_0)T_{1y} & 0 & N_{2,p}(t_0) & \dots & 0 & 0 & N_{n-2,p}(t_0)T_{2y} \\ N_{1,p}(t_0)T_{1z} & 0 & 0 & \dots & N_{n-3,p}(t_0) & 0 & N_{n-2,p}(t_0)T_{2z} \\ N_{1,p}(t_0)T_{1r} & 0 & 0 & \dots & 0 & N_{n-3,p}(t_0) & N_{n-2,p}(t_0)T_{2r} \\ N_{1,p}(t_1)T_{1x} & N_{2,p}(t_1) & 0 & \dots & 0 & 0 & N_{n-2,p}(t_1)T_{2x} \\ N_{1,p}(t_1)T_{1y} & 0 & N_{2,p}(t_1) & \dots & 0 & 0 & N_{n-2,p}(t_1)T_{2y} \\ N_{1,p}(t_1)T_{1z} & 0 & 0 & \dots & N_{n-3,p}(t_1) & 0 & N_{n-2,p}(t_1)T_{2z} \\ N_{1,p}(t_1)T_{1r} & 0 & 0 & \dots & 0 & N_{n-3,p}(t_1) & N_{n-2,p}(t_1)T_{2r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ N_{1,p}(t_{m-1})T_{1x} & N_{2,p}(t_{m-1}) & 0 & \dots & 0 & 0 & N_{n-2,p}(t_{m-1})T_{2x} \\ N_{1,p}(t_{m-1})T_{1y} & 0 & N_{2,p}(t_{m-1}) & \dots & 0 & 0 & N_{n-2,p}(t_{m-1})T_{2y} \\ N_{1,p}(t_{m-1})T_{1z} & 0 & 0 & \dots & N_{n-3,p}(t_{m-1}) & 0 & N_{n-2,p}(t_{m-1})T_{2z} \\ N_{1,p}(t_{m-1})T_{1r} & 0 & 0 & \dots & 0 & N_{n-3,p}(t_{m-1}) & N_{n-2,p}(t_{m-1})T_{2r} \end{pmatrix}$$

$$P_{(4(n-4)+2) \times 1} = \begin{pmatrix} \alpha \\ P_{2x} \\ P_{2y} \\ P_{2z} \\ P_{2r} \\ P_{3x} \\ \dots \\ P_{(n-3)x} \\ P_{(n-3)y} \\ P_{(n-3)z} \\ P_{(n-3)r} \\ \beta \end{pmatrix}, \quad D_{4m \times 1} = \begin{pmatrix} D_{0x} \\ D_{0y} \\ D_{0z} \\ D_{0r} \\ D_{1x} \\ \dots \\ D_{(m-1)x} \\ D_{(m-1)y} \\ D_{(m-1)z} \\ D_{(m-1)r} \end{pmatrix},$$

$$Q_{1(4m \times 1)} = \begin{pmatrix} (N_{0,p}(t_0) + N_{1,p}(t_0))D_{0x} + (N_{n-1,p}(t_0) + N_{n-2,p}(t_0))D_{(m-1)x} \\ (N_{0,p}(t_0) + N_{1,p}(t_0))D_{0y} + (N_{n-1,p}(t_0) + N_{n-2,p}(t_0))D_{(m-1)y} \\ (N_{0,p}(t_0) + N_{1,p}(t_0))D_{0z} + (N_{n-1,p}(t_0) + N_{n-2,p}(t_0))D_{(m-1)z} \\ (N_{0,p}(t_0) + N_{1,p}(t_0))D_{0r} + (N_{n-1,p}(t_0) + N_{n-2,p}(t_0))D_{(m-1)r} \\ (N_{0,p}(t_1) + N_{1,p}(t_1))D_{0x} + (N_{n-1,p}(t_1) + N_{n-2,p}(t_1))D_{(m-1)x} \\ \dots \\ (N_{0,p}(t_{m-1}) + N_{1,p}(t_{m-1}))D_{0x} + (N_{n-1,p}(t_{m-1}) + N_{n-2,p}(t_{m-1}))D_{(m-1)x} \\ (N_{0,p}(t_{m-1}) + N_{1,p}(t_{m-1}))D_{0y} + (N_{n-1,p}(t_{m-1}) + N_{n-2,p}(t_{m-1}))D_{(m-1)y} \\ (N_{0,p}(t_{m-1}) + N_{1,p}(t_{m-1}))D_{0z} + (N_{n-1,p}(t_{m-1}) + N_{n-2,p}(t_{m-1}))D_{(m-1)z} \\ (N_{0,p}(t_{m-1}) + N_{1,p}(t_{m-1}))D_{0r} + (N_{n-1,p}(t_{m-1}) + N_{n-2,p}(t_{m-1}))D_{(m-1)r} \end{pmatrix}.$$

The system of partial derivatives can be written as

$$2(N^T N)P - 2N^T(D - Q_1),$$

so for the term  $S_1$  we have the following equation.

$$0 = (N^T N)P - N^T(D - Q_1) \quad (6.2)$$

## Partial derivatives of the term $S_2$

The partial derivatives of the term  $S_2$  for the control points are

$$\frac{\partial S_2}{\partial P_1} = 2(3D_0 + 5T_1\alpha - 4P_2 + P_3),$$

$$\frac{\partial S_2}{\partial P_2} = 2(-3D_0 - 4\alpha T_1 + 6P_2 - 4P_3 + P_4),$$

$$\frac{\partial S_2}{\partial P_{n-3}} = 2(P_{n-1} - 4P_{n-2} + 6P_{n-3} - 4P_{n-4} + P_{n-5}),$$

$$\frac{\partial S_2}{\partial P_{n-2}} = 2(-2D_{m-1} + 5P_{n-2} - 4P_{n-3} + P_{n-4}).$$

For all  $g \in \{1, n-4\}$ , we can write

$$\frac{\partial S_2}{\partial P_g} = 2(P_{g-2} - 4P_{g-1} + 6P_g - 4P_{g+1} + P_{g+2}).$$

In addition, we have

$$\frac{\partial S_2}{\partial \alpha} = 2(3D_0T_1 + 5T_1.T_1\alpha - 4T_2T_1\beta - 4T_1P_3),$$

$$\frac{\partial S_2}{\partial \beta} = 2((-2 + 5)T_2D_{m-1} + 5\beta T_2.T_2 - 4T_2P_{n-3} + T_2P_{n-4}).$$

$$U_{(4n-2) \times (4n-2)} = \begin{pmatrix} 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -2 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\Delta_{(4(n-4)-4) \times (4(n-4)+2)} = \begin{pmatrix} 5T_1.T_1 & -4 & -4 & -4 & -4 & 1 & 1 & 1 & 1 & 0 & \dots & \dots & \dots & \dots & \dots & 0 \\ -4T_{1x} & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & \dots & 0 \\ -4T_{1y} & 0 & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \dots & 0 \\ -4T_{1z} & 0 & 0 & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 \\ -4T_{1r} & 0 & 0 & 0 & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 & \dots & 0 \\ 1T_{1x} & -4 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & 0 & \dots \\ 1T_{1y} & 0 & -4 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & 0 & 1 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 6 & 0 & 0 & 0 & -4T_{2x} \\ 0 & \dots & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 6 & 0 & 0 & -4T_{2y} \\ 0 & \dots & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 6 & 0 & -4T_{2z} \\ 0 & \dots & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -4 & 0 & 0 & 0 & 6 & -4T_{2r} \\ 0 & \dots & \dots & \dots & \dots & 0 & 0 & 1 & 1 & 1 & 1 & -4 & -4 & -4 & -4 & 5T_2.T_2 \end{pmatrix}$$

and

$$Q_{2((4(n-4)+2) \times x)} = \begin{pmatrix} -2 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ 0 \end{pmatrix} + \begin{pmatrix} 5 \\ -4 \\ -4 \\ -4 \\ -4 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ \dots \\ 0 \end{pmatrix} D_0 + \begin{pmatrix} 0 \\ \dots \\ \dots \\ \dots \\ \dots \\ \dots \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ -2 \end{pmatrix} + \begin{pmatrix} 0 \\ \dots \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ -4 \\ -4 \\ -4 \\ -4 \\ 5 \end{pmatrix} D_{m-1}$$

The equation system for the term  $S_2$  can be written

$$0 = \Delta P + Q_2. \quad (6.3)$$

From equations 6.2 and 6.3, the whole system to solve is

$$P = ((1 - \lambda)(N^T N - \lambda \Delta))^{-1} \cdot ((1 - \lambda)(N^T)(D - Q_1) - \lambda Q_2).$$

---

# Communications

## Peer-reviewed journal articles

Méghane Decroocq, Carole Frindel, Pierre Rougé, Makoto Ohta, Guillaume Lavoué, *Modeling and hexahedral meshing of cerebral arterial networks from centerlines*, Medical Image Analysis (accepted on April 21st, 2023) (**Chapter 1,2,3,4**)

## Peer-reviewed articles in international conferences

Méghane Decroocq, Guillaume Lavoué, Makoto Ohta and Carole Frindel, *A Software to Visualize, Edit, Model and Mesh Vascular Networks*, 2022 44th Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC) Scottish Event Campus, Glasgow, UK, July 11-15, 2022 (**Chapter 5**)

Noëlie Debs, Méghane Decroocq, Tae-Hee Cho, David Rousseau, Carole Frindel, *Evaluation of the realism of an MRI simulator for stroke lesion prediction using convolutional neural network*, International Workshop on Simulation and Synthesis in Medical Imaging, SASHIMI, pp. 151-160, Springer, Cham, Oct. 2019 (**Chapter 1**).

## International conference presentations

Méghane Decroocq, Carole Frindel, Makoto Ohta, Guillaume Lavoué, *Structured meshing of large vascular networks for computational fluid dynamics*, Poster presentation, VPH



## *Chapter 6. Conclusion*

(Virtual Physiological Human), August 26-28 2020, Paris, France.

Meghane Decroocq, Carole Frindel, Makoto Ohta and Guillaume Lavoue, *Meshing Arterial Networks From Manually Extracted Centerlines*, Oral presentation, ICFD, October 28-30 2020, Sendai, Japon.

---

# Acknowledgments

I would like to thank my supervisors; Carole Frindel, Guillaume Lavoué and Makoto Ohta. Carole, you have trusted me from the beginning when I was a master student, you made me discover the world of medical imaging and you have supported me when I had doubts or when I lost confidence. Guillaume, you have given so much time and energy to my thesis project, you were always there to answer my questions and to propose solutions, whether it was related to my thesis or not. Ohta sensei, thank you for inviting me to Ohta Lab; I could experience working abroad and get enriched by a different culture and language. I would also like to thank Simon Tupin for the passionate scientific discussions - which were at the origin of my thesis project! - and for sharing his experience of living in Japan, Hitomi Anzai for her technical help in particular in fluid dynamics, and Yasuno Emori for her kindness and her help in daily life and administrative issues. Finally, I would like to thank the professors who accepted to be part in my thesis jury in Japan; Pr. Noriyasu Homma and M.D. Kuniyasu Niizuma, and in France; Nicolas Passat, Dominique Bechmann, Bastien Chopard, Irène Vignon-Clémentel, Tae-Hee Cho.

I am thankful to all my friends of the CREATIS laboratory, master or PhD students, who gave me their support and friendship; Noélie Debs, Théo Jourdan, Yunyun Sun, Emile Saillard, Pierre Rougé. I would also like to thank the people (master students, interns or professors) who helped me during the realization of my thesis project; Timothée Jacques-son who was always ready to answer my medical questions despite his busy schedule as a neurosurgeon, Odysée Merveille, Stéphanie Salmon, and more generally the members of the ANR PreSPIN who taught me a lot in the fields of segmentation and blood flow simulation for cerebral arteries, Pierre Ferrié who helped me in the most critical moments

## *Chapter 6. Conclusion*

of the technical problems of my computer, Erwan Maury, Kassidy Porche who participated in the development of the algorithms which compose this thesis. I would also like to thank my friends from LIRIS, Jocelyn Meyron, Yana Nehmé, Sandra Clémenceau, Gabriel Meynet, and the members of Ohta Laboratory, who welcomed me warmly and helped me when I needed it; Yuta Muramoto, Fangjia Pan, Yutaro Kohata, Jin Kazuyoshi, Hanif Saifurrahman, Yukiko Kojima, Keiichiro Shiraishi, Riko Hasegawa, Daisuke Goanno. Finally, a special thank to Shiddiq Hashuro and Kotaro Daibo, who were with me in room 208 in the successes and failures of my thesis, always ready to provide snacks if needed, Nicolas Jacquelin, for his unwavering support and his expert advice in deep learning and Etienne Sauvajon, who encouraged me all along my studies. Finally, I would like to thank my family, for their love and support during these three years.

---

## Bibliography

- [1] Marwan Abdellah, Nadir Román Guerrero, Samuel Lapere, Jay S Coggan, Daniel Keller, Benoit Coste, Snigdha Dagar, Jean-Denis Courcol, Henry Markram, and Felix Schürmann. Interactive visualization and analysis of morphological skeletons of brain vasculature networks with vessmorphovis. *Bioinformatics*, 36(Supplement\_1):i534–i541, 2020.
- [2] JPKH Alastruey, KH Parker, J Peiró, SM Byrd, and SJ Sherwin. Modelling the circle of willis to assess the effects of anatomical variations and occlusions on cerebral flows. *Journal of biomechanics*, 40(8):1794–1805, 2007.
- [3] Aneurisk-Team. AneuriskWeb project website, <http://ecm2.mathcs.emory.edu/aneuriskweb>. Web Site, 2012.
- [4] Luca Antiga, Bogdan Ene-Iordache, Lionello Caverni, Gian Paolo Cornalba, and Andrea Remuzzi. Geometric reconstruction for computational mesh generation of arterial bifurcations from ct angiography. *Computerized Medical Imaging and Graphics*, 26(4):227–235, 2002.
- [5] Luca Antiga and David A Steinman. Robust and objective decomposition and mapping of bifurcating vessels. *IEEE transactions on medical imaging*, 23(6):704–713, 2004.
- [6] Stephen R Aylward, Julien Jomier, Christelle Vivert, Vincent LeDigarcher, and Elizabeth Bullitt. Spatial graphs for intra-cranial vascular network characterization, gener-

- ation, and discrimination. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 59–66. Springer, 2005.
- [7] Susann Beier, John Ormiston, Mark Webster, John Cater, Stuart Norris, Pau Medrano-Gracia, Alistair Young, and Brett Cowan. Impact of bifurcation angle and other anatomical characteristics on blood flow—a computational study of non-stented and stented coronary arteries. *Journal of biomechanics*, 49(9):1570–1582, 2016.
- [8] Sylvain Bouix, Kaleem Siddiqi, and Allen Tannenbaum. Flux driven automatic centerline extraction. *Medical image analysis*, 9(3):209–221, 2005.
- [9] Jörg Bredno, Mark E Olszewski, and Max Wintermark. Simulation model for contrast agent dynamics in brain perfusion scans. *Magnetic Resonance in Medicine*, 64(1):280–290, 2010.
- [10] Elizabeth Bullitt, Donglin Zeng, Guido Gerig, Stephen Aylward, Sarang Joshi, J Keith Smith, Weili Lin, and Matthew G Ewend. Vessel tortuosity and brain tumor malignancy: a blinded study<sup>1</sup>. *Academic radiology*, 12(10):1232–1240, 2005.
- [11] Jacob Bumgarner and Randy Nelson. A novel and open-source application for vasculature dataset analysis and visualization. 2021. Preprint on webpage at <https://www.researchsquare.com/article/rs-608609/v1>.
- [12] Robert C Cannon, DA Turner, GK Pyapali, and HV Wheal. An on-line archive of reconstructed hippocampal neurons. *Journal of neuroscience methods*, 84(1-2):49–54, 1998.
- [13] Bongjae Chung and Juan Raul Cebal. Cfd for evaluation and treatment planning of aneurysms: review of proposed clinical uses and their challenges. *Annals of biomedical engineering*, 43(1):122–138, 2015.
- [14] Peter Craven and Grace Wahba. Smoothing noisy data with spline functions. *Numerische mathematik*, 31(4):377–403, 1978.
- [15] Carl De Boor and Carl De Boor. *A practical guide to splines*, volume 27. springer-verlag New York, 1978.

- [16] Gianluca De Santis, Matthieu De Beule, Patrick Segers, Pascal Verdonck, and Benedict 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.
- [17] Gianluca De Santis, Matthieu De Beule, Koen Van Canneyt, Patrick Segers, Pascal Verdonck, and Benedict Verhegghe. Full-hexahedral structured meshing for image-based computational vascular modeling. *Medical engineering & physics*, 33(10):1318–1325, 2011.
- [18] Gianluca De Santis, Peter Mortier, Matthieu De Beule, Patrick Segers, Pascal Verdonck, and Benedict Verhegghe. Patient-specific computational fluid dynamics: structured mesh generation from coronary angiography. *Medical & biological engineering & computing*, 48(4):371–380, 2010.
- [19] Noëlie Debs, Méghane Decroocq, Tae-Hee Cho, David Rousseau, and Carole Frindel. Evaluation of the realism of an mri simulator for stroke lesion prediction using convolutional neural network. In *Simulation and Synthesis in Medical Imaging: 4th International Workshop, SASHIMI 2019, Held in Conjunction with MICCAI 2019, Shenzhen, China, October 13, 2019, Proceedings 4*, pages 151–160. Springer, 2019.
- [20] Noëlie Debs, Pejman Rasti, Léon Victor, Tae-Hee Cho, Carole Frindel, and David Rousseau. Simulated perfusion mri data to boost training of convolutional neural networks for lesion fate prediction in acute stroke. *Computers in Biology and Medicine*, 116:103579, 2020.
- [21] Paul HC Eilers and Brian D Marx. Flexible smoothing with b-splines and penalties. *Statistical science*, 11(2):89–121, 1996.
- [22] Valery L Feigin, Benjamin A Stark, Catherine Owens Johnson, Gregory A Roth, Catherine Bisignano, Gdiom Gebreheat Abady, Mitra Abbasifard, Mohsen Abbasi-Kangevari, Foad Abd-Allah, Vida Abedi, et al. Global, regional, and national burden of stroke and its risk factors, 1990–2019: A systematic analysis for the global burden of disease study 2019. *The Lancet Neurology*, 20(10):795–820, 2021.

- [23] Christophe Geuzaine and Jean-François Remacle. Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*, 79(11):1309–1331, 2009.
- [24] Mahsa Ghaffari, Chih-Yang Hsu, and Andreas A Linninger. Automatic reconstruction and generation of structured hexahedral mesh for non-planar bifurcations in vascular networks. In *Computer Aided Chemical Engineering*, volume 37, pages 635–640. Elsevier, 2015.
- [25] Mahsa Ghaffari, Kevin Tangen, Ali Alaraj, Xinjian Du, Fady T Charbel, and Andreas A Linninger. Large-scale subject-specific cerebral arterial tree modeling using automated parametric mesh generation for blood flow simulation. *Computers in biology and medicine*, 91:353–365, 2017.
- [26] Hamidreza Gharahi, Byron A Zambrano, David C Zhu, J Kevin DeMarco, and Seungik Baek. Computational fluid dynamic simulation of human carotid artery bifurcation based on anatomy and volumetric blood flow rate measured with magnetic resonance imaging. *International journal of advances in engineering sciences and applied mathematics*, 8(1):46–60, 2016.
- [27] S Glaßer, P Berg, M Neugebauer, and B Preim. Reconstruction of 3d surface meshes for blood flow simulations of intracranial aneurysms. In *Proceedings of the Conference of the German Society for Computer and Robotic Assisted Surgery*, pages 163–168, 2015.
- [28] Jixiang Guo, Shun Li, Yim Pan Chui, Jing Qin, and Pheng Ann Heng. Mesh quality oriented 3d geometric vascular modeling based on parallel transport frame. *Computers in biology and medicine*, 43(7):879–888, 2013.
- [29] IV Haljasmaa, AM Robertson, and GP Galdi. On the effect of apex geometry on wall shear stress and pressure in two-dimensional models of arterial bifurcations. *Mathematical Models and Methods in Applied Sciences*, 11(03):499–520, 2001.
- [30] Ghassan Hamarneh and Preet Jassi. Vascusynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis. *Computerized medical imaging and graphics*, 34(8):605–616, 2010.

- [31] Xiaoxiao Han, Richard Bibb, and Russell Harris. Design of bifurcation junctions in artificial vascular vessels additively manufactured for skin tissue engineering. *Journal of Visual Languages & Computing*, 28:238–249, 2015.
- [32] Si Hang. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Trans. Math. Softw.*, 41(2):11, 2015.
- [33] Jiafa He, Chengwei Pan, Can Yang, Ming Zhang, Yang Wang, Xiaowei Zhou, and Yizhou Yu. Learning hybrid representations for automatic 3d vessel centerline extraction. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 24–34. Springer, 2020.
- [34] Adam Hilbert, Vince I Madai, Ela M Akay, Orhun U Aydin, Jonas Behland, Jan Sobesky, Ivana Galinovic, Ahmed A Khalil, Abdel A Taha, Jens Wuerfel, et al. Bravenet: fully automated arterial brain vessel segmentation in patients with cerebrovascular disease. *Frontiers in artificial intelligence*, page 78, 2020.
- [35] Erlend Hodneland, Erik Hanson, Ove Sævareid, Geir Nævdal, Arvid Lundervold, Veronika Šoltészová, Antonella Z Munthe-Kaas, Andreas Deistung, Jürgen R Reichenbach, and Jan M Nordbotten. A new framework for assessing subject-specific whole brain circulation and perfusion using mri-based measurements and a multi-scale continuous flow model. *PLoS computational biology*, 15(6):e1007073, 2019.
- [36] Qingqi Hong, Qingde Li, Beizhan Wang, Kunhong Liu, Fan Lin, Juncong Lin, Xuan Cheng, Zhihong Zhang, and Ming Zeng. Accurate geometry modeling of vasculatures using implicit fitting with 2d radial basis functions. *Computer Aided Geometric Design*, 62:206–216, 2018.
- [37] Qingqi Hong, Qingde Li, Beizhan Wang, Jie Tian, Fei Xu, Kunhong Liu, and Xuan Cheng. High-quality vascular modeling and modification with implicit extrusion surfaces for blood flow computations. *Computer Methods and Programs in Biomedicine*, 196:105598, 2020.
- [38] Richard Izzo, David Steinman, Simone Manini, and Luca Antiga. The vascular modeling toolkit: a python library for the analysis of tubular structures in medical images. *Journal of Open Source Software*, 3(25):745, 2018.



- [39] Zhixin Jiang, Hao Zhang, Yi Wang, and Seok-Bum Ko. Retinal blood vessel segmentation using fully convolutional network with transfer learning. *Computerized Medical Imaging and Graphics*, 68:1–15, 2018.
- [40] Kanchan Kapoor, Balbir Singh, and Inder Jit Dewan. Variations in the configuration of the circle of willis. *Anatomical science international*, 83(2):96–106, 2008.
- [41] Erwan Kerrien, Ahmed Yureidini, Jeremie Dequidt, Christian Duriez, René Anxionnat, and Stéphane Cotin. Blood vessel modeling for interactive simulation of interventional neuroradiology procedures. *Medical image analysis*, 35:685–698, 2017.
- [42] Deepak Keshwani, Yoshiro Kitamura, Satoshi Ihara, Satoshi Iizuka, and Edgar Simo-Serra. Topnet: Topology preserving metric learning for vessel tree reconstruction and labelling. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 14–23. Springer, 2020.
- [43] Marek Kociński, Andrzej Materka, Andreas Deistung, and Jürgen R Reichenbach. Centerline-based surface modeling of blood-vessel trees in cerebral 3d mra. In *2016 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, pages 85–90. IEEE, 2016.
- [44] Fanwei Kong, Nathan Wilson, and Shawn Shadden. A deep-learning approach for direct whole-heart mesh reconstruction. *Medical image analysis*, 74:102222, 2021.
- [45] Monique J Krabbe-Hartkamp, Jeroen Van der Grond, FE De Leeuw, Jan Cees de Groot, Ale Algra, Berend Hillen, MM Breteler, and WP Mali. Circle of willis: morphologic variation on three-dimensional time-of-flight mr angiograms. *Radiology*, 207(1):103–111, 1998.
- [46] Jonas Lamy, Odysée Merveille, Bertrand Kerautret, and Nicolas Passat. A benchmark framework for multiregion analysis of vesselness filters. *IEEE Trans. Medical Imaging*, 41(12):3649–3662, 2022.
- [47] Natalia Lewandowska and Jędrzej Mosieźny. Meshing strategy for bifurcation arteries in the context of blood flow simulation accuracy. In *E3S Web of Conferences*, volume 128, page 02003. EDP Sciences, 2019.

- [48] Yang Li, Jun Yang, Jiajia Ni, Ahmed Elazab, and Jianhuang Wu. Ta-net: Triple attention network for medical image segmentation. *Computers in Biology and Medicine*, 137:104836, 2021.
- [49] Karl-Fredrik Lindegaard, Tryggve Lundar, Jan Wiberg, D Sjøberg, Rune Aaslid, and Helge Nornes. Variations in middle cerebral artery blood flow investigated with noninvasive transcranial blood velocity measurements. *Stroke*, 18(6):1025–1030, 1987.
- [50] Marco Livesu, Alessandro Muntoni, Enrico Puppo, and Riccardo Scateni. Skeleton-driven adaptive hexahedral meshing of tubular shapes. In *Computer Graphics Forum*, volume 35, pages 237–246. Wiley Online Library, 2016.
- [51] Michelle Livne, Jana Rieger, Orhun Utku Aydin, Abdel Aziz Taha, Ela Marie Akay, Tabea Kossen, Jan Sobesky, John D Kelleher, Kristian Hildebrand, Dietmar Frey, et al. A u-net deep learning framework for high performance vessel segmentation in patients with cerebrovascular disease. *Frontiers in neuroscience*, 13:97, 2019.
- [52] Mark H Longair, Dean A Baker, and J Douglas Armstrong. Simple neurite tracer: open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics*, 27(17):2453–2454, 2011.
- [53] D Lopes, Hélder Puga, J Teixeira, and R Lima. Blood flow simulations in patient-specific geometries of the carotid artery: a systematic review. *Journal of Biomechanics*, pages 11–19, 2020.
- [54] Samar A Mahrous, Nor Azwadi Che Sidik, and Khalid M Saqr. Newtonian and non-newtonian cfd models of intracranial aneurysm: a review. *CFD Letters*, 12(1):62–86, 2020.
- [55] Gary E McVeigh, Christopher W Bratteli, Dennis J Morgan, Cheryl M Alinder, Stephen P Glasser, Stanley M Finkelstein, and Jay N Cohn. Age-related abnormalities in arterial compliance identified by pressure pulse contour analysis: aging and arterial compliance. *Hypertension*, 33(6):1392–1398, 1999.
- [56] Odyssée Merveille, Hugues Talbot, Laurent Najman, and Nicolas Passat. Curvilinear structure analysis by ranking the orientation responses of path operators. *IEEE transactions on pattern analysis and machine intelligence*, 40(2):304–317, 2017.

- [57] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4227–4236, 2021.
- [58] Kouichi Misaki, Kazuya Futami, Takehiro Uno, Iku Nambu, Akifumi Yoshikawa, Tomoya Kamide, and Mitsutoshi Nakada. Inflow hemodynamics of intracranial aneurysms: A comparison of computational fluid dynamics and 4d flow magnetic resonance imaging. *Journal of Stroke and Cerebrovascular Diseases*, 30(5):105685, 2021.
- [59] Sara Moccia, Elena De Momi, Sara El Hadji, and Leonardo S. Mattos. Blood vessel segmentation algorithms - review of methods, datasets and evaluation metrics. *Comput. Methods Programs Biomed.*, 158:71–91, 2018.
- [60] Lei Mou, Yitian Zhao, Huazhu Fu, Yonghuai Liu, Jun Cheng, Yalin Zheng, Pan Su, Jianlong Yang, Li Chen, Alejandro F Frangi, et al. Cs2-net: Deep learning segmentation of curvilinear structures in medical imaging. *Medical image analysis*, 67:101874, 2021.
- [61] Yuichi Murayama, Soichiro Fujimura, Tomoaki Suzuki, and Hiroyuki Takao. Computational fluid dynamics as a risk assessment tool for aneurysm rupture. *Neurosurgical focus*, 47(1):E12, 2019.
- [62] Jiajia Ni, Jianhuang Wu, Haoyu Wang, Jing Tong, Zhengming Chen, Kelvin KL Wong, and Derek Abbott. Global channel attention networks for intracranial vessel segmentation. *Computers in biology and medicine*, 118:103639, 2020.
- [63] Shunsuke Omodaka, Takashi Inoue, Kenichi Funamoto, Shin-ichirou Sugiyama, Hiroaki Shimizu, Toshiyuki Hayase, Akira Takahashi, and Teiji Tominaga. Influence of surface model extraction parameter on computational fluid dynamics modeling of cerebral aneurysms. *Journal of biomechanics*, 45(14):2355–2361, 2012.
- [64] Les A Piegl and Wayne Tiller. Least-squares b-spline curve approximation with arbitrary end derivatives. *Engineering with Computers*, 16(2):109–116, 2000.

- [65] Jennifer L Quon, Leo C Chen, Lily Kim, Gerald A Grant, Michael SB Edwards, Samuel H Cheshier, and Kristen W Yeom. Deep learning for automated delineation of pediatric cerebral arteries on pre-operative brain magnetic resonance imaging. *Frontiers in Surgery*, page 89, 2020.
- [66] Ansaar T Rai, Jeffery P Hogg, Brendan Cline, and Gerald Hobbs. Cerebrovascular geometry in the anterior circulation: an analysis of diameter, length and the vessel taper. *Journal of neurointerventional surgery*, 5(4):371–375, 2013.
- [67] Maria José Melo Ramos-Lima, Ismênia de Carvalho Brasileiro, Tamires Layane de Lima, and Pedro Braga-Neto. Quality of life after stroke: impact of clinical and sociodemographic factors. *Clinics*, 73, 2018.
- [68] Sherif Rashad, Shin-ichiro Sugiyama, Kuniyasu Niizuma, Kenichi Sato, Hidenori Endo, Shunsuke Omodaka, Yasushi Matsumoto, Miki Fujimura, and Teiji Tominaga. Impact of bifurcation angle and inflow coefficient on the rupture risk of bifurcation type basilar artery tip aneurysms. *Journal of neurosurgery*, 128(3):723–730, 2017.
- [69] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.
- [70] Laura M Sangalli, Piercesare Secchi, Simone Vantini, and Alessandro Veneziani. A case study in exploratory functional data analysis: geometrical features of the internal carotid artery. *Journal of the American Statistical Association*, 104(485):37–48, 2009.
- [71] Laura M Sangalli, Piercesare Secchi, Simone Vantini, and Alessandro Veneziani. Efficient estimation of three-dimensional curves and their derivatives by free-knot regression splines, applied to the analysis of inner carotid artery centrelines. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 58(3):285–306, 2009.
- [72] Marcelo Sanmartín, Javier Goicolea, Carlos García, Javier García, Antonio Crespo, Javier Rodríguez, and José M Goicolea. Influence of shear stress on in-stent restenosis: in vivo study using 3d reconstruction and computational fluid dynamics. *Revista Española de Cardiología (English Edition)*, 59(1):20–27, 2006.

- [73] Khalid M Saqr, Sherif Rashad, Simon Tupin, Kuniyasu Niizuma, Tamer Hassan, Teiji Tominaga, and Makoto Ohta. What does computational fluid dynamics tell us about intracranial aneurysms? a meta-analysis and critical review. *Journal of Cerebral Blood Flow & Metabolism*, 40(5):1021–1039, 2020.
- [74] Rohan Shad, Alexander D Kaiser, Sandra Kong, Robyn Fong, Nicolas Quach, Cayley Bowles, Patpilai Kasinpila, Yasuhiro Shudo, Jeffrey Teuteberg, Y Joseph Woo, et al. Patient-specific computational fluid dynamics reveal localized flow patterns predictive of post-left ventricular assist device aortic incompetence. *Circulation: Heart Failure*, 14(7):e008034, 2021.
- [75] C Shen, R Gharleghi, DD Li, M Stevens, S Dokos, and S Beier. Secondary flow in bifurcations—important effects of curvature, bifurcation angle and stents. *Journal of Biomechanics*, 129:110755, 2021.
- [76] Suprosanna Shit, Johannes C Paetzold, Anjany Sekuboyina, Ivan Ezhov, Alexander Unger, Andrey Zhylka, Josien PW Pluim, Ulrich Bauer, and Bjoern H Menze. cldice—a novel topology-preserving loss function for tubular structure segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16560–16569, 2021.
- [77] Michal Sofka and Charles V Stewart. Retinal vessel centerline extraction using multiscale matched filters, confidence and edge measures. *IEEE transactions on medical imaging*, 25(12):1531–1546, 2006.
- [78] Shin-ichiro Sugiyama, Kuniyasu Niizuma, Kenichi Sato, Sherif Rashad, Misaki Kohama, Hidenori Endo, Toshiki Endo, Yasushi Matsumoto, Makoto Ohta, and Teiji Tominaga. Blood flow into basilar tip aneurysms: a predictor for recanalization after coil embolization. *Stroke*, 47(10):2541–2547, 2016.
- [79] Amirtahà Taebi, Rex M Pillai, Bahman S. Roudsari, Catherine T Vu, and Emilie Roncali. Computational modeling of the liver arterial blood flow for microsphere therapy: effect of boundary conditions. *Bioengineering*, 7(3):64, 2020.
- [80] Giles Tetteh, Velizar Efremov, Nils D Forkert, Matthias Schneider, Jan Kirschke, Bruno Weber, Claus Zimmer, Marie Piraud, and Björn H Menze. Deepvesselnet:

- Vessel segmentation, centerline prediction, and bifurcation detection in 3-d angiographic volumes. *Frontiers in Neuroscience*, page 1285, 2020.
- [81] Ryo Torii, Marie Oshima, Toshio Kobayashi, Kiyoshi Takagi, and Tayfun E Tezduyar. Fluid–structure interaction modeling of blood flow and cerebral aneurysm: significance of artery and aneurysm shapes. *Computer Methods in Applied Mechanics and Engineering*, 198(45-46):3613–3621, 2009.
- [82] Chaman Singh Verma, Paul F Fischer, Seung E Lee, and Francis Loth. An all-hex meshing strategy for bifurcation geometries in vascular flow simulation. In *Proceedings of the 14th international meshing roundtable*, pages 363–375. Springer, 2005.
- [83] Vincent Vidal, Guillaume Lavoué, and Florent Dupont. Low budget and high fidelity relaxed 567-remeshing. *Computers & Graphics*, 47:16–23, 2015.
- [84] Irene E Vignon-Clementel, CA Figueroa, KE Jansen, and CA Taylor. Outflow boundary conditions for 3d simulations of non-periodic blood flow and pressure fields in deformable arteries. *Computer methods in biomechanics and biomedical engineering*, 13(5):625–640, 2010.
- [85] Samir Vinchurkar and P Worth Longest. Evaluation of hexahedral, prismatic and hybrid mesh styles for simulating respiratory aerosol dynamics. *Computers & Fluids*, 37(3):317–331, 2008.
- [86] Paul Viville, Pierre Kraemer, and Dominique Bechmann. Hexahedral mesh generation for tubular shapes using skeletons and connection surfaces. In *16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2021.
- [87] Tomoya Watanabe, Haruo Isoda, Yasuo Takehara, Masaki Terada, Takehiro Naito, Takafumi Kosugi, Yuki Onishi, Chiharu Tanoi, and Takashi Izumi. Hemodynamic vascular biomarkers for initiation of paraclinoid internal carotid artery aneurysms using patient-specific computational fluid dynamic simulation based on magnetic resonance imaging. *Neuroradiology*, 60(5):545–555, 2018.
- [88] Udaranga Wickramasinghe, Edoardo Remelli, Graham Knott, and Pascal Fua. Voxel2mesh: 3d mesh model generation from volumetric data. In *International Con-*

- ference on Medical Image Computing and Computer-Assisted Intervention*, pages 299–308. Springer, 2020.
- [89] Susan N Wright, Peter Kochunov, Fernando Mut, Maurizio Bergamino, Kerry M Brown, John C Mazziotta, Arthur W Toga, Juan R Cebral, and Giorgio A Ascoli. Digital reconstruction and morphometric analysis of human brain arterial vasculature from magnetic resonance angiography. *Neuroimage*, 82:170–181, 2013.
- [90] Guanglei Xiong, Suraj Musuvathy, and Tong Fang. Automated structured all-quadrilateral and hexahedral meshing of tubular surfaces. In *Proceedings of the 21st international meshing roundtable*, pages 103–120. Springer, 2013.
- [91] Guanglei Xiong and Charles A Taylor. Influence of vessel roughness on wall shear stress in image-based blood flow modeling. In *2010 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 33–36. IEEE, 2010.
- [92] Edward L Yellin. Laminar-turbulent transition process in pulsatile flow. *Circulation research*, 19(4):791–804, 1966.
- [93] Hasballah Zakaria, Anne M Robertson, and Charles W Kerber. A parametric model for studies of flow in arterial bifurcations. *Annals of biomedical Engineering*, 36:1515, 2008.
- [94] Pengyue Zhang, Fusheng Wang, and Yefeng Zheng. Deep reinforcement learning for vessel centerline tracing in multi-modality 3d volumes. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 755–763. Springer, 2018.
- [95] Yongjie Zhang, Yuri Bazilevs, Samrat Goswami, Chandrajit L Bajaj, and Thomas JR 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.



## FOLIO ADMINISTRATIF

### THESE DE L'INSA LYON, MEMBRE DE L'UNIVERSITE DE LYON

NOM : Decroocq

DATE de SOUTENANCE : 31/05/2023

Prénoms : Méghane Chloé Jade

TITRE : Development of a centerline-based arterial network modeling and meshing framework for hemodynamic studies of cerebrovascular pathologies

Développement d'une méthode de modélisation et maillage de réseaux artériels basée sur la ligne centrale pour les études hémodynamiques des pathologies cérébrovasculaires

NATURE : Doctorat

Numéro d'ordre : 2023ISAL0034

Ecole doctorale : Infomaths

Spécialité : Informatique et applications

RESUME : La dynamique des fluides numérique (CFD) est une technique qui fournit des informations précieuses sur l'écoulement du sang à partir de la géométrie vasculaire, ce qui permet de comprendre, de diagnostiquer et de prédire l'issue des maladies vasculaires. Cependant, la résolution des images médicales actuelles n'est pas satisfaisante pour obtenir des maillages répondant aux exigences de la CFD, en particulier pour des réseaux vasculaires larges et complexes telles que les réseaux d'artères cérébrales. Dans cette thèse, nous proposons une méthode en deux étapes pour produire des maillages adaptés à la CFD à partir d'une représentation simplifiée des réseaux vasculaires par leurs lignes centrales. Dans la première étape, pour remédier aux défauts de la représentation basée sur les lignes centrales (dispersées, bruitées), une étape de modélisation a été introduite pour reconstruire un modèle anatomique réaliste à partir d'a-priori sur la géométrie des vaisseaux et des bifurcations. Ensuite, une étape de maillage a été développée pour créer un maillage volumique de bonne qualité avec des cellules hexaédriques structurées et orientées selon l'écoulement du fluide qui répondent aux exigences de la CFD. Nous avons montré que la méthode proposée résout certaines des lacunes des maillages obtenus à partir des méthodes de segmentation de la littérature (e.g. vaisseaux fusionnés ou déconnectés), et conduit à des résultats de CFD plus précis pour un coût de calcul moindre. Cette méthode de maillage a été complétée par une interface utilisateur facilitant la visualisation et l'édition des modèles de réseaux vasculaires, et ouvrant notre code à des utilisateurs non experts. Grâce à cette interface, nous avons créé une base de données de maillages de réseaux entiers d'artères cérébrales qui pourra être utilisée pour l'évaluation de dispositifs médicaux et les études hémodynamiques.

MOTS-CLÉS : Réseaux artériels cérébraux, Maillage hexaédrique, Modélisation paramétrique, Dynamique des fluides computationnelle, Interface utilisateur

Laboratoires de recherche : CREATIS, LIRIS (INSA Lyon), Ohta Laboratory, IFS (Tohoku University)

Directeur de thèse: Guillaume Lavoué, Carole Frindel (INSA Lyon) et Makoto Ohta (Tohoku University)

Président de jury : Irène Vignon-Clementel

Composition du jury : Dominique Bechmann (Rapporteuse), Nicolas Passat (Rapporteur), Irène Vignon-Clementel (Examinatrice), Bastien Chopard (Examineur), Tae-Hee Cho (Invité)