



POLITECNICO DI MILANO  
DIPARTIMENTO DI BIOINGEGNERIA  
DOTTORATO DI RICERCA IN BIOINGEGNERIA  
XV CICLO – 1999-2002

# **Patient-Specific Modeling of Geometry and Blood Flow in Large Arteries**

Ph.D. Thesis of:  
**Luca Antiga**

Tutor:  
**Prof. Andrea Remuzzi**

Co-Advisers:  
**Prof. Roberto Fumero**  
**Ing. Alberto Redaelli**

Coordinator:  
**Prof. Giancarlo Ferrigno**



*Alle mie nonne*



# Acknowledgements

Questo lavoro è frutto di tre anni di attività presso il Dipartimento di Bioingegneria dell’Istituto di Ricerche Farmacologiche *Mario Negri*, sotto la guida del Prof. Andrea Remuzzi, a cui vanno la mia stima e la mia gratitudine, e in compagnia dell’Ing. Bogdan Ene-Iordache, di Abramo Anghileri e di Sergio Carminati, la mia *famiglia* lavorativa, insieme a tutti i ricercatori e collaboratori di Villa Camozzi. Desidero ringraziare il Dott. Lionello Caverni e il Prof. Gian Paolo Cornalba dell’Unità di Radiologia dell’Ospedale San Paolo di Milano per l’entusiasmo dimostrato durante la collaborazione, il Prof. Alessandro Veneziani per la fiducia che mi ha accordato, l’Ing. Alberto Redaelli per la disponibilità dimostrata, il Prof. David Steinman e la moglie Dolores per avermi offerto una grande occasione. Il lavoro è stato in parte finanziato da una borsa di studio messa a disposizione dalla Nando Peretti Foundation.

Sono debitore verso i miei genitori per questo risultato. A loro va la mia più profonda gratitudine. Ringrazio chi mi è stato vicino in questi tre anni e in particolare Carlotta, per avermi condiviso con i modelli tridimensionali.

This work is the result of three years of activity at the Bioengineering Department of *Mario Negri* Institute for Pharmacological Research, under the guide of Prof. Andrea Remuzzi, for whom I feel respect and gratitude, together with Eng. Bogdan Ene-Iordache, Abramo Anghileri and Sergio Carminati, my workplace *family*, and with all the researchers and collaborators in Villa Camozzi. I would like to thank Dr. Lionello Caverni and Prof. Gian Paolo Cornalba at the Radiology Unit of San Paolo Hospital in Milano for the enthusiasm demonstrated during our collaboration, Prof. Alessandro Veneziani for his confidence in me, Ing. Alberto Redaelli for being willing to help, Prof. David Steinman and his wife Dolores for offering me a great chance. The work has been in part supported by a fellowship from the Nando Peretti Foundation.

I owe much to my parents for this result. To them my deepest gratitude goes. I would like to thank the ones who were near during these three years and in particular Carlotta, for having shared me with three dimensional models.



# Abstract

Cardiovascular disease is the main cause of death in Western world and Japan. In particular, atherosclerosis has the greatest mortality and morbidity among vascular pathologies, being responsible for the majority of cardiac and cerebral strokes. Evidence that atherosclerotic plaque formation mainly takes place near bifurcations and bendings has led to the hypothesis that irregular hemodynamic conditions may play a role in initiation and progression of atherosclerotic lesions. In vivo and in vitro studies, both at the macroscopic and cellular level, confirmed these hypotheses, enforcing the importance of studying local hemodynamic conditions in atherosclerotic-prone arterial sites. In the meantime, three-dimensional angiographic techniques, such as computerized axial tomography (CT) and magnetic resonance (MR), provide detailed anatomic information with low invasiveness. Local hemodynamics can then be studied at patient-specific level using computational fluid-dynamics (CFD) applied to realistic geometric models of the vasculature. It is therefore important to solve the problem of accurately reconstructing geometric models from CT or MR angiography images in order to gain accuracy in CFD computations and predictions.

This work has the aim of presenting new tools developed for three-dimensional model reconstruction, geometric analysis, mesh generation and CFD calculation applied to large arteries. Computational geometry techniques have been developed and used to solve specific problems such as segmentation, reconstruction, editing, measurement and discretization of three-dimensional patient-specific models of vasculature. Particular attention was payed to minimizing operator intervention and maximizing the reproducibility and accuracy of the results, making it possible to introduce geometric and fluid-dynamics analyses of normal and diseased vessels at the clinical level.



# Contents

<b>Acknowledgements</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The role of hemodynamics in vascular disease . . . . .	1
1.2 Patient-specific characterization of local hemodynamics . . . . .	3
1.3 Aims and outline of the work . . . . .	4
<b>2 Three-dimensional modeling of vascular structures</b>	<b>7</b>
2.1 Introduction . . . . .	7
2.2 Image acquisition . . . . .	7
2.2.1 Computed tomography . . . . .	7
2.2.2 Magnetic resonance . . . . .	9
2.2.3 Ultrasounds . . . . .	10
2.2.4 Image representation and handling . . . . .	11
2.3 Contouring . . . . .	11
2.3.1 Marching Cubes algorithm . . . . .	11
2.3.2 Limitations of isosurface extraction . . . . .	14
2.4 Parametric deformable models . . . . .	16
2.4.1 Snakes . . . . .	16
2.4.2 Balloons . . . . .	17
2.5 Level sets . . . . .	19
2.5.1 Localization of the evolution equation for deformable surfaces . . . . .	19
2.5.2 Level sets evolution equation . . . . .	21
2.5.3 Numerical approximation . . . . .	22
2.5.4 Sparse-field approach . . . . .	24
2.5.5 Application to blood vessel 3D modeling . . . . .	26
2.6 Implementation . . . . .	30
2.7 Validation . . . . .	33

<b>3 Topological analysis of vascular structures</b>	<b>37</b>
3.1 Introduction . . . . .	37
3.2 Algorithm description . . . . .	37
3.3 Implementation . . . . .	43
3.4 Validation . . . . .	43
3.5 Applications . . . . .	44
3.5.1 Microvascular network analysis . . . . .	44
3.5.2 Single vessel level sets initialization . . . . .	45
3.5.3 Model decomposition for computational mesh generation . . . . .	46
<b>4 Three-dimensional editing of piecewise linear surfaces</b>	<b>47</b>
4.1 Introduction . . . . .	47
4.2 Weighted Laplacian filtering . . . . .	49
4.3 Bi-Laplacian mesh filtering . . . . .	52
4.3.1 Minimal energy surfaces . . . . .	53
4.3.2 Approximation of curvature on piecewise linear surfaces . . . . .	54
4.3.3 Numerical solution by bi-Laplacian factorization . . . . .	57
4.4 Applications . . . . .	59
4.4.1 Local smoothing and bulging removal . . . . .	59
4.4.2 Inlet and outlet extensions . . . . .	60
4.4.3 Virtual intervention . . . . .	61
4.5 Implementation . . . . .	63
4.6 Validation . . . . .	63
<b>5 Geometric characterization of vascular structures</b>	<b>67</b>
5.1 Introduction . . . . .	67
5.2 Medial axis and Voronoi diagram . . . . .	68
5.2.1 Medial axis of continuous objects . . . . .	69
5.2.2 Voronoi diagram and Delaunay tessellation . . . . .	70
5.2.3 Poles and stable Voronoi diagram subsets . . . . .	74
5.3 Centerline calculation . . . . .	77
5.3.1 Weighted geodesic formulation and the Eikonal equation . . . . .	77
5.3.2 Numerical solution . . . . .	80
5.3.3 Centerlines and vessel radius measurements . . . . .	87
5.4 Relating surface points to centerlines . . . . .	87
5.4.1 Geodesic distance to centerlines . . . . .	91
5.4.2 Euclidean distance to centerlines . . . . .	91
5.5 Normalized tangency deviation . . . . .	93
5.5.1 Definition . . . . .	93
5.5.2 Analytical expression for elliptical base cylinders . . . . .	96
5.5.3 Generalization to arbitrary shapes . . . . .	97
5.6 Implementation . . . . .	100
5.7 Validation . . . . .	100

<b>6 Mesh generation</b>	<b>103</b>
6.1 Introduction . . . . .	103
6.2 Hexahedral meshing of arterial bifurcations . . . . .	104
6.2.1 Model decomposition . . . . .	105
6.2.2 Hexahedral mesh generation with the Cooper algorithm .	106
6.3 Hybrid meshing of complex arterial segments with adaptive boundary layers . . . . .	113
6.3.1 Boundary layer generation with adaptive thickness wedges	114
6.3.2 Tetrahedral volume meshing by the advancing front approach	118
6.4 Implementation . . . . .	120
6.5 Validation . . . . .	120
<b>7 Computational hemodynamics</b>	<b>123</b>
7.1 Introduction . . . . .	123
7.2 Navier-Stokes equations . . . . .	123
7.3 Constitutive laws for blood rheology . . . . .	125
7.3.1 Newton's law . . . . .	125
7.3.2 Carreau's law . . . . .	125
7.4 Boundary conditions . . . . .	126
7.4.1 Average boundary conditions . . . . .	128
7.4.2 Patient-specific boundary conditions . . . . .	129
7.4.3 Boundary conditions from multiscale modeling of circulation	130
7.5 Finite-element approximation of Navier-Stokes equations . . . . .	131
7.5.1 Weak formulation of Navier-Stokes equations . . . . .	131
7.5.2 Spatial discretization . . . . .	131
7.5.3 Solution technique . . . . .	133
7.6 Post processing . . . . .	133
7.6.1 Wall shear stress and derived quantities . . . . .	134
7.6.2 Streaklines and Lagrangian particle tracking . . . . .	135
7.7 Applications . . . . .	137
7.7.1 Carotid bifurcation acquired by CT angiography . . . . .	137
7.7.2 Abdominal aorta acquired by MR angiography . . . . .	144
<b>8 Conclusions</b>	<b>151</b>
8.1 Summary of contributions . . . . .	151
8.2 Conclusions and future directions . . . . .	153
<b>A The Euler-Lagrange equation</b>	<b>157</b>
<b>B Element types</b>	<b>159</b>
B.1 2D elements . . . . .	159
B.2 3D elements . . . . .	161
B.3 Differentiation over elements . . . . .	164



# Chapter 1

## Introduction

### 1.1 The role of hemodynamics in vascular disease

Cardiovascular disease is the leading cause of death in the developed world, and atherosclerosis accounts for the majority of the cases. Furthermore, although mortality has slightly decreased in the past few years, morbidity has not, indicating that the improvement in diagnostic and therapeutic methods did not correspond to an improve in preventive measures [55]. A number of risk factors have been associated with atherosclerosis, such as genetic predisposition, hypertension, smoking, hyperlipidemia, diabetes mellitus, social stress, sedentary lifestyle, viral infection. Even though such factors have a systemic nature, atherosclerosis is a focal disease, atherosclerotic lesions being localized near bifurcations and bendings along the arterial tree. Supported by this evidence, the hypothesis on the involvement of local hemodynamic factors in the genesis and progression of atherosclerosis has been formulated and confirmed by several *in vivo* and *in vitro* experiments [67].

The vessel wall of large arteries is composed of three layers, the intima, the media and the adventitia, each with specific functions. The adventitia, the outer layer, is mainly composed by connective tissue, mostly collagen fibers with a longitudinal orientation, and provides mechanical support. The media is the thickest layer, composed by smooth muscle cells and connective tissue, mainly collagen and elastin, with both cells and matrix fibers oriented circumferentially, and its role is to provide vasomotility (this property is prominent in smaller arteries) and elastic compliance (this property is prominent in larger arteries). In normal subjects, the intima is composed almost exclusively by a monolayer of endothelial cells on a basement membrane, and is responsible for the antithrombogenic behaviour of the vascular wall, for vasoregulation (mainly through the production of vasodilator nitric oxide (NO) and vasoconstrictor enothelein-1 (ET-1) which bind to receptors on smooth muscle cells), control of smooth muscle cell proliferation and control of the transport of water, solutes, macromolecules and cells across the vessel wall [46].

The earliest evidence of atherosclerosis is constituted by fatty streaks, origi-

nated by accumulation of LDL, which at a certain extent cannot be counteracted by macrophage phagocytic activity. Localized fatty streaks have been observed even in children, and occur at the same sites as atherosclerotic lesions in older age. Atherosclerosis effectively manifests with the formation of well-defined lipid pools, which usually do not lead to vascular lumen narrowing thanks to compensatory enlargement effects (outward bulging of the thickened wall). Successive stages see the evolution of the lesions into fibrous plaques, made up of intimal smooth muscle cells, connective tissue, eventual calcified regions, and a variable amount of intra and extracellular lipids, contained into dense fibrous caps. Local hemodynamics can be greatly influenced by the resulting alterations in vascular wall morphology. At this stage, damage of the fibrous cap, probably due to stress concentration in the vicinity of calcifications or lipid cores, inflammatory cell infiltrates, surface defects with thrombus formation or intra-plaque hemorrhages may lead to plaque instability, associated with plaque ulceration, disruption, hemorrhage and thrombosis, major causes of acute ischemic events [10, 55, 98, 99].

During atherogenesis, the intima experiences a complex set of alterations, which include thinner endothelial glycocalyx, enhanced permeability, accumulation of albumin, fibrinogen, low-density lipoproteins (LDL), increased monocyte recruitment and higher endothelial cell turnover. Moreover, lesion-prone regions are characterized by polygonally-shaped endothelial cells, contrary to normal regions, where endothelial cells are organized along the main direction of flow. These results suggest a major involvement of local hemodynamics in atherogenesis. Due to migration and proliferation of smooth muscle cells from the media into the intima, intimal thickening occurs. Again, intimal thickness has been found to correlate with local hemodynamic conditions [31, 80].

In vitro evidence that endothelial cells react to local hemodynamic conditions has been largely provided in the last decade. Experiments performed on endothelial cell monolayers have shown that a large number of signal transduction mechanisms are activated when endothelial cells are exposed to flow [19, 46, 68]. In particular, shear stress seems to play a major role in endothelial cell mechanotransduction, as demonstrated in experiments in which shear rate was kept constant and shear stress was varied by changing fluid viscosity. Moreover, different flow conditions (laminar or turbulent, steady, transient or pulsatile) have been demonstrated to elicit different responses. In general, physiologic shear stress levels (assumed  $> 15$  dyne/cm $^2$ ) are associated with cytoskeletal rearrangement resulting in fusiform cell morphology aligned with the main direction of flow, vasodilative response, antioxidant enzymes production, cellular growth inhibition, low adhesion molecule exposure, antithrombotic behaviour and low proliferation/apoptosis turnover. A complementary response is obtained when low and oscillating levels of shear stress ( $\pm 0 - 4$  dyne/cm $^2$ ) are applied to endothelial cell cultures. As a result, physiologic shear stress values seem to induce an atheroprotective phenotype in endothelial cells [27, 33, 45, 49, 67, 122].

The results cited above suggest a model of atherogenesis induced by impaired hemodynamic conditions which exert low and oscillating shear stress on the vas-

cular wall. This hypothesis is sustained by the fact that the same low and oscillating shear stress patterns occur in typical plaque-prone anatomical sites, such as outer bifurcation walls (e.g. the carotid bulb) and tight bendings. This was demonstrated by numerous experimental studies performed on idealized or averaged models [58, 77, 78, 79, 137], or on casts of vascular specimens. Models were designed on the basis of anatomical data and realized in glass or Plexyglas, and flow velocimetry was performed either with laser Doppler anemometry or, more recently, with phase-contrast magnetic resonance. In this context, Ku et al. [58] found significant positive correlation between low and oscillating shear stress and wall thickness by measuring velocity using laser Doppler anemometry in a averaged Plexiglas model of the carotid bifurcation and intimal thickness on histological sections. With the development of numerical methods and the availability of computational resources experienced in the last decade, the experimental approach has been progressively replaced by the computational approach, even if it still represents a fundamental step for computational model validation [22, 86]. In general, computational fluid dynamics methods allow to investigate flow patterns, particle residence times and shear stress distributions at a great level of detail and to easily alter simulation parameters. Thanks to this, increasingly complex models of arterial segments have been generated and studied over the years [40, 50, 51, 63, 87, 88, 89, 117, 118, 119, 128], leading to a deeper understanding of the influence of geometry, flow waveforms, viscosity and wall distensibility on flow and shear stress patterns.

## 1.2 Patient-specific characterization of local hemodynamics

Experimental and computational studies performed on idealized and averaged models have led to the characterization of the behaviour of pulsatile flow in presence of bifurcations, anastomotic joints, side branches and curved vessels, and have shed more light on recirculation phenomena, such as that occurring at bifurcations, and secondary flow patterns developing in cross-sectional planes, frequently observed inside curved vessels. Those studies also evidenced that the geometric configuration of vascular tracts has a major role on the resulting flow patterns. Therefore, the necessity to characterize local hemodynamics inside realistic models of vasculature has been advanced. Moreover, non-invasiveness has become a major requisite, because it may allow to perform such investigation on a number of normal and pathologic real cases, and to follow the temporal evolution of the vascular pathology.

So far, no in-vivo non-invasive velocimetry techniques (such as echo-Doppler ultrasound, or phase-contrast magnetic resonance) allow to perform velocity measurements inside blood vessels of complex geometry with the resolution required by local hemodynamics characterization. These instruments can provide accurate measurements inside relatively straight segments under the assumption of devel-

oped flow. However, since the aim is the investigation complex flow patterns, computational approach is the most adequate tool to accomplish in-vivo non-invasive local hemodynamics characterization on a patient-specific basis.

Recent advances in medical imaging brought to the clinic a set of techniques, such as computed tomography (CT) and magnetic resonance (MR) which allow to acquire three-dimensional (3D) anatomy with high resolution and low invasiveness. On the basis of the acquired images, image processing methods offer the possibility to virtually reconstruct 3D models reproducing the geometry of the vascular wall. After the definition of proper boundary conditions, the reconstructed models can then be used to perform computational simulations of local hemodynamics. This approach has recently found concrete implementations and brought to interesting results [38, 125, 86, 110]. The first studies on small numbers of patients have evidenced the marked inter-subject anatomical variability and its influence on flow and shear stress patterns [76, 112, 120].

Although the computational fluid dynamics (CFD) problem is substantially unchanged with respect to idealized cases, *the real effort involved in image-based CFD modeling occurs well before the CFD simulation is initiated*, as recently pointed out by Steinman [110]. In fact, patient-specific vascular model reconstruction techniques rely on a set of techniques which imply low-level operator interaction, such as identification of vessel lumen on 2D images, reconstruction of the 3D model from the stack of 2D sections by means of CAD software, editing of the model, discretization of the volume into small elements for CFD. The resulting procedures are time-consuming and operator-dependent. Moreover, description of the 3D geometry of the reconstructed models, such as measurement of vessel axis and diameters, is not always univoque, and requires the operator to take arbitrary decisions. Recently, some groups have developed methods relying on more advanced 3D modeling techniques [26, 60, 127] which solve part of these problems. Even though full automation is difficult to achieve, given the variability of image source, image quality and vascular anatomy, robust techniques which respond to well-defined criteria and imply simple high-level interaction are required.

### 1.3 Aims and outline of the work

This work addresses the problem of patient-specific geometric and hemodynamic modeling of arterial segments from CT and MR angiography images. The aim is to provide a set of tools based on mathematically well defined criteria which maximize accuracy and reproducibility and minimize operator-dependent tasks, requiring simple high-level interaction. Justified by what pointed out in the previous section, much emphasis in this work is given to geometric reconstruction and characterization. Where possible, the techniques presented are introduced in variational form, which better describes the underlying criterion.

The work is organized as follows:

- In Chapter 2, 3D modeling of vascular structures from 3D angiography im-

ages is addressed. First a contouring algorithm is presented, along with the limitations of its direct use on angiographic images. Then deformable models are introduced. A brief description of parametric deformable models is followed by the extensive description of implicit deformable models, used in this work, from a mathematical and numerical point of view. Successively this technique is tailored to the problem of reconstruction of models of vasculature, and a fast initialization and single vessel evolution strategy is presented.

- Chapter 3 is dedicated to the description of a 3D topological analysis algorithm for tubular surfaces which provides an approximation of vessel centerlines. This algorithm is used twice throughout this work, once for fast interactive level sets initialization, and successively for automatic model decomposition during volume discretization.
- Chapter 4 addresses the problem of editing the surface obtained from 3D modeling for subsequent CFD analysis. This step is necessary when surface features due to unreconstructed side branches or image artifacts have to be removed, or when cylindrical extensions at the extremities of the model have to be created in order to let flow develop before it enters the realistic domain. Model editing is usually performed on a manual basis. We here propose two methods of increasing complexity and accuracy to perform this task, both based on variational criteria. The first is based on second-order weighted Laplacian filtering, and the second on the solution of a fourth-order bi-Laplace-Beltrami equation on the mesh, which allows mean surface normal continuity at the boundary of the edited region.
- In Chapter 5, geometric analysis of the reconstructed models is addressed. The problem is decomposed in the subproblems of finding vessel centerlines given the surface mesh, and successively relating surface mesh points to centerlines. Both subproblems are solved with the same mathematical techniques. First the concepts of medial axis and of its approximation for sampled surfaces are given. Centerlines are then defined with a variational criterion, and their computation, performed by efficiently solving a partial differential equation on the medial axis, is described. Based on the resolution of a similar equation, a robust method for the retrieval of the relationship between surface points and centerlines is then presented. Finally, a normalized quantity for scale-invariant characterization of surface features on vascular models is proposed.
- Chapter 6 is dedicated to the presentation of two techniques for the generation of computational meshes for the reconstructed models. The first generates hexahedral meshes for geometries containing single bifurcations, and it is based on automatic model decompositions and on the application of a sweeping scheme. The second is an hybrid meshing scheme, which has no

limitations on the geometry to be discretized. It is based on the geometric analysis technique presented in Chapter 5 and automatically produces adaptive-thickness boundary layers.

- In Chapter 7 the fluid dynamics problem is presented. The Navier-Stokes equations are briefly introduced, and two popular rheologic models for the blood are presented. The problem of boundary condition specification is then addressed. Successively, a sketch of the finite elements method and the resolution strategies for the Navier-Stokes equations is given. Post-processing techniques, tailored to the computation of relevant quantities for local hemodynamic characterization are then introduced, such as wall shear stress, oscillatory shear index, wall shear stress gradient and wall shear stress angle deviation. Lagrangian particle trajectory computation and visualization in pulsatile environments is also described. Finally an example of local hemodynamic characterization is presented for a patient-specific abdominal aorta model.
- Chapter 8 the original contributions brought by this work are summarized, and a few conclusions and future directions are drawn.
- In Appendix A the Euler-Lagrange equation, fundamental result of variational calculus frequently used in this work, is reminded.
- In Appendix B a brief description of the finite element types used in this work is given.

# **Chapter 2**

## **Three-dimensional modeling of vascular structures**

### **2.1 Introduction**

Angiographic image acquisition techniques provide detailed anatomic data on vascular structures. Graphic workstations usually linked to CT or MR scanners are used in the clinical practice to produce 3D patient-specific representations on the basis of the acquired data. The techniques used today for such purposes are in general not adequate for accurate geometric analysis and CFD computation which we address here, since the main effort is directed toward producing high-quality visual feedback rather than accurate geometric modeling of a particular anatomic structure. In our case, vascular modeling requires, as a first step, the extraction of vascular wall position from medical images. This is what this chapter concerns, first by introducing a well-known technique for polygonal surface construction from 3D images, which is also used in clinical visualization, and then more advanced techniques, such as parametric and implicit deformable models, which allow to construct 3D models of the desired accuracy.

### **2.2 Image acquisition**

We will first introduce the principal imaging techniques used in the clinical practice for the acquisition of 3D anatomy of vascular segments.

#### **2.2.1 Computed tomography**

Computed tomography (CT) is a technique for imaging cross-sections of a subject using series of X-ray measurements taken at different angles around the subject. The intensity of X-rays passing through the imaged body is attenuated according to the density of tissues encountered, so that the line integral of tissue density is measured. The source and the detector rotate around the subject and collect a row

of X-ray measurements for each rotation angle. In helical CT scanners, the subject is continuously moved through the plane of the rotating source and detector, so that an helical trajectory around the subject is generated. In multirow helical CT scanners, the actual state of the art scanners, the signal is acquired simultaneously from up to four rows, thus optimizing the trade-off between acquisition time, z-resolution and noise.

The signal resulting from the acquisition is called *sinogram*, and is represented as a series of images with detected attenuation on the x-axis and rotation angle on y-axis. The image is then reconstructed by solving the inverse Radon transform on the image grid. The reconstructed image contains attenuation values expressed by Hounsfield units (HU), for which water is conventionally represented by 0. Consequently, fat is associated with negative values, connective tissue by low positive values and calcium by high positive values. Scanners today available in the clinical context allow in-plane resolution < 0.5mm and slice thickness < 1mm of anatomical structures in a single breathhold.

For vascular imaging, radio-opaque contrast is injected intravenously. The HU values yielded by contrast agents are positive values between that of connective tissue and that of calcium. Therefore, an artery affected by atherosclerotic plaques will be surrounded by connective tissue values and low contrast (lipid pools) and high contrast (calcified plaques) regions (see Figure 2.1). Although widely employed in the clinical practice, CT angiography presents some disadvantages related to the X-ray dose absorbed by the patient and the contrast agents employed, which are iodinated and may lead to intolerance problems.



Figure 2.1: Contrast-enhanced CT angiography image of the carotid bifurcations of a patient affected by severe atherosclerosis.

### 2.2.2 Magnetic resonance

Magnetic resonance (MR) is based on the measurement of relaxation times of the net magnetization vectors induced in tissues when a magnetic field at a given frequency is applied. Net magnetization occurs because the magnetic momentum of nuclei whose Larmor frequency is that of the applied field tends to orientate along the direction of the field. In medical imaging the Larmor frequency of hydrogen nuclei is usually employed. When the magnetic field is ceased, the magnetic momentum of hydrogen nuclei return to equilibrium by local field inhomogeneities and by interaction with other relaxing nuclei, and the net magnetization returns to 0. Both relaxation processes are exponential, and their time constants (named  $T_1$  and  $T_2$ ), which are characteristic of different tissues, can be measured. As a result, images can reflect proton density,  $T_1$ ,  $T_2$  or a mixture of these quantities. Localization of the voxels from the decaying magnetization signal (or free induction decay, FID) is accomplished by additional magnetic fields, acting as gradients along the directions of imaging axes. Signal is collected in lines of data in the frequency domain (the k-space), and is then brought to the image domain by Fourier transform techniques. Several sequences of application of magnetic fields and gradients are available, which offer great flexibility in imaging different anatomical structures.

An advantage of MR over CT is that the imaging planes can be oriented by changing the direction of gradients. Moreover, potentially harmful effects of magnetic fields have not been demonstrated, so that MR is considered a non-invasive investigation. Recent developments of MR imaging techniques allow  $< 1\text{mm}$  resolutions within a single breathhold, although speed and resolution are lower than that obtained in CT imaging.

Various modalities are available for 3D MR vascular imaging. In time of flight (TOF) MR, flowing blood is excited in one plane and its signal acquired in a downstream plane. This allows to enhance saturated blood over the unexcited static tissue without using contrast agents, with the drawbacks of low spatial resolutions, slow acquisition and artifacts from complex flow. Conversely, black-blood techniques suppress the signal from flowing blood entering the imaged region, producing high resolution images in which the vascular lumen is black and the vessel wall can be identified. This is an ideal modality to non-invasively acquire vessel geometry and wall thickness at the same time. However, plaque-mimicking artifacts can occur in regions of slow flow. MR can take advantage from the use of contrast agents which modify the magnetic properties of blood. In particular, gadolinium (Gd-DTPA) shortens  $T_1$  of blood, resulting in fast high-contrast high-resolution acquisitions of vessel lumen, with image quality not depending on blood motion. The MR images employed in this work were obtained by Gd-DTPA enhanced MR (see Figure 2.2). The drawback of this method is that it is not completely non-invasive, although MR contrast agents are well tolerated compared to that used in CT angiography. Magnetic resonance can also measure velocity components non-invasively with phase contrast (PC) techniques, which rely on velocity induced phase shifts of the transverse magnetization. Since in the presence of complex flow patterns ve-

lacity measurements can be affected by artifacts, acquisition is usually performed in relatively straight vessels.



Figure 2.2: Maximum intensity projection of a Gd-DTPA enhanced MR angiography of the abdominal aorta of a 78-years-old patient.

### 2.2.3 Ultrasounds

Ultrasound imaging is based on the generation of ultrasounds from a piezoelectric transducer which is then used as a receiver for the waves reflected at the interfaces between two tissues with different acoustic impedance. The brightness modulation (B-mode) imaging modality produce morphologic images which represent the echogenicity of the tissues and of tissue interfaces. Recently, image processing techniques allow to combine acquired 2D images into volumes, yielding 3D ultrasound. Although image resolution still cannot compete with CT or MR angiography and the investigation is limited to superficial vessels, such acquisition method is attractive for the possibility of accurately discerning the morphology of the vessel wall, and for the relative inexpensiveness and flexibility of the technique.

A useful imaging modality is Doppler ultrasound, which accounts for the frequency shift of waves reflecting over flowing red blood cells. This way velocity measurements can be performed real-time with high temporal resolution in a sample volume.

#### 2.2.4 Image representation and handling

Independently from the image acquisition technique employed, the acquired 3D images, usually in the form of stacks of 2D images, are stored on workstations linked to the scanners, and must be transferred to calculators for processing. This has to be accomplished without loss of information. Since medical images are usually represented by a number of gray levels greater than 256, 8-bit image formats, such as TIFF, handled by usual image-editing software are not suited for the representation of medical data. Moreover, given the number of informations associated with acquisition, such as patient's and investigator's data, image number, image position, image resolution, acquisition time, acquisition modality and scan parameters, the need of a consistent way to handle such amounts of data has led to the definition of a standard for communication and storage of medical image data, DICOM (DIgital COmmunication in Medicine). Images stored in the DICOM format contain an header which contain a number of tags organized into groups, followed by image data, which for CT and MR is represented in signed 16-bit format. By reading the tags and the image data it is possible to reconstruct the acquired volume from the image stack without the need of further information. In the following we will denote the scalar field of gray levels sample on the 3D structured grid made by image voxels as  $I(\mathbf{x})$ , with  $\mathbf{x} \in \mathbb{R}^3$ .

### 2.3 Contouring

Since, as a first approximation, different tissues correspond to different gray levels on radiologic images, the simplest approach for the reconstruction of 3D models representing the vessel wall is the creation of a surface in correspondence of a given gray level of the acquired 3D image volume (see Figure 2.3). This technique is commonly known as contouring, and can be performed both on single 2D images constituting the 3D volume, and directly in 3D. In the following we give an overview of the most popular contouring algorithm, Marching Cubes, and we illustrate the limitations of the direct application of this algorithm to angiographic images for blood vessel reconstruction. Nevertheless, as shown in the following sections, Marching Cubes will be employed both for the initialization and as a final step of level sets-based reconstruction, our major instrument for blood vessel modeling.

#### 2.3.1 Marching Cubes algorithm

The Marching Cubes algorithm has been proposed by Lorensen et al. [66] to solve the problem of constructing a polygonal surface located on an interface of a scalar field sampled on a regular grid, such as a 2D or 3D image. Its first formulation traces back to 1987, nevertheless this algorithm is still commonly used in a variety of fields, and represents the gold standard against which several improvements [11] or alternatives [132] have been validated.

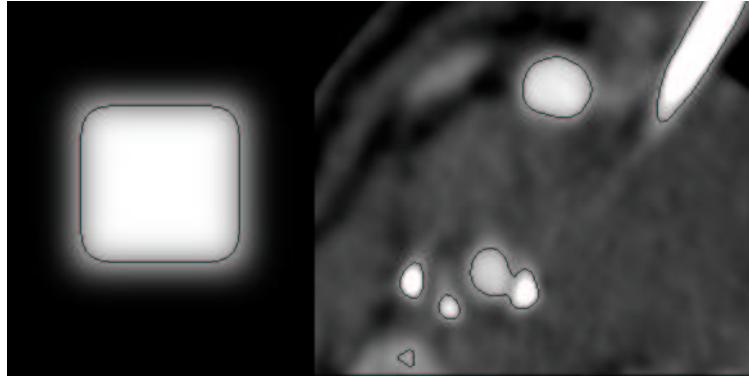


Figure 2.3: Examples of contouring on a synthetic image (left) and on a angio-CT image. The contouring levels were chosen manually.

We here give a brief description of the Marching Cubes algorithm in 3D. Given a 3D image and a level corresponding to the isovalue of interest, it is possible to label image grid vertices, the voxels, as *above* or *below* if their scalar value is higher or lower than the specified contouring level, respectively. If we consider the set of one voxel-wide cubes defined by eight neighboring voxels, we can identify the cubes intersected by the isosurface of interest as the ones in which some of the voxels are labeled *above* and the rest *below*. The base concept of the Marching Cubes algorithm is that the *above* and *below* voxels of each cube can be partitioned by a set of triangles whose vertices lie on cube edges in a finite number of ways, called *cases*. Therefore a table of cases can be constructed, containing all topological configurations of *above* and *below* voxels and triangles partitioning them, regardless the exact position of triangle vertices along cube edges (as in Figure 2.4, which only has a topological meaning. Triangle vertices must be considered in a generic position along cube edges). Since in most cases (see further) there is only one configuration of surface triangles for each of the 64 combinations of *above* and *below* voxels, surface construction can be performed independently for each cube constructed in the contoured image, retrieving the proper surface configuration from the case table. The exact position of triangle vertices is successively computed by linear interpolation of voxel scalar values on cube vertices. The surface produced by marching through the whole image volume is therefore first-order sub-voxel accurate.

Figure 2.4 shows the table of cases reduced to 15 by symmetry. For a small number of cases, namely 3, 6, 7, 10, 12, 13 there is more than one possibility of constructing a surface partitioning *above* and *below* voxels, as shown in Figure 2.5 as *complementary* cases. The complementary cases are generated from the base case by swapping *above* and *below* voxels. This possibility generates ambiguity in surface construction, since an arbitrary choice among ambiguous cases can give rise to changes in surface topology (i.e. holes). The ambiguity can be overcome

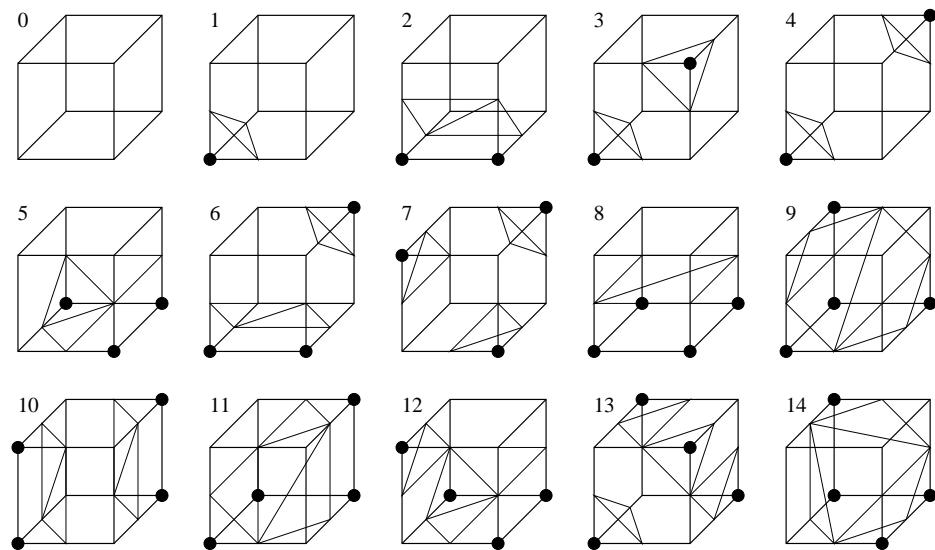


Figure 2.4: Marching cubes cases.

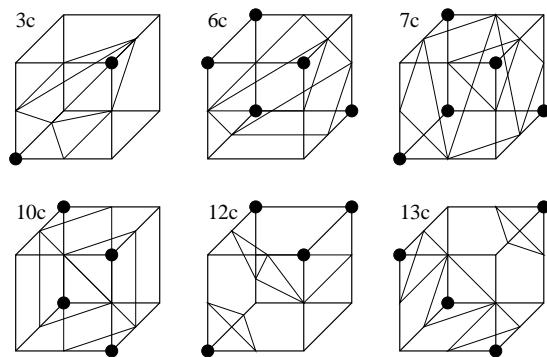


Figure 2.5: Complementary marching cubes cases.

with the introduction of a set of rules which select the proper case according to the configuration of surrounding cubes. From a theoretical point of view, the rules can be avoided if contouring problem statement, besides image and contouring level, also includes the definition of different connectivity neighborhoods for *above* and *below* voxels, so that *above* voxels that are considered connected would not necessarily be considered connected if labeled *below*. This way *above/below* swapping does not produce indistinguishable configurations in terms of cube voxel partitioning, therefore resolving ambiguity. In practice, it is not always possible to choose the proper neighborhoods for the whole 3D image prior to constructing the isosurface, so the use of local rules is often preferred.

### 2.3.2 Limitations of isosurface extraction

Since angiographic 3D images represent scalar fields sampled on regular grids, it is possible to apply Marching Cubes algorithm to construct the isosurface located over the transition from vessel lumen to the surrounding tissue. Thanks to its simplicity, this approach is currently implemented in software applications for radiological visualization. But while for the purpose of visualization this approach produces adequate results, for accurate geometric and CFD analysis it shows some limitations. The major limitation is that resulting surface depends on the contouring level chosen, and this choice is normally left to the operator. Given the variability in absolute gray-levels, which in general depends on acquisition modalities and on physical features of the patient, it is not possible to define a fixed scalar level for vessel lumen boundary. Moreover, since image gray level varies more near interfaces, a slight change in contouring level may produce great changes in geometric and even topological features of the resulting surface, as shown in Figure 2.6, where models are generated by contouring at different levels and by discarding portions of surface not connected to the bifurcation. Even a single contouring level could also not be suitable for an angiographic acquisition whenever contrast medium is used to opacize vessels, because, due to blood flow, sections acquired in different times may exhibit different opacification levels due to variation in time of contrast medium concentration. The same problem can be experienced in time of flight and phase contrast MR angiographic acquisitions, due to complex blood flow patterns altering the received signal.

One more issue involving isosurface extraction is the inability of selectively reconstruct vessels of interest ignoring branching vessels, or to avoid the reconstruction of calcified plaques on CT images, since they have a higher gray value than the contrast medium, without manually editing the source images. This task can introduce operator dependency if the edited regions extend over several serial images and involve geometrically complex regions.

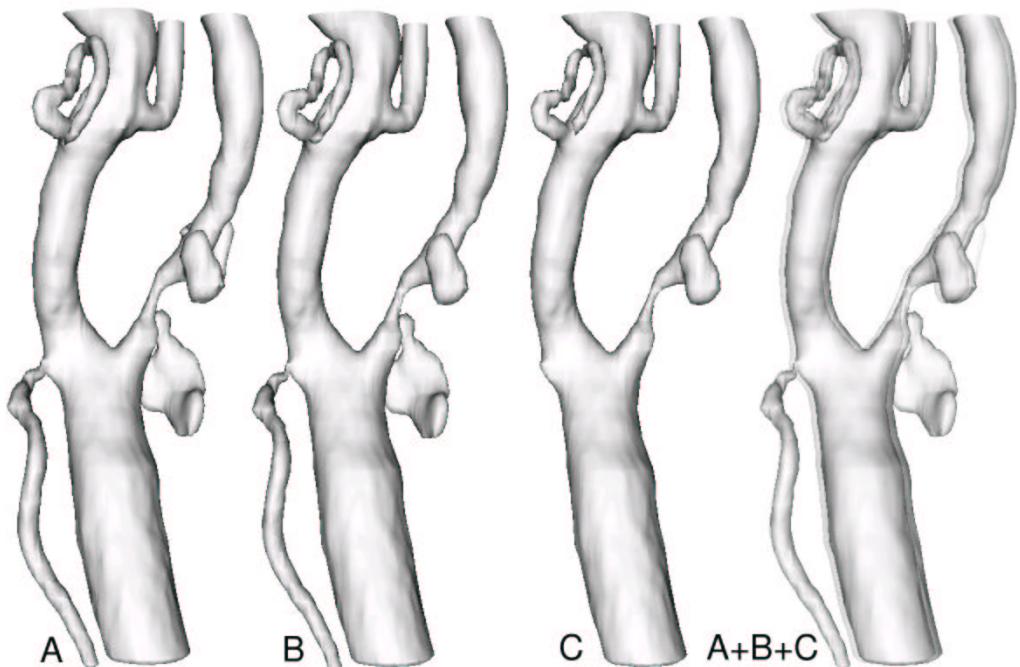


Figure 2.6: Change in surface position and topology due to a slight change in contouring level for a model of carotid bifurcation. Contouring levels: A: 215 HU; B: 225 HU; C: 235 HU. A+B+C shows the superimposition of the three models.

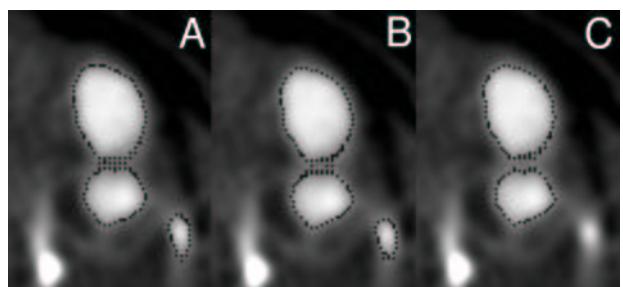


Figure 2.7: Model points plotted on the CT images at the level of the carotid bifurcation apex for three different contouring levels (A: 215 HU; B: 225 HU; C: 235 HU), showing a change in apex position, and a plaque being disconnected.

## 2.4 Parametric deformable models

The lack of an absolute correspondence between tissues and gray levels in radiologic images leads to the need of constructing models based on image features, rather than on absolute gray level values. In this direction we now present parametric deformable models, which are models evolving in the image space and whose deformations are described from the Lagrangian point of view, for which deformations are referred to the undeformed configuration, so that material points are followed during evolution (the same approach is adopted in solid mechanics) [73].

### 2.4.1 Snakes

Active contours, also known as snakes, constitute a commonly employed solution in the field of 2D image segmentation and shape retrieval [134]. A snake is a parameterized curve evolving on the basis of image features and internal constraints. Its description, in a Lagrangian frame, can be given as a function

$$\mathbf{C} = \underset{s}{U} \times \underset{t}{\mathbb{R}^+} \rightarrow \mathbb{R}^2 \quad (2.1)$$

where  $U \subset \mathbb{R}$  (e.g.  $U = [0, 1]$ ) and  $s \in U$  is curve parameterization. The evolution equation for snakes can be derived from the minimization of an energy functional

$$\mathcal{E}_{\text{snake}}(\mathbf{C}) = \mathcal{E}_{\text{smooth}}(\mathbf{C}) + \mathcal{E}_{\text{image}}(\mathbf{C}) \quad (2.2)$$

where

$$\mathcal{E}_{\text{smooth}}(\mathbf{C}) = \int_0^1 [w_1 |\mathbf{C}_s|^2 + w_2 |\mathbf{C}_{ss}|^2] ds \quad (2.3)$$

is internal deformation energy (the first term controls snake tension, having the effect of reducing snake length, and the second term controls snake rigidity, producing a smoothing effect), and

$$\mathcal{E}_{\text{image}}(\mathbf{C}) = \int_0^1 w_3 P(\mathbf{C}) ds \quad (2.4)$$

is driving energy,  $P(\mathbf{x})$  being a scalar potential function taking into account image features (e.g.  $P(\mathbf{x}) = -|\nabla I(\mathbf{x})|$ , where  $I(\mathbf{x})$  is the scalar field of image gray values) and user-defined landmarks.

In order to minimize the functional  $\mathcal{E}(\mathbf{C})$ , the active contour  $\mathbf{C}(s)$  must satisfy the Euler-Lagrange equation (see Appendix A):

$$-\frac{w_1 \partial \mathbf{C}_s}{\partial s} + w_2 \frac{\partial^2 \mathbf{C}_{ss}}{\partial s^2} + w_3 \nabla P(\mathbf{C}) = 0 \quad (2.5)$$

Instead of using a energy gradient descent algorithm to find a configuration which minimizes  $\mathcal{E}_{\text{snake}}$ , it is possible to adopt the equivalent strategy of making the snake

evolve in time toward a configuration satisfying the Euler-Lagrange equation (see Appendix A), resulting in the evolution equation

$$\frac{\partial \mathbf{C}}{\partial t} = w_1 \mathbf{C}_{ss} - w_2 \mathbf{C}_{ssss} - w_3 \nabla P(\mathbf{C}) \quad (2.6)$$

It is worth to note that if only the energy term  $\mathcal{E}_{image}(\mathbf{C})$  is considered in Equation 2.2, the associated Euler-Lagrange equation states that the snake stops on stationary points of the scalar field  $P(\mathbf{x})$ .

Since in 2D any closed curve shrinking under the effect of its curvature evolves into a circle before collapsing into one point, snakes are usually initialized as closed lines surrounding regions of interest. They then move by shrinking until driving energy makes them approach and stop on desired features. Depending on problem requirements, snakes can also be initialized as small circles located inside the areas of interest and inflated until image features are encountered (this strategy is more popular in blood vessel modeling).

Active contours have been widely used for 3D vessel reconstruction. The strategy is to extract vessel contours from single 2D images constituting the 3D acquired volume and then to loft the obtained profiles to reconstruct the 3D vessel surface. Assuming that vessel wall is located over the locus of maximal gray level transition from the inside to the outside of the vessel lumen, the presented evolution equation can be used to obtain a set of 2D contours located over gray level gradient magnitude ridges. In fact, the term  $-w_3 \nabla P(\mathbf{x}) = w_3 \nabla |\nabla I(\mathbf{x})|$ , where  $I(\mathbf{x})$  can be substituted with its smoothed version in order to select the proper feature scale, is a vector field pointing toward vessel wall from either sides of the wall.

The use of snakes for vessel modeling can take advantage from their simplicity and the high control on the results, as demonstrated by the results of several groups [65, 75, 96], but it suffers from some limitations. In fact, segmentation of 3D structures performed on 2D images can give rise to problems in correct shape retrieval, since image slice direction is in general not aligned with vessel axis (although this is true in some cases, such as for the main branches of the carotid bifurcation) and partial volume effects can take place. Moreover, initialization, and sometimes evolution, is controlled by an operator, making it time demanding to process a whole stack of 50 to 150 images containing information about complex 3D branching vessels. Finally, constructing a parametric surface (making use of NURBS or polygonal primitives) from a set of 2D contours is not always a trivial task, especially in correspondence of bifurcations.

#### 2.4.2 Balloons

The 3D counterparts of snakes are commonly known as balloons, which are parametric deformable surfaces which can be represented as

$$\mathbf{S} : U_r \times U_s \times \mathbb{R}_t^+ \rightarrow \mathbb{R}^3 \quad (2.7)$$

where  $U \subset \mathbb{R}$  and  $r, s \in U$  is surface parameterization, which we will assume orthonormal, so that

$$\begin{aligned} |\mathbf{S}_r| &= |\mathbf{S}_s| = 1 \\ \mathbf{S}_r \cdot \mathbf{S}_s &= 0 \end{aligned} \quad (2.8)$$

In 3D, conversely to 2D case, it is not true that a surface shrinking under the effect of its curvature always evolves into a sphere before collapsing into one point. The lack of such a result on the regularity of surface evolution under curvature makes it usually more convenient to inflate the 3D surface making use of an internal pressure term rather than making it shrink under curvature. This approach is also convenient for 3D initialization, since it only requires the specification of the center and the radius of a sphere located inside the 3D object to reconstruct.

The evolution equation for balloons which we now present incorporates one more term, which takes into account image driven balloon inflation, resulting in The energy functional we now present for balloons contains a new term which takes into account balloon inflation

$$\mathcal{E}_{\text{balloon}}(\mathbf{S}) = \mathcal{E}_{\text{infl}}(\mathbf{S}) + \mathcal{E}_{\text{smooth}}(\mathbf{S}) + \mathcal{E}_{\text{image}}(\mathbf{S}) \quad (2.9)$$

The corresponding evolution equation is

$$\frac{\partial \mathbf{S}}{\partial t} = w_1 G(\mathbf{S}) \mathbf{N} + w_2 (\mathbf{S}_{rr} + \mathbf{S}_{ss}) - w_3 \nabla P(\mathbf{S}) \quad (2.10)$$

where  $G(\mathbf{S})$  is scalar inflation speed, which can be constant or depending on image features,  $\mathbf{N}$  is outward surface normal and  $(\mathbf{S}_{rr} + \mathbf{S}_{ss})$  is an average second-order smoothing term, which can be substituted with more complex regularizing expressions to take into account the expected shape of the result (e.g. cylinders). Again, the evolution equation is solved making use of numerical methods, such as finite elements or simpler methods, which describe the deformation of the initial model through its parameterization [24, 29, 32, 71].

The use of balloons in vessel modeling [60] makes it possible to directly deal with 3D geometry without processing single 2D images, gaining in speed and operator independence, even if user interaction is more difficult. In [43, 81, 135] deformation takes place constrained to a generalized cylindrical coordinate system, rather than in the classic image Cartesian reference frame.

The major drawback of dealing with balloons is the parametric nature of their description, which can introduce constraints on their evolution. In fact, reparameterization has to be performed whenever great deformations take place, because excessive surface distortion gives rise to interpolation problems and prevents the balloon from catching the geometry of smaller structures. Moreover, topological constraints are introduced due to parameterization. Every change in surface genus, such that involved morphing a sphere into a torus, requires a reparametrization. In practice this means that merging of two colliding balloon portions cannot be described within the parameter space, so that *ad hoc* merging rules have to be introduced [32, 72].

## 2.5 Level sets

An alternative to parametric deformable models is represented by implicit models, which are scalar functions defined in  $\mathbb{R}^2$  or  $\mathbb{R}^3$  whose isosurface of level  $k$  is the model of interest. This change of perspective allows to overcome the limitations on topology and great deformations affecting parametric deformable models. We here give a sketch of the embedding process which allows to switch from the Lagrangian (used in solid mechanics) to the Eulerian (used in fluid mechanics) perspective in model description, in which deformations are referred to the deformed configuration. In practice, instead of describing the evolution of individual material surface points, surface motion is observed through the evolution of scalar field values at fixed points in space. In this context we introduce the *level sets* technique [106], which allows to deal with evolution equations for deformable implicit models, we give some numerical and implementation details and introduce the application of level sets to vessel modeling. In the following we will concentrate on the 3D case, but the same equations can be derived for a different number of dimensions. The independence of the final equations from the number of dimensions constitutes one of the several advantages of level sets over parametric deformable models.

### 2.5.1 Localization of the evolution equation for deformable surfaces

A surface evolving in time  $\mathbf{S} : \mathbb{R}^2 \times \mathbb{R}^+ \rightarrow \mathbb{R}^3$  can be represented as an isosurface of level  $k$ , or  $k$  level set, of a time-dependent scalar function  $F : \mathbb{R}^3 \times \mathbb{R}^+ \rightarrow \mathbb{R}$ , so that

$$\mathbf{S}(t) = \{\mathbf{x} | F(\mathbf{x}, t) = k\} \quad (2.11)$$

Since  $\mathbf{S}$  remains the  $k$  level set of  $F$  over time,

$$\begin{aligned} \frac{\partial F(\mathbf{S}, t)}{\partial t} &= -\nabla F(\mathbf{S}, t) \cdot \frac{\partial \mathbf{S}}{\partial t} \\ &= -|\nabla F(\mathbf{S}, t)| \frac{\partial \mathbf{S}}{\partial t} \cdot \mathbf{N} \end{aligned} \quad (2.12)$$

where  $\mathbf{N} = \frac{\nabla F}{|\nabla F|}$  is the outward normal to level sets, supposed that the embedding function has lower values inside and higher values outside the model.

Equation 2.11 represents the embedding of an evolution equation for a parametric deformable surface  $\mathbf{S}(r, s, t)$ , such as Equation 2.10, into the evolution equation of a scalar function  $F(\mathbf{x}, t)$  whose  $k$  level set is  $\mathbf{S}(t)$  [129]. The great advantage of this approach is that the description of the embedded version of  $\mathbf{S}(t)$  does not require a global parameterization, previously expressed through parameters  $(r, s)$ , but relies only on local geometric properties of  $F(\mathbf{x}, t)$ . The last step required to achieve localization is therefore to express the embedded version of the right-hand side of Equation 2.10 in terms of differential expression on  $F(\mathbf{x}, t)$ . In the following we will assume that the  $(r, s)$  parameterization of surface  $\mathbf{S}(r, s, t)$  is orthonormal, i.e. follows Equation 2.8.

The embedding of the inflation term  $w_1 G(\mathbf{S}) \mathbf{N}$  in Equation 2.10 can be rearranged as following

$$|\nabla F| w_1 G(\mathbf{x}) \mathbf{N} \cdot \mathbf{N} = w_1 G(\mathbf{x}) |\nabla F| \quad (2.13)$$

The embedding of the smoothing term  $(\mathbf{S}_{rr} + \mathbf{S}_{ss})$  requires to express  $\mathbf{S}_{rr}$  (and  $\mathbf{S}_{ss}$ ) in terms of differential expressions on  $F(\mathbf{x}, t)$ . In order to do this, we first compute the derivative of  $F(\mathbf{x}, t)$  with respect to  $r$ :

$$\frac{\partial F}{\partial r} = \nabla F \cdot \mathbf{S}_r = 0 \quad (2.14)$$

where  $|\mathbf{S}_r| = 1$  since we assumed surface parameterization to be orthonormal. This quantity is zero because  $\mathbf{S}(r, s, t)$  is a constant level set of  $F(\mathbf{x}, t)$ , therefore  $F(\mathbf{x}, t)$  doesn't vary with  $r$ . We then perform one more derivative on  $r$  from Equation 2.14, obtaining

$$\begin{aligned} \frac{\partial^2 F}{\partial r^2} &= [H_F(\mathbf{x}) \mathbf{S}_r] \cdot \mathbf{S}_r + \nabla F \cdot \mathbf{S}_{rr} \\ &= F_{uu} + \nabla F \cdot \mathbf{S}_{rr} = 0 \end{aligned} \quad (2.15)$$

where  $H_F(\mathbf{x})$  is the Hessian matrix of  $F$ , and  $F_{uu}$  is the second directional derivative of  $F$  in direction  $\mathbf{S}_r$ . The same expression can be written for  $\frac{\partial^2 F}{\partial s^2}$  substituting  $F_{uu}$  with  $F_{vv}$ , second directional derivative of  $F$  in direction  $\mathbf{S}_s$ . The smoothing term  $(\mathbf{S}_{rr} + \mathbf{S}_{ss})$  can then finally be written as

$$\begin{aligned} |\nabla F| w_2 (\mathbf{S}_{rr} + \mathbf{S}_{ss}) \cdot \mathbf{N} &= -w_2 (F_{uu} + F_{vv}) \\ &= -2w_2 H(\mathbf{x}) |\nabla F| \end{aligned} \quad (2.16)$$

where  $H(\mathbf{x})$  is the mean curvature of level sets<sup>1</sup>, which is invariant to rotations of the  $(u, v)$  coordinate system within the tangent plane. To justify the last equality in Equation 2.16, we start from the definition of mean curvature, always under the assumption of orthonormal parameterization, which provides  $\mathbf{S}_r \cdot \mathbf{S}_s = 0$ :

$$\begin{aligned} H(\mathbf{x}) &= \nabla \cdot \left( \frac{\nabla F}{|\nabla F|} \right) = \frac{(F_{xx} + F_{yy} + F_{zz}) |\nabla F| - (\nabla F_w \cdot \mathbf{N}) |\nabla F|}{|\nabla F|^2} \\ &= \frac{\nabla \cdot \nabla F - F_{ww}}{|\nabla F|} = \frac{1}{|\nabla F|} (F_{uu} + F_{vv}) \end{aligned} \quad (2.17)$$

where  $x, y, z$  is a generic orthonormal reference frame and  $u, v, w$  is the reference frame relative to the plane tangent to the level sets (i.e.  $F_w = |\nabla F|$ ). It follows

---

<sup>1</sup>The mean curvature of a surface at one point is defined as the mean value of the normal curvatures at that point, which in turn are the curvatures of the curve defined by the intersection of the surface with a plane perpendicular to each direction belonging to the tangent plane and passing in that point.

from the third equality in Equation 2.17 that  $H(\mathbf{x})$  can be computed from the second order differential structure of  $F$ , regardless of the particular coordinate system adopted.

The embedding of the image driving term,  $w_3 \nabla P(\mathbf{S})$ , is straightforward, resulting in

$$|\nabla F| w_3 \nabla P(\mathbf{x}) \cdot \mathbf{N} = w_3 \nabla P(\mathbf{x}) \cdot \nabla F \quad (2.18)$$

### 2.5.2 Level sets evolution equation

The localized level sets equation for  $F(\mathbf{x}, t)$  is finally obtained by substituting the embedded versions of the evolution terms into Equation 2.12, yielding

$$\frac{\partial F(\mathbf{x}, t)}{\partial t} = -w_1 G(\mathbf{x}) |\nabla F| + 2w_2 H(\mathbf{x}) |\nabla F| + w_3 \nabla P(\mathbf{x}) \cdot \nabla F \quad (2.19)$$

This equation describes a deformable surface, such as a balloon following Equation 2.10, embedded as a level set of a scalar field evolving in time. As previously noted, the great advantage of using the model described in Equation 2.19, rather than a balloon model, is the lack of parameterization in level sets formulation. The model can therefore freely deform and change its topology without the need of any particular care, such as reparameterization strategies or *ad hoc* merging rules. Moreover, when dealing with 3D images, Equation 2.19 can be directly solved on the image regular grid by one of classic numerical methods, such as finite differences (see further).

Image-based evolution terms (i.e.  $G(\mathbf{x})$ , inflation speed, and  $P(\mathbf{x})$ , attraction potential) can be defined as following

$$G(\mathbf{x}) = \frac{1}{1 + |\nabla I(\mathbf{x})|} \quad (2.20)$$

so that inflation speed is lower when image gradient is higher, such as near image features, and

$$P(\mathbf{x}) = -|\nabla I(\mathbf{x})| \quad (2.21)$$

which gives origin to the valleys of the attraction potential. These definitions were proposed by Malladi [69] in the context of medical imaging, and are well suited for 3D modeling of blood vessels from 3D angiographic images. Eventually image  $I(\mathbf{x})$  in both  $G(\mathbf{x})$  and  $P(\mathbf{x})$  can be substituted by  $G_\sigma * I(\mathbf{x})$ , i.e. the convolution of the image with a Gaussian kernel of proper scale, in order to smooth out image features of finer scales than the ones of interest.

It is possible to adopt a slightly different formulation of Equation 2.19

$$\frac{\partial F(\mathbf{x}, t)}{\partial t} = -w_1 G(\mathbf{x}) |\nabla F| + 2w_2 G(\mathbf{x}) H(\mathbf{x}) |\nabla F| + w_3 \nabla P(\mathbf{x}) \cdot \nabla F \quad (2.22)$$

in which the curvature term is weighted by function  $G(\mathbf{x})$ , so that the smoothing effect is stronger in regions of lower image gradient magnitude, where less image features are present, while more surface details are allowed where  $|\nabla I(\mathbf{x})|$  is higher.

To complete the framework, initialization of the embedding function  $F(\mathbf{x}, t_0)$  is required. If we are interested in embedding surface  $\mathbf{S}(t)$  in the 0 level set of function  $F(\mathbf{x}, t)$ , at time step  $t_0$ ,  $F(\mathbf{x}, t_0)$  can be modeled as the *signed distance function*

$$F(\mathbf{x}, \bar{t}) = \begin{cases} -D_{\mathbf{S}(t_0)}(\mathbf{x}) & \text{if } \mathbf{x} \text{ inside } \mathbf{S}(t_0) \\ D_{\mathbf{S}(t_0)}(\mathbf{x}) & \text{if } \mathbf{x} \text{ outside } \mathbf{S}(t_0) \end{cases} \quad (2.23)$$

where  $D_{\mathbf{S}(t_0)}(\mathbf{x}) = \min\{|\mathbf{x} - \mathbf{S}(t_0)|\}$  is the distance function from  $\mathbf{S}(t_0)$  and  $\mathbf{S}(t_0)$  is the embedded surface defined as in Equation 2.11 with  $k = 0$ .

### 2.5.3 Numerical approximation

In order to solve Equation 2.22, several numerical methods are available. For the means of blood vessel model reconstruction from 3D images, one possibility is to numerically solve Equation 2.22 by finite difference method, using image domain as structured grid for problem discretization. In order to apply finite differences to level sets equation, some numerical considerations have to be introduced.

Equation 2.22 belongs to the class of Hamilton-Jacobi equations, which has the following general form [106]:

$$\frac{\partial F(\mathbf{x}, t)}{\partial t} + \mathcal{H}(\mathbf{x}, \nabla F(\mathbf{x}, t)) = 0 \quad (2.24)$$

The Hamilton-Jacobi equation specialized for level sets is

$$\frac{\partial F(\mathbf{x}, t)}{\partial t} = G(\mathbf{x}, t) |\nabla F(\mathbf{x}, t)| \quad (2.25)$$

which is a class of nonlinear hyperbolic partial differential equations. The forward scheme for the numeric approximation of Equation 2.25 is

$$\begin{aligned} F(\mathbf{x}, t + \Delta t) &= F(\mathbf{x}, t) + \frac{\partial F(\mathbf{x}, t)}{\partial t} \Delta t \\ &= F(\mathbf{x}, t) + G(\mathbf{x}, t) |\nabla F(\mathbf{x}, t)| \Delta t \end{aligned} \quad (2.26)$$

Due to the particular behavior of hyperbolic equations, for which information only flows from the direction of front advancement, the use of central two-sided finite differences leads to instabilities for the scheme of Equation 2.26. For example, a surface through which  $F(\mathbf{x})$  is non-differentiable may be produced after merging of two level set fronts moving along incident directions have occurred. Approximating  $\nabla F(\mathbf{x})$  by central differences in this region produces incorrect results despite arbitrary grid refinement, because information from each side of first-order discontinuity is averaged in calculating gradient value. This inaccuracy can then propagate to the neighboring grid points giving rise to instable behavior of the numerical solution. To overcome this problem, Sethian et al. proposed the use of upwind (one-sided) finite differences for the approximation of odd-order derivatives. Central finite difference expression for the first derivative of  $F(\mathbf{x})$  in grid

point  $(i, j, k)$  on a grid of spacing  $h$  can be derived from Taylor expansion of  $F(\mathbf{x})$  around  $\mathbf{x}_{ijk}$  up to the first order, resulting in

$$F_x|_{\mathbf{x}_{ijk}} \approx D_{ijk}^0 = \frac{F(i+1, j, k) - F(i-1, j, k)}{2h} \quad (2.27)$$

This expression results to be second-order accurate, but it incorporates contributions from both positive and negative directions. In order to take into account only upstream contributions, upwind finite differences have to be introduced. Upwind finite difference expressions for  $F_x|_{\mathbf{x}_{ijk}}$  are<sup>2</sup>

$$\begin{aligned} F_x|_{\mathbf{x}_{ijk}} \approx D_{ijk}^{+x} &= \frac{F(i+1, j, k) - F(i, j, k)}{h} \\ F_x|_{\mathbf{x}_{ijk}} \approx D_{ijk}^{-x} &= \frac{F(i, j, k) - F(i-1, j, k)}{h} \end{aligned} \quad (2.28)$$

Though these expressions are only first-order accurate, they are capable to correctly approximate first derivatives in regions where solution present cusps. A drawback of the use of one-sided derivatives is that of adding numerical viscosity to the hyperbolic equation to be solved. The solution always converges to the exact solution for grid spacing  $h \rightarrow 0$ , but convergence is slower [94]. Upwind schemes of higher-order of accuracy tend to reduce this effect.

Using upwind finite difference for the level set approximation leads to

$$F(\mathbf{x}, t + \Delta t) = F(\mathbf{x}, t) - [\max(G(\mathbf{x}), 0)\nabla^+ + \min(G(\mathbf{x}), 0)\nabla^-]\Delta t \quad (2.29)$$

where

$$\begin{aligned} \nabla^+ = [ & \max(D_{ijk}^{-x}, 0)^2 + \min(D_{ijk}^{+x}, 0)^2 + \\ & \max(D_{ijk}^{-y}, 0)^2 + \min(D_{ijk}^{+y}, 0)^2 + \\ & \max(D_{ijk}^{-z}, 0)^2 + \min(D_{ijk}^{+z}, 0)^2 ]^{1/2} \end{aligned} \quad (2.30)$$

$$\begin{aligned} \nabla^- = [ & \max(D_{ijk}^{+x}, 0)^2 + \min(D_{ijk}^{-x}, 0)^2 + \\ & \max(D_{ijk}^{+y}, 0)^2 + \min(D_{ijk}^{-y}, 0)^2 + \\ & \max(D_{ijk}^{+z}, 0)^2 + \min(D_{ijk}^{-z}, 0)^2 ]^{1/2} \end{aligned} \quad (2.31)$$

It is possible to demonstrate that this scheme yields stable *viscosity* solutions to Equation 2.25 [106].

Besides first-order derivatives involved in the computation of quantities related to  $\nabla F(\mathbf{x}, t)$ , Equation 2.22 also requires the computation of second-order derivatives for the approximation of level sets curvature  $H(\mathbf{x})$  for the smoothing term. Given the diffusive nature of second-order derivatives, the use of central finite differences for their approximation is adequate.

---

<sup>2</sup>The extension of Equation 2.28 to the  $y, z$  axes is straightforward.

One last numerical issue concerns discretization in time. Since forward time update in Equation 2.26 is not unconditionally stable,  $\Delta t$  is required to be equal or lower than a limit time step in order to ensure stability. The limit time step for explicit time integration is given by the CFL [94] condition, which ensures that the rate of change in the solution doesn't exceed the maximum velocity which can be modeled with a given grid of spacing  $h$ , i.e.

$$\max \left( \frac{\partial F}{\partial t} \right) \leq \frac{h}{\Delta t} \quad (2.32)$$

Substituting left-hand side with Equation 2.25 yields a condition on  $\Delta t$ :

$$\Delta t \leq \frac{h}{\max(G(\mathbf{x}, t)|\nabla F(\mathbf{x}, t)|)} \quad (2.33)$$

#### 2.5.4 Sparse-field approach

Embedding parametric deformable surface into a scalar function allows to drop global parameterization, achieving topology independence and great deformations, but the price to pay is that a function defined on the whole image volume has to be described, instead of a parametric surface. Therefore complexity depends on 3D image size rather than on model surface size. This increase in computational expensiveness can make it impossible to gain real time control on model evolution.

The sparse-field approach, proposed by Whitaker [130], solves this problem by tracking the set of voxels, called *active set*, intersected by the level set of interest (usually the 0 level set) at each time step (see Figure 2.8), as well as two layers of voxels around the active set, to compute the required derivatives. This way only voxels in the active set have to be updated, bringing complexity to depend once again on model size<sup>3</sup>. One more advantage of the sparse-field approach is that 0 level set position can be estimated from the values of  $F(\mathbf{x})$  and  $\nabla F(\mathbf{x})$  of the voxels in the active set, so that all differential and image-based quantities can be computed with sub-voxel accuracy. This is done by Newton's method with first-order accuracy

$$\hat{\mathbf{x}} = \mathbf{x} - \frac{F(\mathbf{x})}{|\nabla F(\mathbf{x})|} \mathbf{N}(\mathbf{x}) \quad (2.34)$$

where  $\hat{\mathbf{x}}$  is the estimated zero-crossing position nearest to  $\mathbf{x}$ . This way, even making use of first-order upwind finite differences, the embedded surface does not present aliasing problems.

To gain insight into the sparse-field approach, we first remind that at each time step  $\bar{t}$ ,  $F(\mathbf{x}, \bar{t})$  can be modeled as the signed distance function from  $\mathbf{S}(\bar{t})$ , the 0-level set (see Equation 2.23). Therefore, the active voxels must have a value of  $F(\mathbf{x}, \bar{t}) \in [-\frac{1}{2}h, \frac{1}{2}h]$ ,  $h$  being the voxel spacing, since the 0-level set must intersect the active

---

<sup>3</sup>A fast level set method based on a similar approach was previously proposed by Adalsteinsson et al. [1]

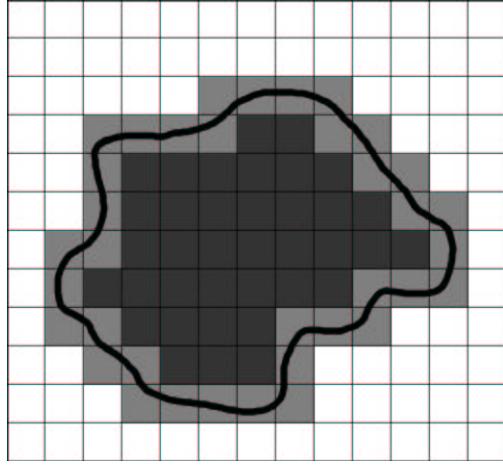


Figure 2.8: In the sparse field approach, level sets evolution is only computed for the set of voxels intersecting the contour of interest.

voxels by definition. Since Equation 2.22 requires the computation of up to second-order derivatives, the values of  $F$  for two layers of voxels on each side of the active set are provided. Voxel values for each inner (or outer) layer are computed from

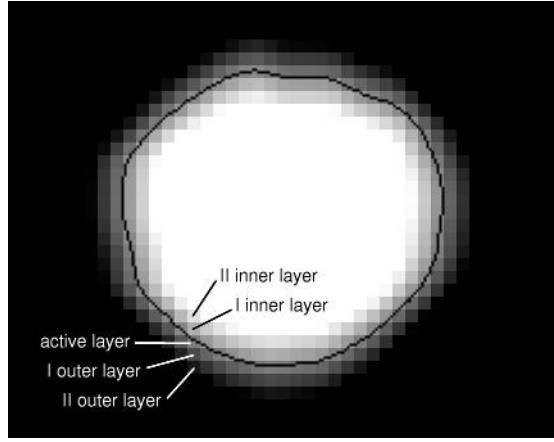


Figure 2.9: Level sets in the sparse field approach, showing the active layer and the four adjacent layers. The 0-level set is contoured.

the value of the voxels of the neighboring layer by subtracting (or adding)  $h$  to the minimum (or maximum) valued neighbor. Therefore, in the sparse field approach, the embedding function is locally reinitialized to the signed distance function at every iteration. As long as one voxel in the active set is updated, its value may become lower than  $-\frac{1}{2}h$  (or greater than  $\frac{1}{2}h$ ). In this case the voxel is moved

from the active set to the adjacent inner (or outer) layer, and the opposite outer (or inner) layer voxel enters the active set. The same procedure is applied to the voxels belonging to the four neighbor layers. The active set and the four neighbor layers are stored in lists which allow fast insertion, removal and neighborhood retrieval. Thanks to this approach, computational complexity does not depend on image grid size, as in dense field level sets solvers, but on active set size, therefore on model size. This way level sets evolution can be made interactive.

The flow chart of the sparse field algorithm is reported in Figure 2.10.

### 2.5.5 Application to blood vessel 3D modeling

Level sets can be successfully applied to blood vessel 3D modeling [4, 35, 124, 123, 127] by performing the initialization inside the vessels of interest on angiographic images. Under the influence of the inflation term of Equation 2.22, single points turn into sphere-like shapes which eventually merge with their neighbors. Surfaces during evolution, as well as final model surface, can be extracted by contouring the 0 level set of  $F(\mathbf{x}, t)$  using Marching Cubes algorithm, as shown in Figure 2.11 superimposed to a surface obtained by contouring with user-defined threshold.

Inflation term is deactivated when the 0 level set approaches the vessel wall, whereas the feature attraction term is activated, in order to make the 0 level set converge to gradient magnitude ridges, as shown in Figure 2.12.

Some methods have been proposed to automatize level sets evolution [13, 108, 115] from initialization to convergence on the vessel wall, but, dealing with generic vessel segments, we preferred to make level sets evolution be driven by an operator, in order to gain more control when acquisition artifacts affect the angiographic images. This way we avoid the application of Gaussian smoothing filters to the images for the calculation of image-based terms, which can have an effect on reconstruction accuracy. The operator can receive feedback on level sets evolution from at least three sources: plots of 0 level set over serial angiographic images or their gradient magnitude images, intermediate polygonal surfaces rendering (eventually superimposed to isolevel surfaces extracted from angiographic images), and the maximum value of level set speed. This latter quantity is employed to automatically detect convergence in the last phase of level sets evolution, the one driven by attraction potential, after a convergence tolerance has been fixed.

If level sets are initialized over an entire vessel tract which includes branches and vessels of different scales, level sets evolution can become difficult to tune, because vessel wall can be approached earlier in smaller vessels than in bigger ones, so that a single set of parameters may not be adequate for all scales. The use of one level sets evolution for an entire branching vessel tract is problematic also because, due to CFL condition,  $\Delta t$  may become small because low gradient is encountered in one image region, slowing down the evolution for the remaining portion of the model. Solutions to both problem include the use of locally adaptive  $w_1, w_2, w_3$  parameters in Equation 2.22 or the use of adaptive grids in which CFL

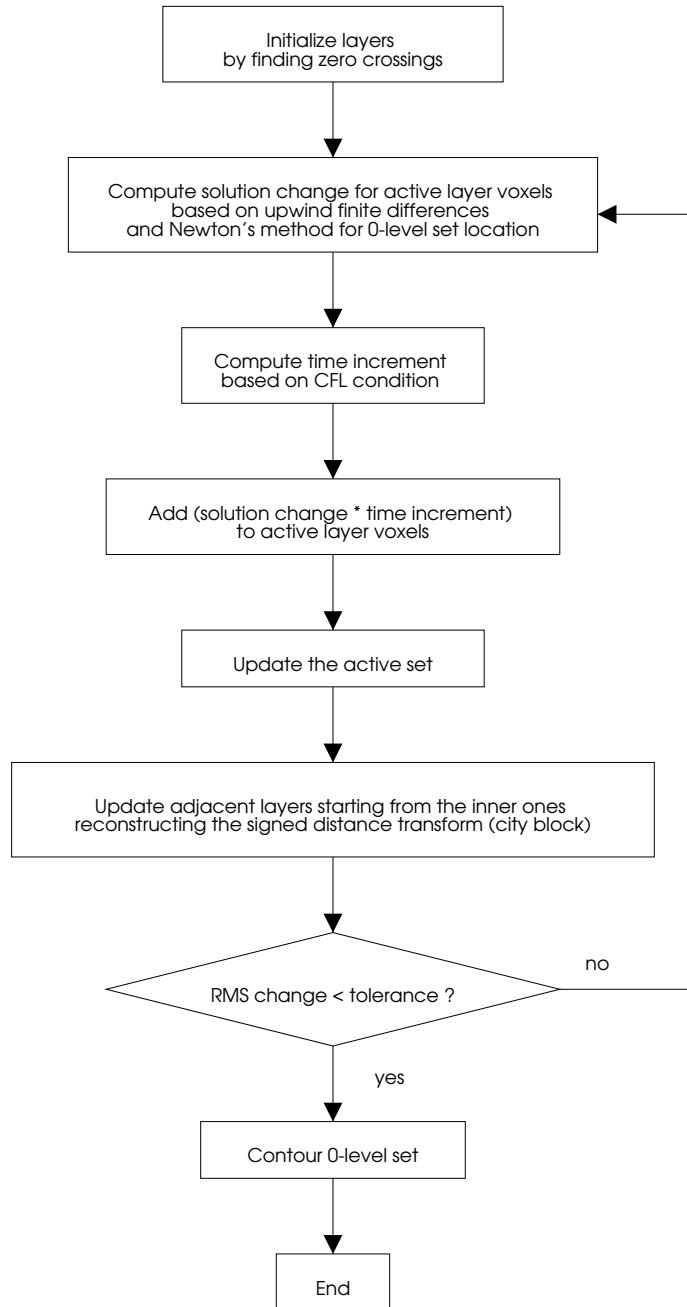


Figure 2.10: Flow chart of the sparse field approach for efficient level sets evolution.

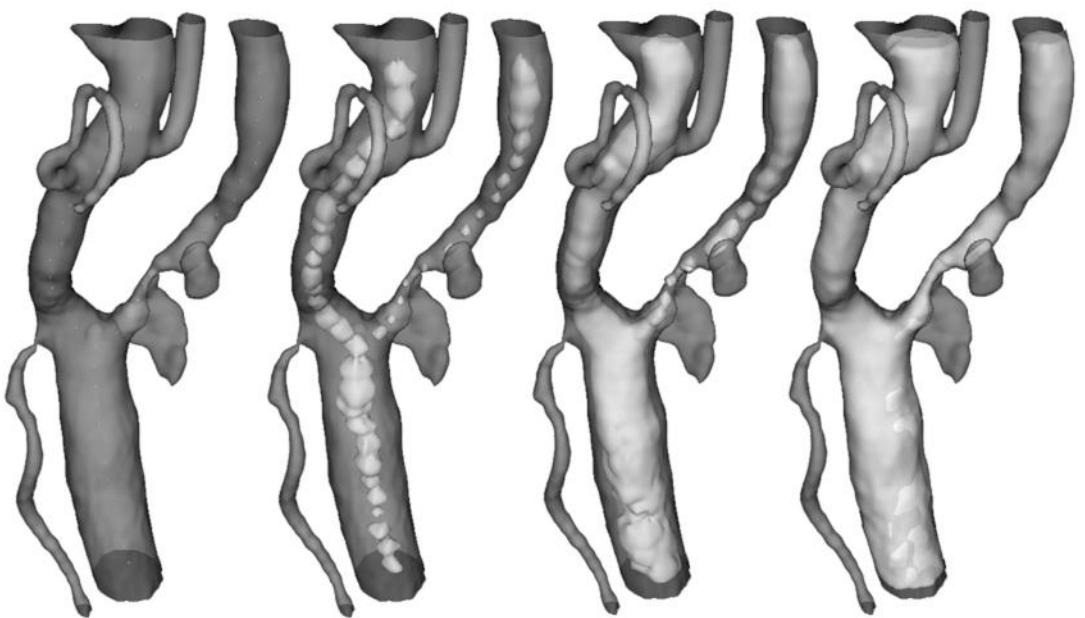


Figure 2.11: Level sets evolution inside a carotid bifurcation acquired by angio-CT. Initialization performed on section images. Notice how collateral vessels and calcified plaques are properly avoided.

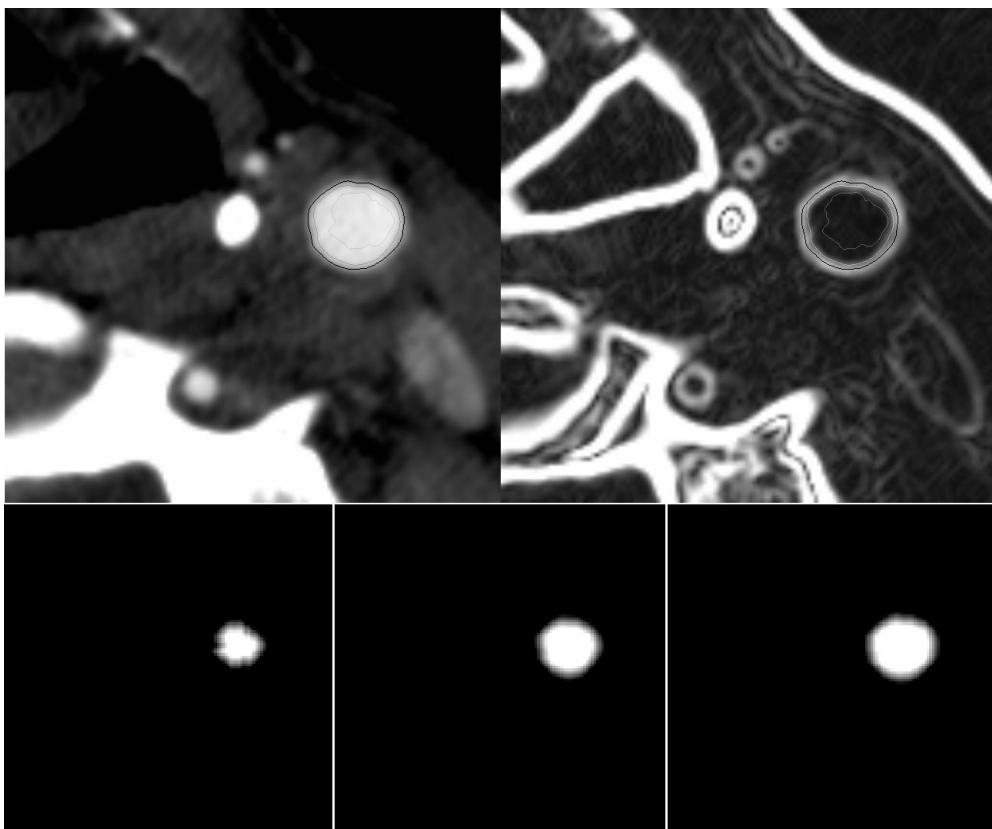


Figure 2.12: Evolution of the 0-level set plotted on the source image (upper left) and on its gradient modulus (upper right). The two inner contours correspond to the inflation phase, while the outer is the result of the activation of the attraction to gradient modulus peaks term. The three lower pictures depict the corresponding three level sets functions.

condition is satisfied acting on  $h$ , by locally refining or coarsening image grid. We instead adopted a simpler approach, made possible by the particular nature of our problem: since evolution parameters are dependent on vessel scale, we let level sets evolve into single vessels, or into groups of vessels of similar scale<sup>4</sup>. Thanks to the implicit nature of level sets, we then merge the  $F_i(\mathbf{x})$  functions resulting from  $N$  single vessel evolutions, and finally extract model surface by contouring the merged  $F_m(\mathbf{x})$  function with Marching Cubes algorithm. Since in the sparse-field approach level sets represent the signed distance function from the 0 level set, with negative values inside the model and positive values on the rest of the domain, merging of  $N$  level sets scalar fields is performed selecting their minimum value

$$F_m(\mathbf{x}) = \min_{i \in [1, N]} F_i(\mathbf{x}) \quad (2.35)$$

An example of single vessel evolution and merging for a carotid bifurcation is depicted in Figure 2.13, while the intersections of model points with gradient magnitude images in the region of the bifurcation apex are shown in Figure 2.14. The evolution of level sets into vessels of similar scale has two beneficial effects. The first is the increased ease of interactively setting evolution parameters (e.g. to switch from inflation to attraction to gradient magnitude ridges). The second is that similar solution changes are computed over the domain, so that more adequate time step values are chosen by the CFL condition, effectively speeding up evolution.

## 2.6 Implementation

All the techniques developed in this work have been implemented in C++ using Visualization Toolkit (VTK 4.1, see [105]) library.

Images were transferred from CT and MR scanners and read using a simple DICOM reader implemented by us as a subclass of `vtkImageDataReader`, on the basis of the DICOM standard protocol.

The reconstruction technique was implemented in C++ using VTK 4.1, on the basis of R. Whitaker implementation given in VISPACK library. VTK classes were mainly used to provide basic data structures (`vtkStructuredGrid` and `vtkPolyData`), input/output operations, image gradient computation (`vtkImageGradient`) and the contouring algorithm (`vtkMarchingCubes`).

The sparse field level sets algorithm was implemented as a VTK filter, derived from `vtkStructuredPointsToStructuredPointsFilter` class. A base sparse field level sets solver (`vtkLevelSetsMachinery`) class was constructed as a derived class of `vtkStructuredPointsToStructuredPointsFilter` class, which provided the sparse field level sets mechanism of Figure 2.10. The methods for the computation of image-based terms in level sets equation were kept virtual, and were implemented in a derived class (`vtkLevelSets`), inherited from `vtkLevelSetsMachinery`. Class `vtkLevelSets` takes as the input an image, which is used for level sets initialization,

---

<sup>4</sup>In the next chapter a method for fast single vessel initialization will be presented.

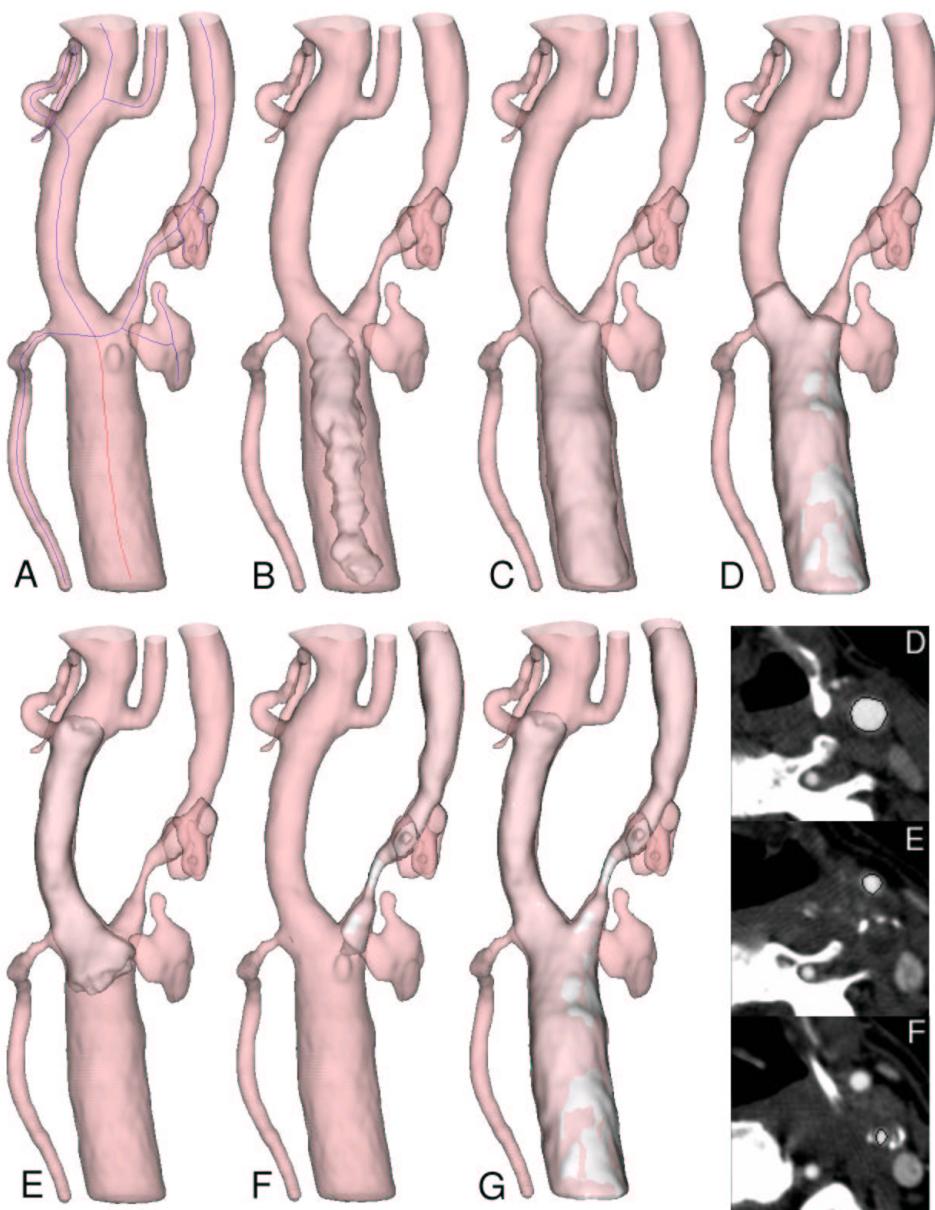


Figure 2.13: Level sets evolution inside a carotid bifurcation acquired by contrast-enhanced CT. Initialization in single vessels based on centerline and bifurcation identification (A). B, C, D: common carotid artery reconstruction (B, C: evolution under inflation; D: evolution under attraction to gradient peak). E: external carotid artery reconstruction. F: highly stenotic internal carotid artery reconstruction. G: final merging of the three reconstructed branches. Lower right: plots of 0-level set over the source images for the corresponding branches.

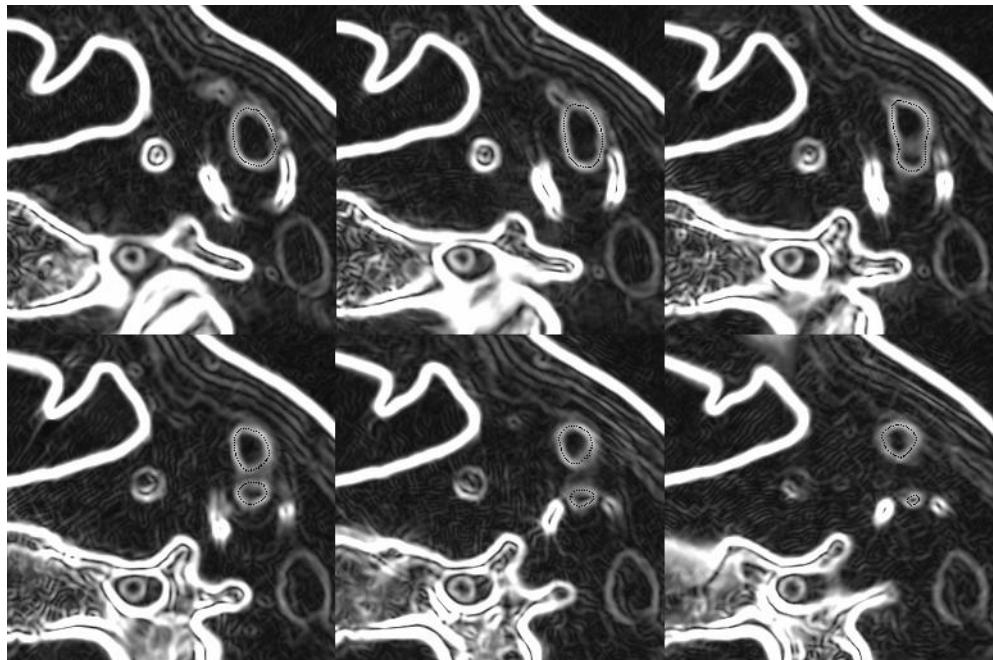


Figure 2.14: Gradient magnitude images of the carotid bifurcation shown in Figure (2.13) in the region of the bifurcation apex. The points of the model resulting from level sets evolution have been plotted on the images, demonstrating that 0-level set correctly converged on gradient magnitude ridges.

and returns as the output the same image at the end of evolution. Level sets initialization can also be performed from the points of a `vtkPolyData` object, as in the case of initialization from centerlines. Two more images are used as input for the computation of inflation (a scalar field, such as source image gradient magnitude) and force (a vector field, such as source image gradient) terms. The evolution parameters and the number of iterations to perform with that parameters are also set. At the end of the specified number of iterations it is possible to change parameter values.

Single vessel evolution is performed instantiating multiple `vtkLevelSets` classes, and the merging step is handled outside the `vtkLevelSets` classes.

## 2.7 Validation

The reconstruction algorithm has been validated on synthetic images of cylinders with different resolutions, orientations and noise levels. Some of the results are reported in the following. Synthetic cylinders were constructed in such a way to mimic contrast medium in medical images, which decreases smoothly and rapidly near the vessel wall. To this extent, a translated sigmoid function with cylindrical symmetry was constructed, such that its zero level-set was located over sigmoid inflection curve, which corresponds to gradient modulus maxima. The resulting function is

$$F(r) = C \left( \frac{1}{2} - \frac{1}{1 + e^{-a(r-R)}} \right) \quad (2.36)$$

where  $R$  is the radius of the cylinder, that is the distance between the symmetry axis and the zero level set of  $F(r)$ ,  $a$  controls the steepness of the sigmoid at the inflection point (the gradient modulus at the inflection point is  $a$ ), and  $C$  is a scale factor. Figure 2.15 shows a plot of function  $F(r)$ .

Table 2.1 shows the results of testing reconstruction performance against cylinder axis orientation with respect to grid axes and sampling density. The parameters for function  $F(r)$  were set as in Figure 2.15, i.e.  $C = 1$ ,  $a = 10$ ,  $R = 1$ , from the observation of a set of real cases. Reconstruction was performed activating only inflation and image driving terms, without employing the regularizing curvature term of Equation 2.22. Results in Table 2.1 show that reconstruction is accurate for cylinders whose radius of down to 4 voxel wide, while accuracy is lower in the 2 voxel radius case. As to orientation, better results are obviously obtained when the cylinder axis is aligned with the grid. However, good results are also obtained for rotation of cylinder axis of 30 and 45 degrees.

In Table 2.2 level sets performance is tested against image noise. Noise was generated by filtering with a Gaussian smoothing filter, of standard deviation equal to 2 pixels, an uniform probability image. Again, reconstruction was performed without using the curvature term of Equation 2.22.

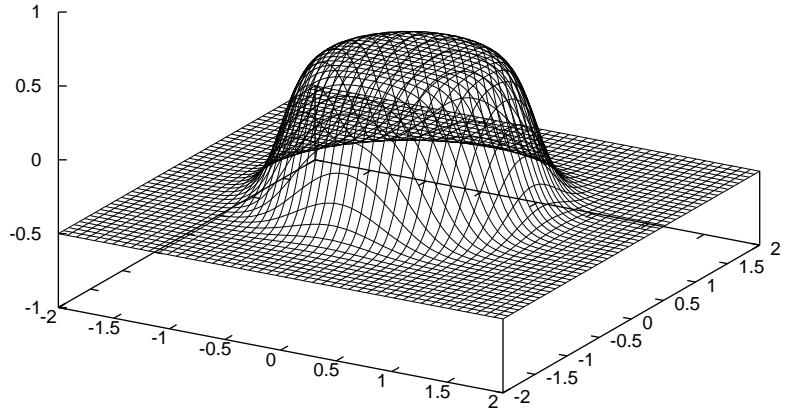


Figure 2.15: Plot of  $F(r)$  in Equation 2.36 with  $C = 1$ ,  $a = 10$ ,  $R = 1$ .

$h/R$	$\theta$	$\mu(\sigma) \%$	Max %
0.125	0	0.6 (0.3)	1.1
	30	0.4 (0.3)	1.5
	45	0.5 (0.3)	1.1
0.25	0	1.2 (0.6)	2.0
	30	1.7 (1.2)	4.9
	45	1.0 (0.8)	2.8
0.5	0	6.7 (5.1)	11.9
	30	6.5 (3.7)	16.1
	45	8.7 (2.8)	13.2

Table 2.1: Level sets validation results on sigmoidal cylinders sampled at different resolutions ( $h/R$  is voxel side size relative to cylinder radius) and with different orientations with respect to grid axes ( $\theta$  is the angle of rotation of cylinder axis around one grid axis starting from alignment at  $\theta = 0$ ). Mean, standard deviation and maximum error are expressed in percentage as the difference between distances of model points to cylinder axis and cylinder radius with respect to cylinder radius.

$h/R$	SNR	$\mu(\sigma) \%$	Max %
0.125	1	1.4 (1.1)	6.7
	0.5	2.6 (2.1)	14.0
	0.3	4.0 (3.8)	32.5
0.25	1	1.4 (0.9)	7.9
	0.5	1.9 (1.5)	8.7
	0.3	2.5 (2.1)	14.5

Table 2.2: Level sets validation results on sigmoidal cylinders sampled at different resolutions ( $h/R$  is voxel side size relative to cylinder radius) and with different noise levels (uniform noise was generated and then filtered with a Gaussian smoothing filter with  $\sigma = 2$  pixels, in order to select a scale similar to that of real angiographic images. The obtained image was then added to the cylinder function image. SNR represents the ratio between cylinder function range ( $\pm 0.5$ ) and noise range.). Mean, standard deviation and maximum error are expressed in percentage as the difference between distances of model points to cylinder axis and cylinder radius with respect to cylinder radius.



# Chapter 3

## Topological analysis of vascular structures

### 3.1 Introduction

Dealing with vascular models directly in three dimensions places the problem of drawing synthetic information about their 3D structure, for description and user interaction. In particular, for complex interconnecting vessels it is informative to describe the mutual organization of bifurcations and single segments and to give a sketch of their spatial configuration. We will refer to this task as to topological analysis, even if some geometric information will be derived, in order to distinguish this problem from accurate geometric analysis performed in Chapter 5.

We here present a fast algorithm for topological analysis of interconnecting vessels which has been developed for the analysis of large microvascular networks [7]. In the context of large arteries modeling, this technique will be employed for fast interactive level sets initialization, which is therefore performed directly in 3D, and for automatic model decomposition in the generation of computational meshes made of hexahedral elements.

### 3.2 Algorithm description

The topological analysis algorithm we present in this section was developed with the main aim of identifying vessel bifurcations and vascular network organization, and to give an approximate sketch of segment centerlines starting from model surface in the form of a piecewise linear surface, such as that obtained from Marching Cubes algorithm.

The idea of the algorithm is that a single vessel, from a topological point of view<sup>1</sup>, is a cylinder limited by two topologically circular sections, also known as  $2\frac{1}{2}$ D geometry. This terminology comes from the fact that a  $2\frac{1}{2}$ D geometry can be

---

<sup>1</sup>i.e. regardless the particular shape taken by the vessel surface

obtained by sweeping a 2D variable shape along a 1D path. A topological cylinder can be subdivided into smaller cylindrical pieces by cutting its surface in such a way that a limiting circular section is produced. We refer to limiting circular section as to a section which is not the boundary of a topological disk. Such a cut can be performed with a sphere centered in a limiting section barycenter and with radius slightly greater than the maximum distance between the barycenter and the farthest point on section profile. Thus, if we proceed along a topological cylinder starting from one of the two limiting sections and cutting small subcylinders with a sphere centered in each new limiting section created and completely containing it, at each step one new limiting section will be created. In contrast, if presence of a bifurcation, two or more new limiting sections will be created. Based on this obser-

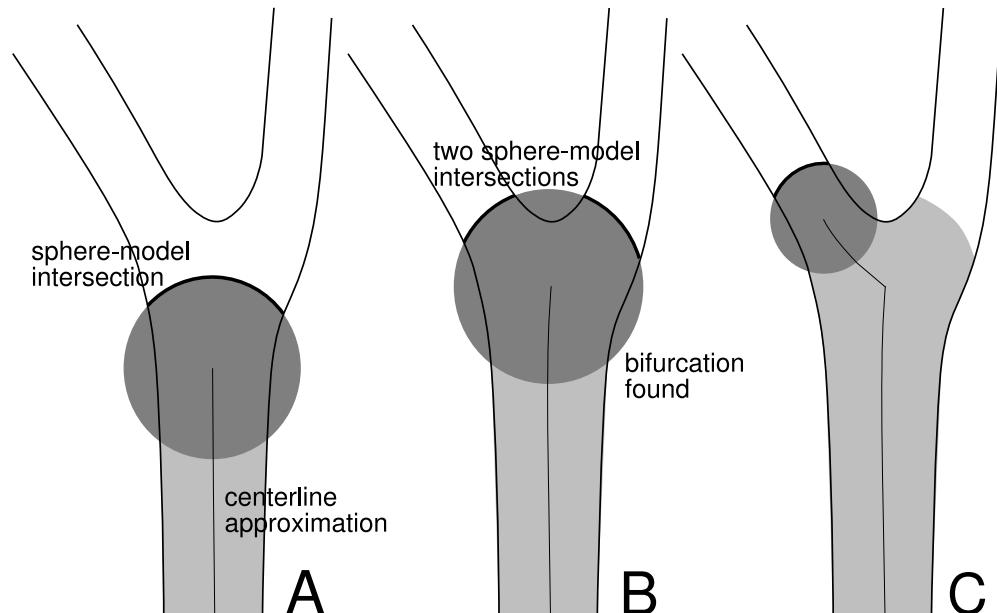


Figure 3.1: Conceptual representation of the topological analysis algorithm. A: one intersection between the probing sphere and the model is generated, and a topological cylinder is identified. B: two intersections are generated, and a bifurcation is identified. C: the probing sphere restarts from one of the intersection profiles. Centerlines are generated by connecting the centers of the probing sphere during its movement.

vations, the algorithm consists in visiting the whole network of vascular segments with a moving sphere, which we will name *probing sphere*, such that all bifurcations are identified, thus decomposing the network into bifurcations and single vessel segments (see Figure 3.1). Moreover, the barycenter of limiting sections provide an approximation of vessel centerlines, used to draw information about vessel spatial configuration. Instead of effectively cut the vessel segments with the

moving sphere, which would require a retriangulation of the triangles intersecting the sphere, vessel surface points are visited with the probing sphere and limiting section profiles are simply drawn on the surface, thus limiting the computational effort required.

In more detail, referring to Figure 3.2, the algorithm starts with the identification of the open profiles of the vessel structure, i.e. the inlet and outlet section profiles. Therefore, if the model is a closed surface, at least one open profile must be created. Once one profile is chosen, its barycenter is calculated. Since profiles are piecewise linear lines, their barycenter is computed taking into account the segments connecting polyline vertices by weighting point positions with the half-length of their neighboring segments. The probing sphere is then defined with the center  $\mathbf{c}_s$  on the barycenter  $\mathbf{b}$  of the defining profile  $\mathbf{P}$ , and with a radius equal to

$$r_s = k d_{max} \quad (3.1)$$

where  $d_{max} = \max \{|\mathbf{b} - \mathbf{P}|\}$  is the maximum distance of a profile point to the barycenter, and  $k$  is a user-defined constant, set slightly greater than 1, e.g.  $k = 1.05$ .

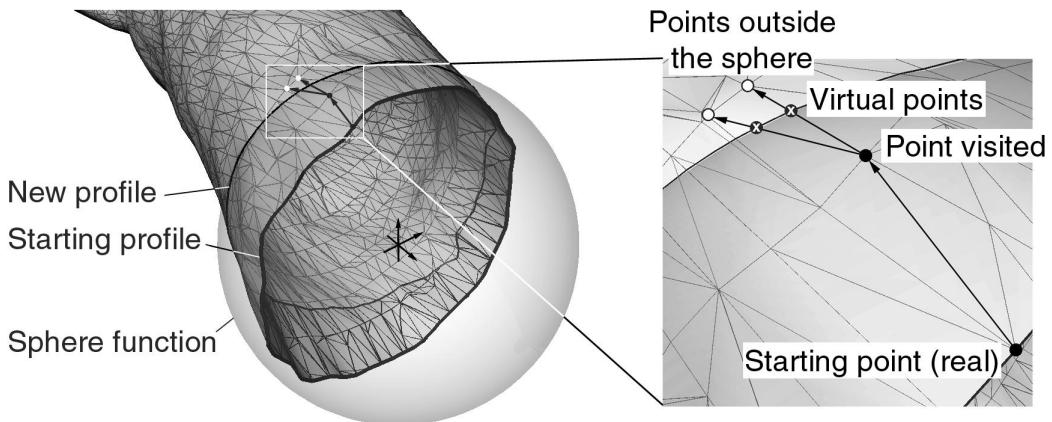


Figure 3.2: Particular of one step of the topological analysis algorithm. Intersection points are referred to as virtual points to evidence that they are not vertices of surface triangles, but they are created on triangle edges.

At this point, the intersection of the sphere with the surface connected to the defining profile is calculated. This is done by iteratively visiting neighboring surface mesh vertices starting from the defining profile and checking if

$$|\mathbf{p} - \mathbf{c}_s| > r_s \quad (3.2)$$

where  $\mathbf{p}$  is the position of a generic surface mesh vertex. Visited mesh vertices which do not satisfy the above condition are marked as *visited*. As soon as a point

satisfies the above condition, it means that the intersection between the probing sphere and the surface is located on the edges connected to that point, and whose other point is a visited point lying inside the sphere (i.e. it doesn't satisfy the condition in Equation 3.2). The exact position of intersection is determined with the following relations

$$\begin{aligned} s &= (r_s^2 - h^2)^{\frac{1}{2}} - (t^2 - h^2)^{\frac{1}{2}} && \text{if } |\mathbf{p} - \mathbf{q}|^2 < (u^2 - h^2) \\ s &= (r_s^2 - h^2)^{\frac{1}{2}} + (t^2 - h^2)^{\frac{1}{2}} && \text{if } |\mathbf{p} - \mathbf{q}|^2 \geq (u^2 - h^2) \end{aligned} \quad (3.3)$$

where quantities are specified as in Figure 3.3, and the drawing plane is that defined by edge  $(\mathbf{p} - \mathbf{q})$  and sphere center  $\mathbf{c}_s$ . Each intersection point constitutes a vertex of the polyline(s) representing newly generated profile(s). In order to reconstruct polyline connectivity, each two intersection points are connected by an edge if they lie on two edges of the same triangle.

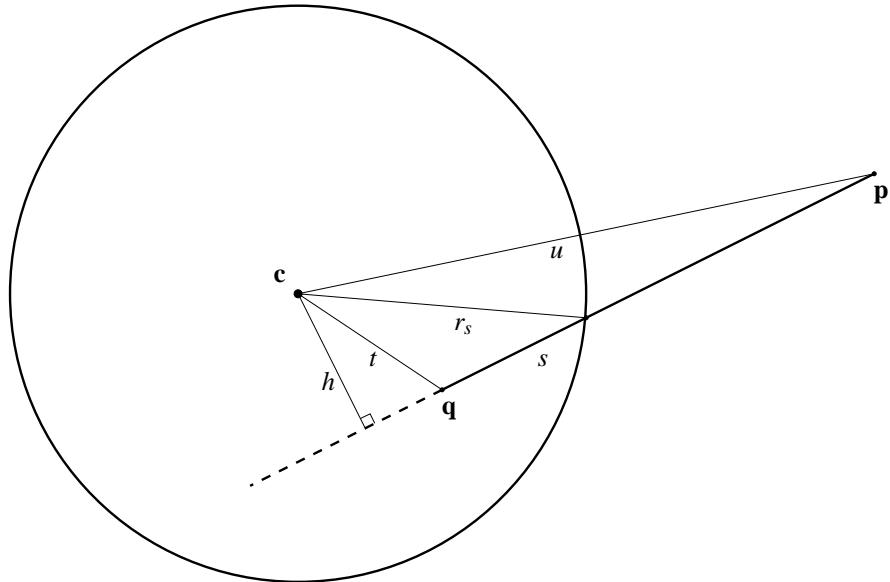


Figure 3.3: Determination of the intersection between the probing sphere and an edge.

Once intersection profiles are reconstructed, they can be counted. If one new profile is generated, the surface between the defining and the new profile is a topological cylinder, therefore the new profile is taken as defining profile, and a new step is performed. If no profiles are generated, a dead end of a vessel segment has been found, so the set of probing sphere centers are stored as a vessel segment and the analysis continues with a new segment from an available (i.e. not visited) profile. If two or more profiles are generated, the surface contained in the sphere is a bifurcation, therefore the vessel segment is stored, and the new profiles are

made available. It can also happen that during point visiting procedure an available profile is encountered. This means that the current segment ends on a bifurcation which has been already identified, or on an outlet of the model.

Each time a bifurcation is encountered, it is assigned a unique id, and its profiles are marked with it. This way, whenever a segment has been constructed, its endpoints are associated with the id of the bifurcations enclosing it. Therefore vascular network topology can be reconstructed without the need of any particular order in visiting segments.

During the analysis of segments, an estimate of mean radius is calculated at each step of the probing sphere by computing the mean distance of the points visited at that step from the line defined by the barycenter of the defining and the intersection profiles, which approximates vessel centerline. For bifurcations, the barycenter of the bifurcation surface is calculated, and it is joined to the endpoints of the afferent branches.

A flow-chart of the algorithm is shown in Figure 3.4. The algorithm is very efficient, because its complexity is linear with the number of surface mesh vertices, since surface points are visited once. Anyway, the algorithm presents some limitations. From the topological point of view, topology can be misinterpreted in two situations. First, when a segment branches into two segments which in turn rejoin into one right after, the two bifurcations may be ignored if the probing sphere radius is greater than the distance between the bifurcations. Second, for the same reason, nested bifurcations occurring at very short distance from each other may be grouped into a single bifurcation comprehending all the vessels departing from the bifurcation group. Both limitations derive from the fact that probing sphere size can grow when the defining profile has a flat elliptical shape, which can happen when approaching bifurcations. However, these cases are not limiting, because all salient branches are counted with appropriate global connectivity. On the other hand, limitations on geometric analysis performance are more important. In fact, there is no guarantee that segment centerlines lie inside the associated vessels, mainly due to the definition of centerlines adopted here as the locus of defining profile barycenters. Centerlines can lie outside vessels when the sections have particularly non-convex shape. However, the most limiting geometric problem is that nothing is inferred about bifurcation geometry once they have been included into the probing sphere. Typically, single vessel centerlines are connected to the barycenter of the bifurcation surface, but this can result in oversimplified geometric approximation. In order to reduce these effects,  $k$  in Equation 3.1 must be chosen near to 1. On the other hand, choosing a value for  $k$  excessively near to 1 can produce the effect of identifying false bifurcations when the surface presents bulging. Although it is possible to detect and remove the majority of false bifurcations by thresholding short vessels which have a dead-end, information on vessel radius is lost whenever a false bifurcation is identified, so that  $k$  must not be chosen too close to 1. We empirically found a good compromise to be in the range  $1.01 \leq k \leq 1.1$ , depending on surface regularity.

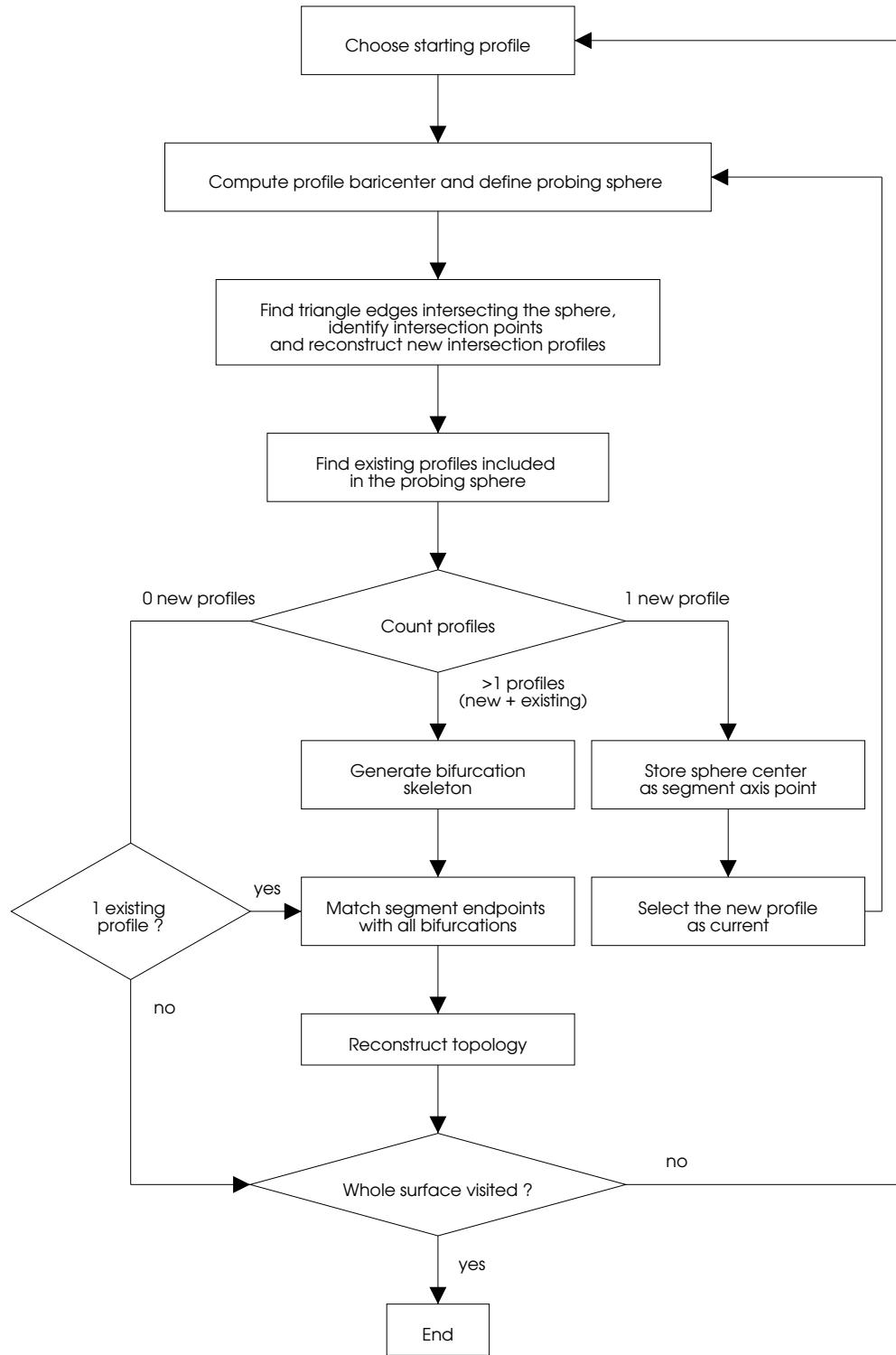


Figure 3.4: Flow chart of the **topological analysis algorithm**.

### 3.3 Implementation

The algorithm summarized in Figure 3.4 has been implemented in C++ using VTK class `vtkPolyData` as the basic data structure.

### 3.4 Validation

The algorithm has been validated on a set of structures of known topology and geometry, as the one in Figure 3.5. The results of validation on such structure are

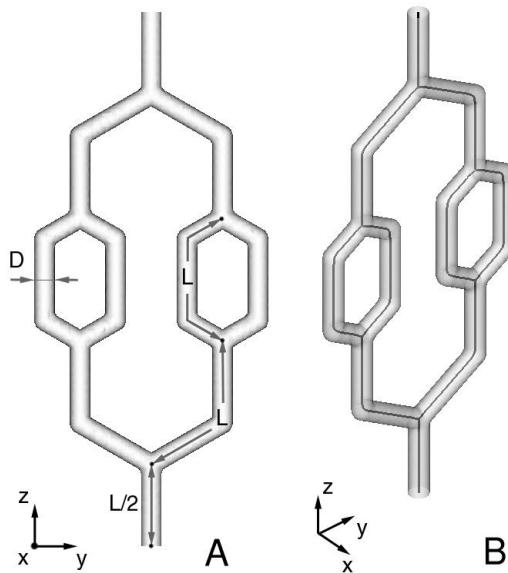


Figure 3.5: A: synthetic structure employed for the validation of the topological analysis algorithm. B: centerline resulting from the application of the algorithm.

reported in Table 3.1.

$L/D$	Length $\mu(\sigma)$ %	Radius $\mu(\sigma)$ %
12	1.3 (0.8)	0.2 (0.1)
8	1.5 (1.0)	0.1 (0.1)
4	3.0 (1.0)	0.3 (0.2)

Table 3.1: Validation results for the structure in Figure 3.5, expressed as percentage errors relative to the true values.

## 3.5 Applications

As anticipated in the introduction, the algorithm has not been developed in the context large artery modeling. Anyway, thanks to its computational inexpensiveness and the ability of automatically and robustly detecting salient bifurcations and constructing approximations of vessel segment centerlines, it can be conveniently used in this field. In the following we will briefly describe the original application of the algorithm, and then describe two applications to large artery modeling.

### 3.5.1 Microvascular network analysis

Microvascular networks, such as the renal glomerulus, the subunit of the nephron responsible for ultrafiltration, have a very complex 3D structure of interconnecting capillary segments. Investigation of such structure has been typically performed on 2D images of histological sections acquired by means of optical or confocal microscopy. However, investigation of 3D structure of microvascular networks may provide an important insight on physiology and pathology of microcirculation, given the tight relation between structure and function in such context. In fact, the observation of a local vascular feature on a 2D section gives no information about the role of such feature on the function of the whole network, basically because the position of such feature inside network topology is unknown. This is the context in which the presented algorithm was developed. The main requirements were therefore to accurately detect network topology and to easily deal with large datasets.

Briefly, microvascular network endothelial surface can be reconstructed by first segmenting a volume of histological section 2D images, which must be performed under operator supervision given the variability of histological images and the importance of detecting the whole luminal space, and then by contouring the segmented volume with an algorithm such as Marching Cubes. The reconstructed surface can then be processed with the described algorithm, after opening the surface at one or more network inlets to create a starting profile. As a result, a 3D network of interconnecting centerlines is obtained (see Figure 3.6 A), with associated mean radius values. A convenient representation of microvascular networks is obtained by drawing a 2D layout of network topology. In practice, each bifurcation is represented as a node, and every capillary segment as a straight line between two nodes, disregarding real bifurcation position and capillary segment spatial configuration and length. A convenient organization of nodes on a 2D plane is then searched, in order to minimize line intersections and to distribute nodes according to their role in network topology (e.g. their position on the vertical axis can reflect the value of pressure obtained from a simple lumped parameter fluid-dynamic simulation on the network), as shown in Figure 3.6 B. This helps to enlighten the functional role of segments and bifurcations. The algorithm and its application to the glomerular capillary were presented in [7].

Thanks to its efficiency, the algorithm described here makes it possible to ex-

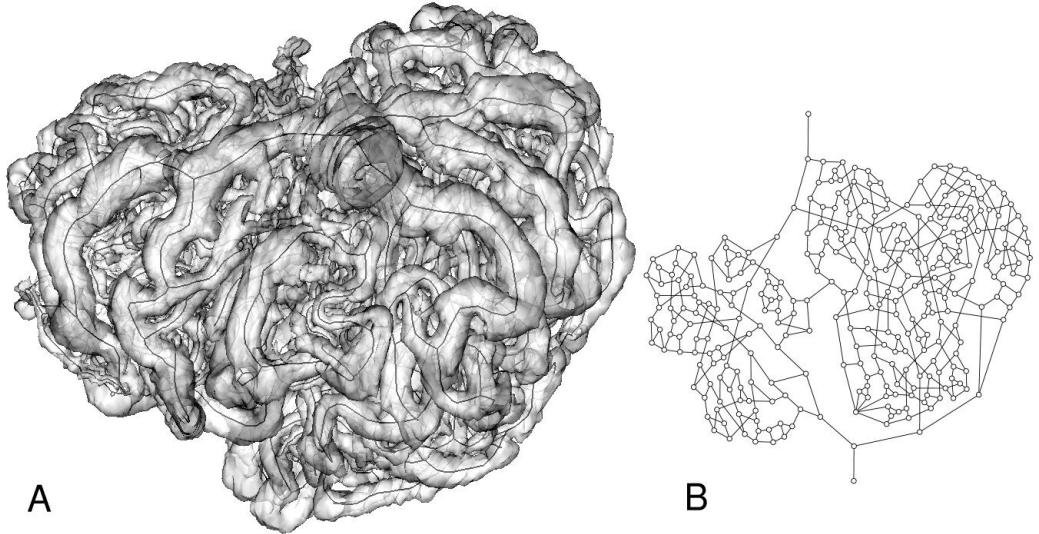


Figure 3.6: A: renal glomerulus surface and network of centerlines. B: 2D layout of glomerular network topology.

tend such 3D analysis to populations of modelled microvascular networks with a large spectrum of applications, ranging from quantification of glomerular capillary structural damage under pathologic conditions, to the study of microcirculation in tumoral tissues, to the characterization of angiogenesis in engineered tissues.

### 3.5.2 Single vessel level sets initialization

While for microvascular networks the algorithm plays a central role, for the analysis of arterial tracts the main need is to accurately analyze geometry (see Chapter 5), since topology is usually much simpler. However, in some cases, automatic detection of vessel topology and quick retrieval of vessel centerline approximations can result to be very useful. One case is that of deformable model initialization, usually performed selecting single points or regions on 2D images.

We used the following approach for initialization of level sets, presented in Chapter 2. First the volume of interest is contoured with Marching Cubes at a user-specified threshold near the maximum transition between contrast medium and the surrounding tissue. The contouring surface is then near to the vascular wall and reproduces the topology of the vascular tract. The application of the topological analysis algorithm presented in this chapter to the contouring surface yields an approximation of interconnecting centerlines. Level sets initialization is then performed by interactively selecting single centerlines, and making the deformable model start to evolve from centerline points. The great advantage of this approach is that single vessel initialization does not require to identify points inside specific vessel segments on the angiographic image stack, which requires interpretation of

the 3D structure from 2D vessel profiles. This way it is possible to separately initialize and control the evolution of level sets inside vessels of the same size, thus avoiding scale-dependence effects on evolution parameters, as described in the last section of Chapter 2.

### 3.5.3 Model decomposition for computational mesh generation

One more application of the topological analysis algorithm concerns the generation of a computational mesh based on the geometry of the reconstructed models. We here give an anticipation of what will be presented in detail in Chapter 6. While it is simple to generate meshes made by hexahedral elements for a single topological cylinder by mapping it onto a discretized reference cylinder, generating hexahedral meshes for generic interconnecting  $2\frac{1}{2}$ D geometries in an automatic way is not a trivial task. One approach is that of decomposing the model in single  $2\frac{1}{2}$ D sub-units, to create mapped meshes for each subunit, and finally to reassemble the sub-units into the original volume. Since this technique requires to adequately split the model in correspondence of bifurcations, the topological analysis algorithm turns out to be useful in automatically detecting bifurcation location and branches orientation. Based on this information, the model can be automatically split, meshed and re-merged. This approach was adopted for mesh generation of arterial tracts containing one bifurcation, such as the carotid or the iliac bifurcation. The reason of this choice is that better results are obtained when the common vessel branches into two vessels of similar caliber. The splitting modalities will be shown in detail in Chapter 6.

## Chapter 4

# Three-dimensional editing of piecewise linear surfaces

### 4.1 Introduction

Models obtained with reconstruction techniques such as level sets or balloons are not directly suitable for geometric analysis or CFD simulation. In fact, imaging artifacts can lead to regions of local irregularities for which the use of global smoothing filters is not adequate. Furthermore, bulging regions can be present near the branching of unreconstructed secondary vessels. In fact, even with the use of single vessel initialization, deformable models tend to seep into secondary vessels in which level sets have not been initialized (see Figure 4.1). The activation of the curvature term in level sets evolution equation (Equation 2.22) limits this effect, and it can eventually prevent it, but the price to pay is global loss of model detail. One last issue is that level sets cannot properly model inlet and outlet vessel boundaries. In fact, the models obtained by level sets evolution, besides being closed surfaces by construction, present a rounded shape at their inlets and outlets, due to smoothness in angiographic image gradient modulus and to the eventual activation of curvature terms. Therefore, boundary between inlets/outlets and vessel wall is not clearly discernible, and analytic expression of boundary conditions for CFD simulation becomes troublesome.

The solution to these problems requires either the imposition of modeling constraints in the reconstruction phase, or the use of techniques which allow to edit the reconstructed models. Modeling constraints include angiographic image editing, which is usually done manually on 2D serial images. By manually identifying single pixels or image regions it is possible to avoid deformable model seeping into unwanted branches, but with operator-dependent and slice orientation-dependent results. A more advanced approach of modeling constraint imposition consists in single vessel reconstruction with deformable generalized cylinder models [135], which can only deform in the radial direction, so that seeping is avoided and inflow/outflow boundaries are properly generated. Although this technique has been

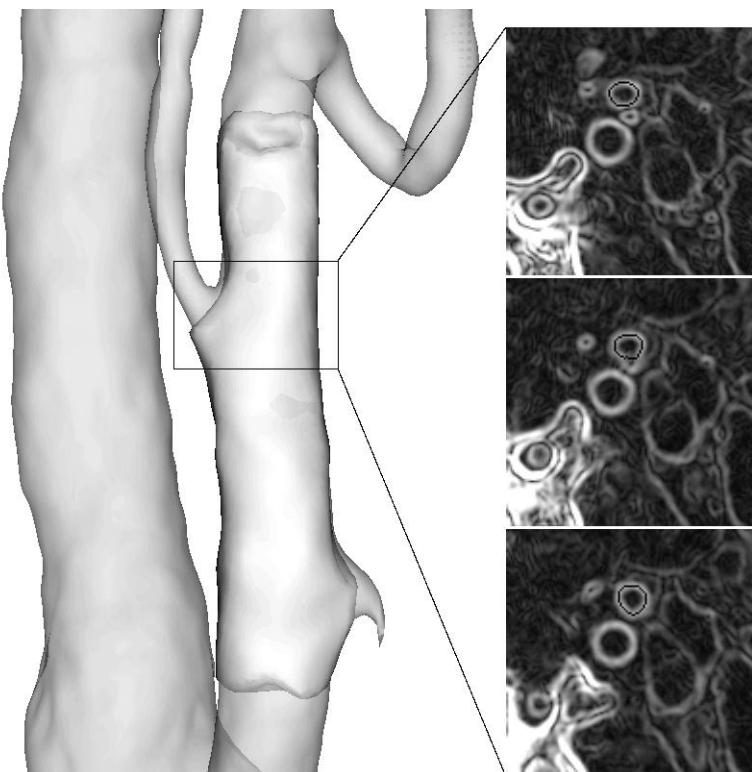


Figure 4.1: Example of surface bulging caused by the presence of unreconstructed branches.

proved to be effective, it can present limitations when complex non-tubular vascular anatomic structures, such as aneurysms, have to be reconstructed. For this reason we instead chose to develop instruments for editing models obtained after level sets evolution and directly in three-dimensions, so to avoid slice orientation-dependence effects.

Undesired surface feature removal, which can be referred to as to *local surface fairing*, can be formulated as finding a new configuration for a subset  $\mathbf{B}$  of a surface  $\mathbf{S}$  such that the resulting surface  $\mathbf{S}'$  minimizes mesh frequency content in a neighborhood of  $\mathbf{B}'$ . In other words, the feature removal operation must remove undesired surface information giving rise to the minimum amount of new surface information. This criterion helps to modify vessel wall surface in such a way to obtain regular flow field solutions from CFD computations. In this chapter we will present two techniques of increasing complexity to accomplish this task, along with their application to local smoothing and bulging removal and inlet/outlet boundaries enhancement.

## 4.2 Weighted Laplacian filtering

Before treating the problem of local surface fairing, we will introduce the closely related problem of surface smoothing [116], which will be useful in introducing mesh processing instruments. Given a surface  $\mathbf{S}$ , surface regularization can be obtained by moving toward the minimization of the following membrane energy functional [34]

$$\mathcal{E}_{membr}(\mathbf{S}) = \frac{1}{2} \int_{\Omega} (\mathbf{S}_r^2 + \mathbf{S}_s^2) dr ds \quad (4.1)$$

where  $(r,s)$  is surface parameterization, which for the moment is supposed orthonormal (see Equation 2.8). The first variation of  $\mathcal{E}_{membr}(\mathbf{S})$  corresponds to the Laplacian operator

$$\Delta \mathbf{S} = \mathbf{S}_{rr} + \mathbf{S}_{ss} \quad (4.2)$$

which has the meaning of a local *averaging* operation among surface points (Laplacian is in fact the operator associated with diffusion phenomena). Following the same approach used for minimizing snake energy functional in Equation 2.2, it is possible to achieve steepest gradient minimization of the membrane energy functional in Equation 4.1 through the solution of the following evolutionary partial differential equation

$$\frac{\partial \mathbf{S}}{\partial t} = \lambda \Delta \mathbf{S} \quad (4.3)$$

which describes the evolution in time of surface  $\mathbf{S}(r,s,t)$  according to the membrane analogy.

To apply the above evolution equation to surface meshes it is necessary to discretize it both in time and in space. Time integration for Equation 4.3 can be performed by means of a forward scheme such as

$$\mathbf{S}^{k+1} = \mathbf{S}^k + \lambda \Delta \mathbf{S}^k \Delta t \quad (4.4)$$

where  $\Delta t$  must satisfy the CFL condition to achieve stability. Spatial discretization of the Laplacian operator over piecewise linear surfaces can be given with the following general expression (see Figure 4.2)

$$\Delta x_i = w_{ij} \sum_{j \in N_1(i)} (x_j - x_i) \quad (4.5)$$

where  $x_i$  is the  $i$ th mesh vertex,  $N_1(i)$  its 1-neighborhood, i.e. the set of vertices connected to  $x_i$  by a triangle edge, and  $w_{ij}$  are weights whose values depend on local geometry and mesh topology. The choice of  $w_{ij}$  influences the accuracy of the

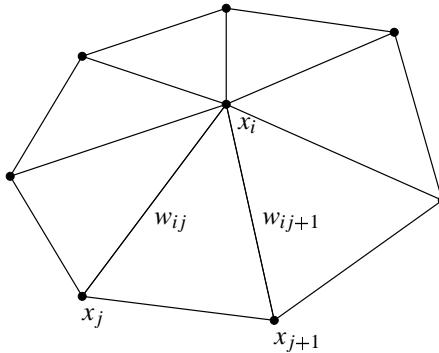


Figure 4.2: 1-neighborhood for a piecewise linear surface.

discretization. The scheme obtained by substituting this discretized Laplacian into time integration scheme in Equation 4.4 is also known as *mass-spring* smoothing scheme, as mesh edges can be assumed to be springs attracting mesh vertices. In practice, as previously anticipated, this scheme moves a point along a weighted average of its neighbors, thus performing low-pass filtering of the surface.

Among possible choices of weights in Equation 4.5 for Laplace operator approximations, the *umbrella operator* is the simplest

$$\Delta x_i = \frac{1}{m} \sum_{j \in N_1(i)} (x_j - x_i) \quad (4.6)$$

where  $m$  is the number of vertices of  $N_1(i)$ . For the purpose of shape-preserving mesh smoothing, the discretization in Equation 4.6 suffers from several limitations, deriving from the fact that  $x_j$  vertices are equally weighted regardless the width of the angles  $x_{j-1}\hat{x}_ix_j$  and  $x_j\hat{x}_ix_{j+1}$ . In fact, due to this, two very different neighborhood configurations, such as the ones in Figure 4.3, which represent two different mesh frequencies, receive the same smoothing amount [34]. Therefore, since regularization equally involves low and high mesh frequencies, surface shape is not preserved. Moreover, such surface mesh regularization also affects surface parameterization (i.e. surface triangle edges and inner angles), which is not a desired effect when the original mesh present regions of local refinement. However, while

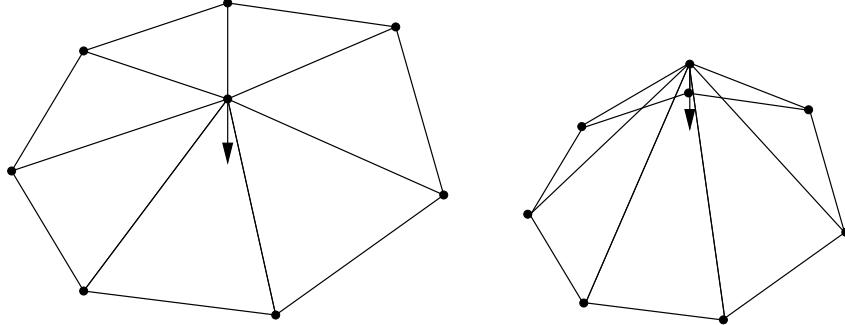


Figure 4.3: Two different neighborhoods which receive the same smoothing amount with the umbrella operator.

for high-frequency shape-preserving noise smoothing the umbrella operator is not adequate<sup>1</sup>, the low-frequency regularization feature turns out to be useful for our local surface fairing problem.

Let's consider the smoothing scheme of Equation 4.4 in the context of local surface fairing. If this scheme is solved until convergence for a subset  $\mathbf{B} \subset \mathbf{S}$ , with point positions on  $\partial\mathbf{B}$  as boundary conditions, the resulting  $\mathbf{B}'$  will be configured as a membrane, that is, frequency content inside  $\mathbf{B}'/\partial\mathbf{B}$  is minimized, but this is not true for its boundary  $\partial\mathbf{B}$ . In fact, it is not possible to enforce smoothing second-order constraints on  $\partial\mathbf{B}$  along with point positions in Equation 4.3, therefore only  $C_0$  continuity can be ensured on  $\partial\mathbf{B}$ . In order to overcome this limitation, we proposed [4] to specify such constraint by means of parameter  $\lambda$  in Equation 4.4. Let's assume that subset  $\mathbf{B}$  is defined as

$$\mathbf{B} = \{\mathbf{x} \in \mathbf{S} : |\mathbf{x} - \mathbf{c}|^2 \leq r^2\} \quad (4.7)$$

so that  $\mathbf{B}$  is the intersection of surface  $\mathbf{S}$  with a sphere of center  $\mathbf{c}$  and radius  $r$ , which can be easily specified interactively in 3D. Local feature removal scheme can then be written as

$$\mathbf{S}^{n+1} = \mathbf{S}^n + \lambda(\mathbf{S}^n) \Delta \mathbf{S}^n \Delta t \quad (4.8)$$

where  $\Delta \mathbf{x}$  is approximated by the umbrella operator, and  $\lambda(\mathbf{x})$  is expressed by

$$\lambda(\mathbf{x}) = \begin{cases} 0 & \text{if } |\mathbf{x} - \mathbf{c}|^2 > r^2 \\ \lambda_0 \left( \frac{|\mathbf{x} - \mathbf{c}|^2 - r^2}{r^2} \right)^2 & \text{if } |\mathbf{x} - \mathbf{c}|^2 \leq r^2 \end{cases} \quad (4.9)$$

This way smoothing amount decays smoothly from a maximum near the center of the sphere, to zero at  $\partial\mathbf{B}$ . Even if equilibrium configuration resulting from the scheme in Equation 4.8 is the same of that resulting from the scheme in Equation

---

<sup>1</sup>Better approximations of Laplace operator have been given, for example, in [116, 34], with scale-dependent volume-preserving schemes.

4.4, since it equally derives from the membrane energy functional minimization, in practice, by stopping surface evolution before equilibrium, this scheme is capable to yield smooth results also in correspondence of  $\partial\mathbf{B}'$  (see Figure 4.4). The stopping criterion can be based on maximum speed of mesh vertices. In this context the employment of the umbrella operator for Laplacian discretization is convenient, because it tends to uniform vertex spacing, therefore regularizing the mesh inside  $\mathbf{B}'$ . Instead of fully explicit time integration scheme expressed in Equation 4.8,

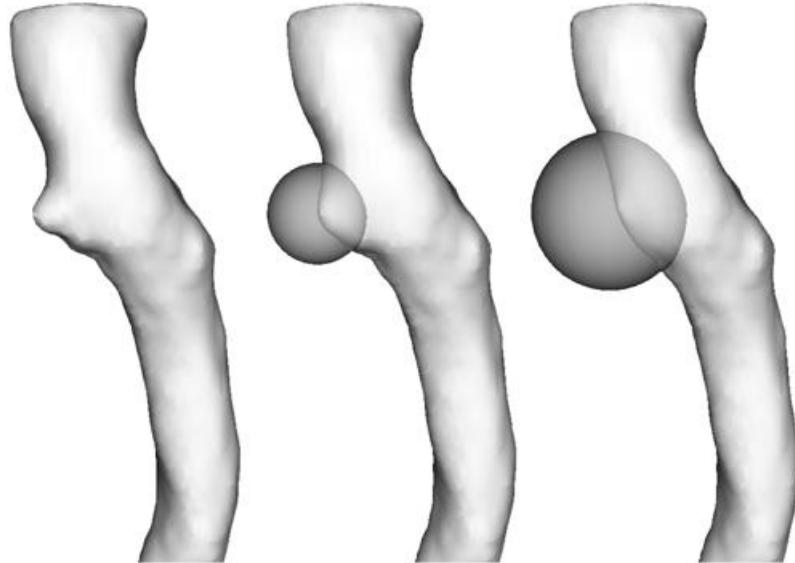


Figure 4.4: Application of weighted Laplacian filtering to surface bulging fairing.

Gauss-Seidel update scheme can be employed, in which updated mesh vertex positions are used as soon as they are computed to update the remaining vertices. This scheme is easy to implement and memory efficient when dealing with meshes<sup>2</sup>.

### 4.3 Bi-Laplacian mesh filtering

Weighted Laplacian surface feature removal scheme, although it can work well in practice, has the limitation that the desired result doesn't correspond to equilibrium of the update equation. In fact, stopping criteria must be employed to interrupt the evolution before the minimum of energy in Equation 4.1 is reached. In order to achieve smoothness and minimal frequencies over  $\mathbf{B}'$ , including  $\partial\mathbf{B}$ , at equilibrium, higher-order schemes must be taken into account, so that it is possible to impose smoothing constraints on  $\partial\mathbf{B}$  in the form of boundary conditions. In particular, it is at least necessary to achieve  $G^1$  continuity at the boundary  $\partial\mathbf{B}$ , which means

---

<sup>2</sup>Unconditional stable mesh filtering was achieved by [34] with the use of implicit schemes

that point positions and surface normals are assigned at the boundary and vary continuously on the surface at equilibrium [104].

### 4.3.1 Minimal energy surfaces

An exact solution to the problem of local surface fairing which ensures  $G^1$  continuity is given by *minimal energy surfaces*, which derive from the minimization the following total curvature energy functional

$$\mathcal{E}_{totcurv}(\mathbf{S}) = \int_{\Omega} (\kappa_1^2 + \kappa_2^2) d\Omega \quad (4.10)$$

where  $\kappa_1$  and  $\kappa_2$  are surface principal curvatures, defined as the minimum and maximum values of normal curvatures. Normal curvatures at one point are the curvatures of the curve defined by the intersection of the surface with a plane perpendicular to each direction belonging to the tangent plane and passing in that point. Since curvature is a second-order differential property, minimization of  $\mathcal{E}_{totcurv}(\mathbf{S})$  leads to a fourth-order problem on the surface, with  $G^1$  conditions on the boundary. The main problem in minimizing  $\mathcal{E}_{totcurv}(\mathbf{S})$  is that principal curvatures  $\kappa_1$  and  $\kappa_2$  depend nonlinearly on surface  $\mathbf{S}$ .

It can be demonstrated [47, 103] that minimum of the following functional

$$\mathcal{E}_{bilaplace}(\mathbf{S}) = \int_{\Omega} (\Delta_B \mathbf{S})^2 d\Omega \quad (4.11)$$

is a surface which is close to a minimal energy surface. The associated Euler-Lagrange equation (see Appendix A) is the bi-Laplace-Beltrami equation

$$\Delta_B \Delta_B \mathbf{S} = 0 \quad (4.12)$$

As shown in the following, the advantage in taking into account the functional in Equation 4.11 is that the associated nonlinearity is easier to handle thanks to the possibility of factorizing this fourth-order problem in Equation 4.12 into two second-order subproblems, as shown further. Laplace-Beltrami operator  $\Delta_B$  in Equation 4.12 is the extension of the Laplacian operator over surfaces when the underlying parameterization is not necessarily assumed orthonormal. Finding an orthonormal parameterization for a generic surface is not a trivial problem, so this case must be taken into account if accurate calculation must be performed on surfaces. If surface parameterization is not orthonormal, it is not true that the infinitesimal arc length over the surface can be expressed by  $dl^2 = dr^2 + ds^2$ , therefore the metric under which differential quantities must be evaluated is not Euclidean, but Riemannian. The general expression of differential operators in Riemannian metric would go beyond the scope of this chapter<sup>3</sup>. However, a fundamental result is that Laplace-Beltrami operator can be expressed in terms of intrinsic surface properties, with no assumptions on the underlying parameterization. In fact, it can be demonstrated that

$$\Delta_B \mathbf{S} = 2H(\mathbf{S})\mathbf{n}(\mathbf{S}) \quad (4.13)$$

---

<sup>3</sup>For deeper insight on this topic, see [47]

where  $H(\mathbf{S})$  is surface mean curvature, and  $\mathbf{n}(\mathbf{S})$  is surface normal. Mean curvature, as pointed out previously, is the mean value of normal curvatures on the tangent plane, or, equivalently, the mean value of the principal curvatures. In the following we will denote the vector  $H(\mathbf{S})\mathbf{n}(\mathbf{S})$  as *mean curvature vector*. Equation 4.12 allows us to rewrite Equation 4.12 as

$$\Delta_B(H(\mathbf{S})\mathbf{n}(\mathbf{S})) = 0 \quad (4.14)$$

which gives us a physical interpretation of Equation 4.12 as of a diffusion equation for mean curvature vectors on  $\mathbf{B}$ .

### 4.3.2 Approximation of curvature on piecewise linear surfaces

The first problem to solve for approximating Equation 4.14 on piecewise linear surfaces is to find an expression for mean curvature vector in a discrete setting, similarly to that in Equation 4.5. This can be done accurately with a mixed finite elements/finite volume approach, described in the following, as proposed by [74].

Since mean curvature is a second-order property and piecewise linear surfaces are obviously  $C^0$ , it is necessary to define discrete versions of differential operators as *spatial averages* of their continuous counterparts in such a way that they converge to the continuous versions as mesh density goes to infinity. By defining a control surface patch  $\mathbf{A}_C(\mathbf{x})$  (the equivalent of control volumes in finite volume techniques) around point  $\mathbf{x} \in \mathbf{S}$ , the locally averaged version of mean curvature vector can be expressed as

$$\hat{H}(\mathbf{x})\hat{\mathbf{n}}(\mathbf{x}) = \frac{1}{|\mathbf{A}_C(\mathbf{x})|} \int_{\mathbf{A}_C(\mathbf{x})} H(\mathbf{x})\mathbf{n}(\mathbf{x}) d\Omega \quad (4.15)$$

where  $|\mathbf{A}_C(\mathbf{x})|$  is the area of  $\mathbf{A}_C(\mathbf{x})$ . Dealing with surface meshes made up of linear triangular elements, definition of control surface patches  $\mathbf{A}_C$  can be performed involving the 1-neighborhood  $N_1(i)$  of vertex  $x_i$ , in such a way that the tiling of control patches over the whole surface doesn't present patch overlapping, so that each surface point is taken into account once. This is obtained by building piecewise linear surface patch boundaries connecting midpoints of the edges  $x_i x_j$ , with  $x_j \in N_1(i)$ , to a point internal to each of the triangles sharing  $x_i x_j$  (see Figure 4.5). For each triangle, the internal point can be for example defined as its circumcenter or barycenter. It is possible to demonstrate that the choice of circumcenters yields optimal approximation for non-obtuse triangulations [74]. However, since the circumcenter of obtuse triangles falls outside their boundary, circumcenter itself is not suitable for generic triangulations. Good results are obtained using circumcenter as the internal point for non-obtuse triangles, and midpoint of the edge opposite to the obtuse angle for obtuse triangles, as shown in Figure 4.6.

Since each triangle defines a locally orthonormal surface metric, the Laplace-Beltrami operator can be locally approximated by a Laplacian over each triangle of the control surface patch, so that

$$\int_{\mathbf{A}_C(\mathbf{x})} \Delta_B \mathbf{x} d\Omega = \int_{\mathbf{A}_C(\mathbf{x})} \Delta_{r,s} \mathbf{x} dr ds \quad (4.16)$$

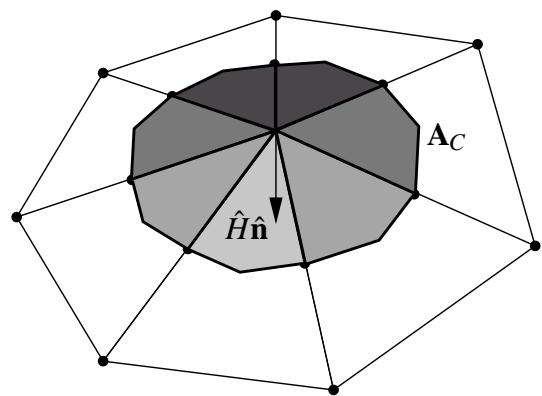


Figure 4.5: Example of surface patch defined on the 1-neighborhood.

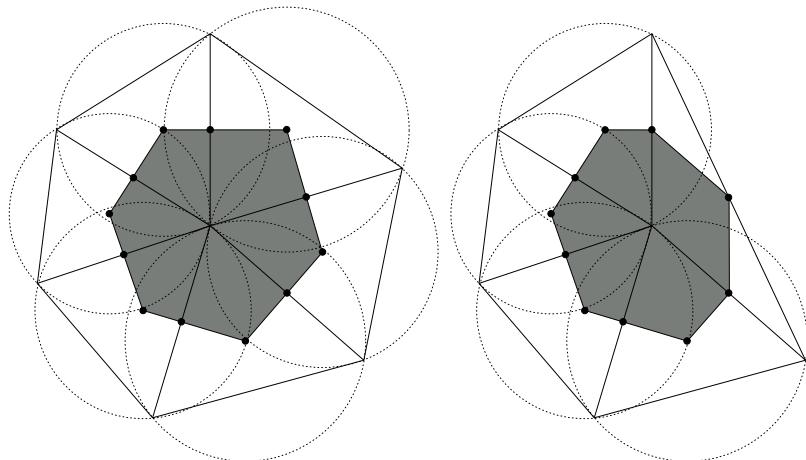


Figure 4.6: Left: surface patch for a 1-neighborhood made up of non-obtuse triangles; internal points are triangle circumcenters. Right: surface patch for a 1-neighborhood in which one triangle is obtuse; for that triangle the internal point is the midpoint of the edge facing the obtuse angle.

where  $\Delta_{r,s}\mathbf{x} = \mathbf{x}_{rr} + \mathbf{x}_{ss}$  is the Laplacian operator calculated with respect to the local parameters  $(r, s)$ . By means of Gauss theorem, the right-hand side surface integral in Equation 4.16 can be turned into a line integral over the boundary of control surface patches:

$$\int_{\mathbf{A}_C(\mathbf{x})} \Delta_{r,s}\mathbf{x} \, dr \, ds = \int_{\partial \mathbf{A}_C(\mathbf{x})} \nabla_{r,s}\mathbf{x} \cdot \mathbf{n}_{r,s} \, dl \quad (4.17)$$

where  $\mathbf{n}_{r,s}$  is the normal to control surface patch boundary with respect to the local parameter space. Since piecewise linear discretization of the surface is assumed,  $\nabla_{r,s}\mathbf{x}$  is constant over each triangle, and line integral of right-hand side integral of Equation 4.17 results in

$$\int_{\partial \mathbf{A}_C(\mathbf{x})} \nabla_{r,s}\mathbf{x} \cdot \mathbf{n}_{r,s} \, dl = \frac{1}{2} |\mathbf{x}_j - \mathbf{x}_{j+1}| \nabla_{r,s}\mathbf{x} \cdot \mathbf{n}_{r,s}^{x_j x_{j+1}} \quad (4.18)$$

where  $\mathbf{n}_{r,s}^{x_j x_{j+1}}$  is the normal to edge  $x_j x_{j+1}$ . This expression doesn't depend on the choice of the point internal to triangles for control surface patch definition (e.g. barycenter or circumcenter), provided that control surface patch boundary passes through the midpoints of edges  $x_i x_j$  and  $x_i x_{j+1}$  of triangle  $x_j x_i x_{j+1}$ . Further development of the right-hand side of Equation 4.18 leads to the following expression

$$\int_{\partial \mathbf{A}_C(\mathbf{x})} \nabla_{r,s}\mathbf{x} \cdot \mathbf{n}_{r,s} \, dl = \frac{1}{2} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (\mathbf{x}_j - \mathbf{x}_i) \quad (4.19)$$

where  $\alpha_{ij}$  and  $\beta_{ij}$  are the angles of triangles  $x_{j-1} x_i x_j$  and  $x_j x_i x_{j+1}$  opposed to edge  $x_i x_j$  (i.e. angles  $x_j x_{j-1} x_i$  and  $x_i x_{j+1} x_j$ , see Figure 4.7). Again, this expression is independent from the choice of internal points.

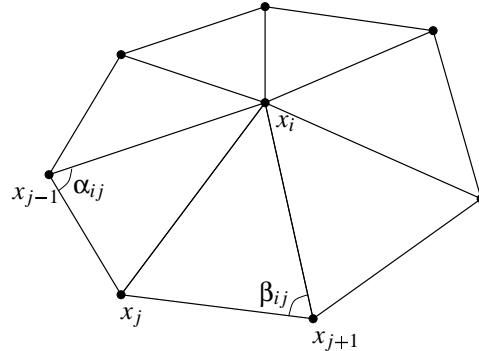


Figure 4.7: Quantities for the computation of the discrete Laplace-Beltrami operator.

From Equation 4.19 we can finally derive a discretized expression for the Laplace-Beltrami operator (and consequently for mean curvature vector)

$$\Delta_B \mathbf{x} \approx \frac{1}{2|\mathbf{A}_C(x_i)|} \sum_{j \in N_1(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (\mathbf{x}_j - \mathbf{x}_i) \quad (4.20)$$

where vertex  $x_i$  is located in  $\mathbf{x}_i = \mathbf{x}$ , and  $|\mathbf{A}_C(x_i)|$  is control surface patch area, which is computed summing the contributions of each triangle  $T_j$  in the neighborhood. The area  $|\mathbf{A}_{C_j}|$ , which is the area of the portion of the surface patch relative to

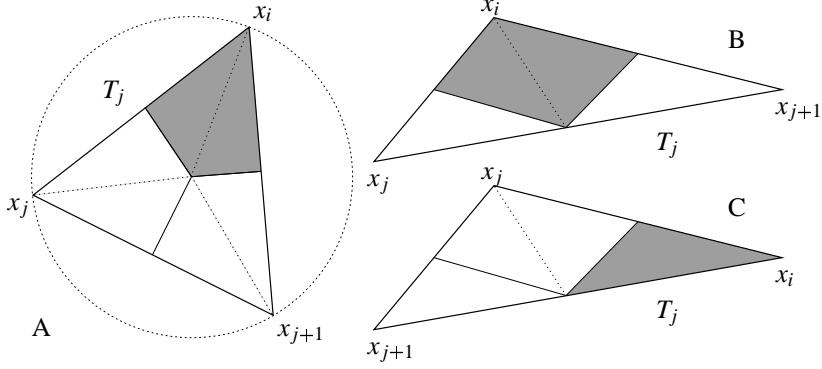


Figure 4.8: Triangle patch area for non-obtuse triangle (A), obtuse triangle with obtuse angle in  $x_i$  (B), obtuse triangle with obtuse angle not in  $x_i$  (C).

triangle  $T_j$ , has the following expressions, according to the choice of internal points (refer to Figure 4.8)

$$|\mathbf{A}_{C_j}| = \frac{1}{8} (\cot(x_{j+1}\hat{x}_jx_i)|\mathbf{x}_{j+1} - \mathbf{x}_i|^2 + \cot(x_ix_{j+1}x_j)|\mathbf{x}_j - \mathbf{x}_i|^2) \quad (4.21)$$

if  $T_j$  is non-obtuse (i.e. circumcenter is used as the internal point) (see Figure 4.8 A),

$$|\mathbf{A}_{C_j}| = \frac{|T_j|}{2} \quad (4.22)$$

if  $T_j$  is obtuse in  $i$  (see Figure 4.8 B), and

$$|\mathbf{A}_{C_j}| = \frac{|T_j|}{4} \quad (4.23)$$

if  $T_j$  is obtuse, but not in  $i$  (in both last two cases the midpoint of the edge opposite to the obtuse angle is used as the internal point) (see Figure 4.8 C). Therefore,  $|\mathbf{A}_{C_j}|$  depends on the choice of internal points, therefore affecting the quality of the discretization.

### 4.3.3 Numerical solution by bi-Laplacian factorization

The discrete expression for Laplace-Beltrami operator over piecewise linear surfaces given in Equation 4.20, although simple to implement, is nonlinear, therefore a numerical solution of the fourth-order bi-Laplacian problem in Equation 4.12 cannot be obtained directly. A possible approach is instead to factorize the fourth-order problem into two coupled second-order subproblems to be solved iteratively

[104]. The idea is to calculate mean curvature vectors for the initial surface, solve some steps of the diffusion equation of mean curvature vectors on the fixed initial surface and then to update initial surface position so that mean curvature vectors of the resulting surface coincide with the diffused mean curvature vectors, repeating this procedure iteratively until convergence. In a mathematical framework this translates into

$$\begin{cases} \Delta_B(\tilde{H}(\mathbf{S}^k)\tilde{\mathbf{n}}(\mathbf{S}^k)) = 0 & n \text{ steps} \\ H(\mathbf{S}^{k+1})\mathbf{n}(\mathbf{S}^{k+1}) = \tilde{H}(\mathbf{S}^k)\tilde{\mathbf{n}}(\mathbf{S}^k) & m \text{ steps} \end{cases} \quad (4.24)$$

where  $\tilde{H}(\mathbf{S}^k)\tilde{\mathbf{n}}(\mathbf{S}^k)$  is the vector field which initially coincides with mean curvature vectors of surface  $\mathbf{S}^k$ , and subsequently undergoes  $n$  steps of diffusion. Both mean curvature vectors and Laplace-Beltrami operator for the diffusion equation are approximated with Equation 4.20. Afterward,  $m$  steps of surface position update are performed to make mean curvature vector distribution approach to the diffused vector field  $\tilde{H}(\mathbf{S}^k)\tilde{\mathbf{n}}(\mathbf{S}^k)$ . Surface update must be performed iteratively since, as noted previously, the expression for mean curvature vector is nonlinear. Instead of a fixed number of point update steps, point update can be performed until the maximum update amount falls below a specified threshold. In order to create minimal distortion of surface elements, at expense of convergence speed, point update is performed iteratively along surface normals

$$\mathbf{x}_i^{h+1} = \mathbf{x}_i^h + t^h \mathbf{n}(\mathbf{x}_i^h) \quad (4.25)$$

where  $t$  is the update amount for each surface point  $\mathbf{x}_i$  at point update iteration  $h$ . Update amount can be expressed as

$$t^h = \frac{4|\mathbf{A}_C(x_i^h)|}{\sum_{j \in N_1(i)} (\cot \alpha_{ij}^h + \cot \beta_{ij}^h)} \left[ H(\mathbf{x}_i^h)\mathbf{n}(\mathbf{x}_i^h) - \tilde{H}(\mathbf{x}_i^h)\tilde{\mathbf{n}}(\mathbf{x}_i^h) \right] \cdot \mathbf{n}(\mathbf{x}_i^h) \quad (4.26)$$

that is the projection of the difference between mean curvature vector of the updated surface at update step  $h$  and diffused mean curvature vector along surface normal, scaled with a neighborhood-dependent weight [6]. If we compute the mean curvature vector for point  $\mathbf{x}^{h+1}$  determined with Equation 4.25 using update amount in Equation 4.26, we obtain

$$\begin{aligned} H(\mathbf{x}_i^{h+1})\mathbf{n}(\mathbf{x}_i^{h+1}) &= \frac{1}{4|\mathbf{A}_C(x_i^{h+1})|} \sum_{j \in N_1(i)} \left( \cot \alpha_{ij}^{h+1} + \cot \beta_{ij}^{h+1} \right) \left( \mathbf{x}_j^h - \mathbf{x}_i^h - t^h \mathbf{n}(\mathbf{x}_i^h) \right) \\ &\approx \left[ \tilde{H}(\mathbf{x}_i^h)\tilde{\mathbf{n}}(\mathbf{x}_i^h) \cdot \mathbf{n}(\mathbf{x}_i^h) \right] \mathbf{n}(\mathbf{x}_i^h) \end{aligned} \quad (4.27)$$

where the last approximation holds in the hypothesis that  $t^h$  is small, so that  $|\mathbf{A}_C(x_i)|$ ,  $\alpha_{ij}$  and  $\beta_{ij}$  do not change much between the  $h$ -th and the  $h+1$ -th update iteration, since point  $\mathbf{x}_i$  is constrained to move along the normal direction. Apparently it would be possible to achieve  $H(\mathbf{x}_i^{h+1})\mathbf{n}(\mathbf{x}_i^{h+1}) \approx \tilde{H}(\mathbf{x}_i^h)\tilde{\mathbf{n}}(\mathbf{x}_i^h)$  if the update in Equation 4.25 was not projected along the normal direction, but in this case point  $\mathbf{x}_i$

would move on its neighborhood plane, and the hypothesis of small variations on angles  $\alpha_{ij}^h$  and  $\beta_{ij}^h$  would not hold anymore.

We chose to adopt an explicit scheme for the iterative solution of the two coupled subproblems. This allows us to take advantage of the CFL condition in order to adaptively select the update step such that stability is guaranteed. The CFL condition of unstructured meshes is written as

$$\Delta t \leq \Delta t_{CFL} = \frac{\min\{\Delta x\}}{\max\{u\}} \quad (4.28)$$

where  $\Delta t$  is the update step,  $\Delta x$  is the length of a mesh edge, and  $u$  is solution update at a mesh point. Typically we choose the update step as a fraction of the limit update step, e.g.  $\Delta t = 0.1\Delta t_{CFL}$ .

Convergence is detected either by curvature diffusion residuals or magnitude of point update  $t^h$  at the beginning of point update subiterations. In practice, numerical experiments have shown that convergence detection is better if curvature diffusion residuals are evaluated in  $L^2$  norm, in order to smooth eventual numerical problems caused by bad-shaped elements, while magnitude of point update, being more regular, can be evaluated in  $L^\infty$  norm.

## 4.4 Applications

We will now describe the possible applications of surface fairing to blood vessel modeling. The first two applications will be used in practice for vessel modeling throughout this work, while the third represents a possible direction for future development.

### 4.4.1 Local smoothing and bulging removal

As introduced previously, the immediate application of surface fairing techniques to vascular modeling is represented by local surface enhancement for models obtained with level sets or balloon evolution [6]. Local enhancement is required, for example, when angiographic image acquisition artifacts produce local changes in image contrast of a scale similar to that of the structures of interest, so that global smoothing is inadequate. A similar problem is that of surface bulging produced when deformable models seep into branching vessels which are not being reconstructed. Branching vessels can be ignored when they have a small diameter with respect to acquisition resolution, so that reconstruction is not possible, or when plausible boundary conditions are not available for a subsequent CFD analysis and the associated flow features are of little interest. An example of the first case is the abdominal aorta, where the inferior mesenteric artery is often not reconstructible from MR angiography images due to its dimensions. The second case can be encountered at the branching vessels of the external carotid artery, the superior thyroid artery, the lingual artery, the occipital artery and the external maxillary artery.

Usually boundary conditions for the carotid bifurcation are expressed as the flow rate or velocity profile at the common carotid artery inlet, and flow rate repartition between internal and external carotid arteries. It has been demonstrated [138] that removing the superior thyroid artery, which is the nearest to the carotid bifurcation among external carotid artery branching vessels, produces small effects on flow field computation in the region of carotid bifurcation, carotid bulb and proximal internal carotid artery, usually the regions of interest for hemodynamics modeling.



Figure 4.9: Application of bi-Laplacian filtering to surface bulging fairing.

The selection of the surface region which undergoes fairing can be performed interactively using a sphere function defined directly on the 3D model by specifying its center and radius. Figure 4.9 shows the application of bi-Laplacian fairing to an external carotid bulging caused by an unreconstructed branching vessel.

#### 4.4.2 Inlet and outlet extensions

The problem of inlet/outlet regions has been introduced at the beginning of this chapter, and consists in substituting the rounded-shape closed extremities of the models obtained with level sets with well-shaped open cylindrical prolongations, in order to express analytical boundary conditions and let the computed flow field solution develop from those boundary conditions before entering the true vessel

domain. A solution to such a problem is to clip the extremities of the models, therefore opening the closed surface of the model, attach cylindrical prolongations at the model openings, and then fair the added cylindrical surface, in order to create a smooth transition from the true model to the final circular section and produce minimal perturbation on the computed flow field.

Clipping of the extremities is performed using an interactive version of the algorithm described in the previous chapter. A probing sphere is interactively initialized on the surface around the inlet/outlet extremity, and the selection is propagated to neighbor points by moving the sphere on its intersection with surface for a user-defined number of steps. When the desired vessel portion has been selected and removed, cylindrical prolongations are constructed with orientation and radius estimated from the last sphere step, and length specified by the user in terms of a multiple of the radius. This way surface selection and cylinder definition are easily performed directly in 3D. The constructed cylindrical prolongation is finally faired with  $G^1$  boundary conditions on the clipping profile and on the outermost circular section of the prolongation [6]. In Figure 4.10 an example of the construction of cylindrical extensions is shown. In Figure 4.11, a more challenging situation is successfully handled by the algorithm. A problem of using the fairing functional in Equation 4.11 in this context, where a cylindrical surface is faired with  $G^1$  boundary condition at the two outermost sections, is that minimization of  $\mathcal{E}_{totcurv}$  cannot yield cylinders. In fact, minimization of  $\mathcal{E}_{totcurv}$  tends to penalize both principal curvatures, while on the surface of a cylinder the minimum principal curvature is always zero and the maximum is equal to the inverse of the radius. As a result, the cylinder tends to shrink as  $\mathcal{E}_{totcurv}$  is being minimized. This effect can be effectively limited adopting a low number of subiterations for the curvature diffusion subproblem, so that high frequency mesh components are faired, while the lower ones are preserved. However, the adoption of a fairness functional which can guarantee  $G^1$  boundary conditions while preserving cylindrical shapes constitutes an important point for future development.

#### 4.4.3 Virtual intervention

The last application of surface fairing doesn't play a direct role in the present work. Nevertheless it is introduced here because it is closely related to the problem of patient-specific vessel modeling. One of the issues arising in vascular surgery is to find an optimal configuration of artificial or native grafts employed when substituting a vascular tract or creating a bypass. The resulting geometric configuration and its effects on blood flow dynamics can heavily affect the outcome of the intervention, and eventually lead to failure. For this reason, in recent years, a number of collaborations between vascular surgeons and bioengineering groups has led to the study of fluid dynamics after simulated intervention. Virtual intervention was usually performed either using fully idealized models or by editing patient-specific models using classical CAD tools. Surface fairing can be successfully employed in this context to enhance the blending between native and edited parts of the model,

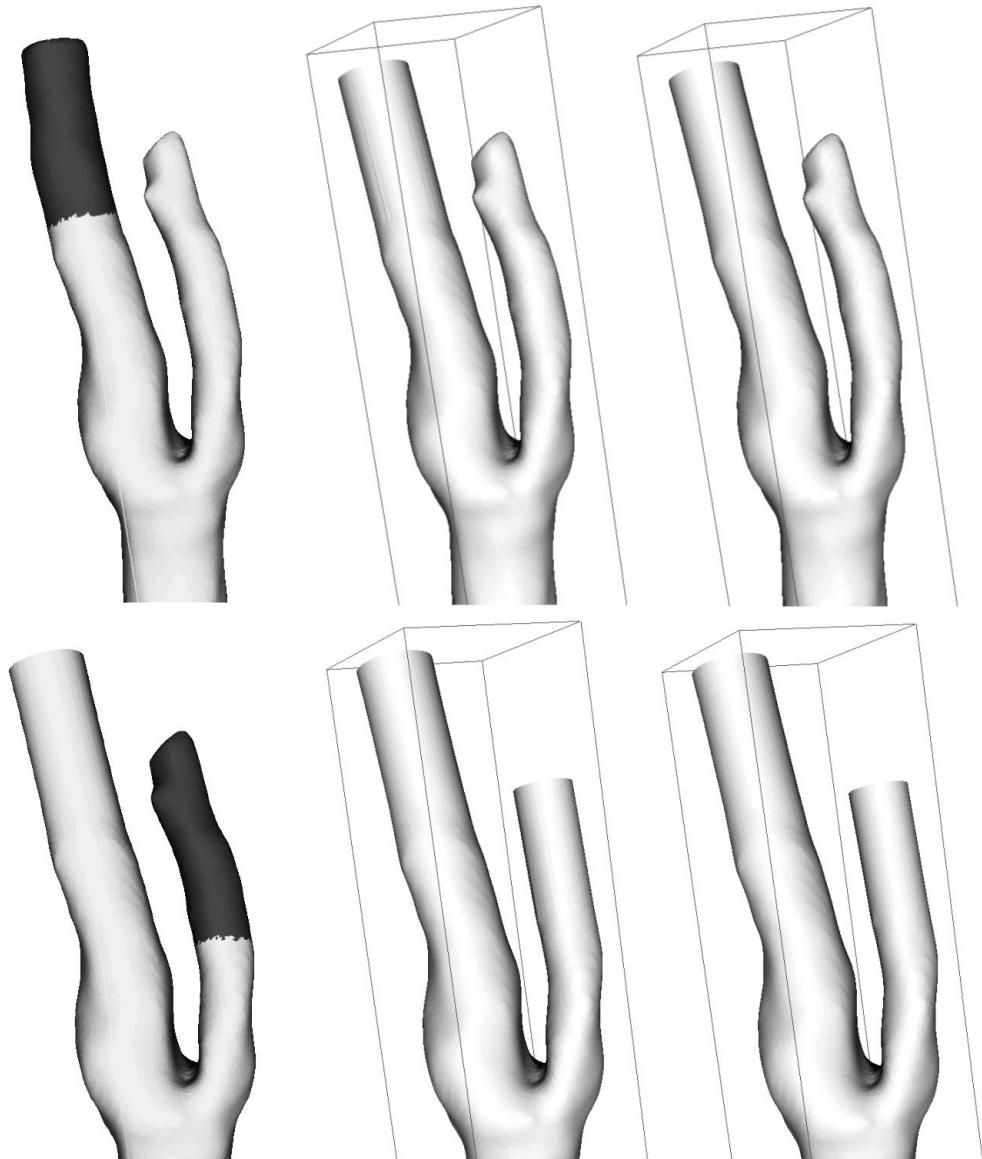


Figure 4.10: Construction of open extensions on a model obtained by level sets evolution. Left: interactive selection of the surface to be removed. Middle: automatic construction of cylindrical prolongations. Right: fairing of the prolongations to ensure smooth transition with the surface of the model.

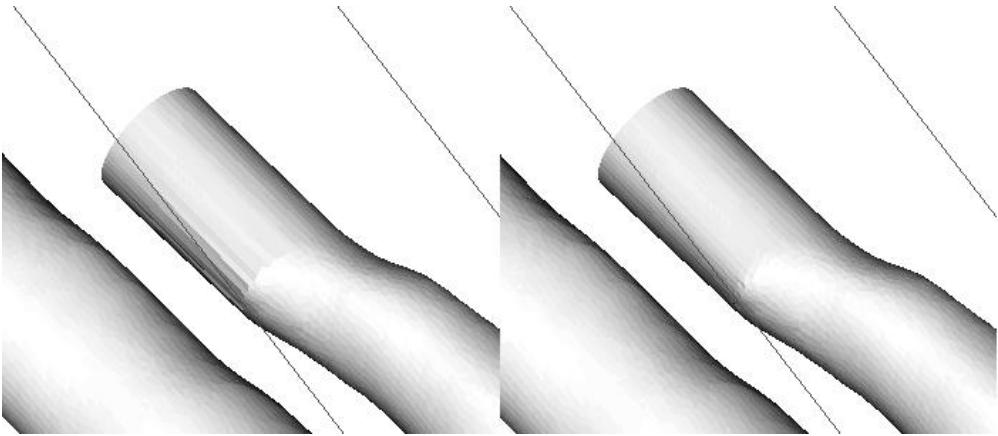


Figure 4.11: Construction and result of fairing of a badly oriented cylinder.

or even to find the minimal energy configuration for the artificial tracts, to yield more realistic results than that obtained by analytical geometry definition. As just pointed out, an important enhancement required for this application is the adoption of cylinder-preserving fairing functionals, as well as the use of implicit schemes for the resolution of linear systems [34]. Being unconditionally stable, implicit schemes allow to model large displacements without constraints on the update step, achieving robust real-time interaction.

## 4.5 Implementation

Both weighted Laplacian filtering and bi-Laplacian fairing were implemented in C++ using VTK data structures (`vtkPolyData`), as subclasses of `vtkPolyDataToPolyDataFilter`.

Update step for explicit schemes is chosen as 0.1 times the limit update step computed from the CFL condition. A fixed number of subiterations are imposed for the mean curvature vector diffusion step, while point update subiterations are stopped when point update convergence is detected. Global convergence is detected with  $L^\infty$  norm of point update at the beginning of point update subiterations. The value assumed for both convergence criteria is  $10^{-3}$  times the shortest mesh edge.

## 4.6 Validation

We validated the bi-Laplacian fairing technique on analytical geometries created with solid modeling tools. A subset of a sphere and a cylinder have been first generated, and spherical bulgings have been added, as shown in Figure 4.12. The sharp intersections between the bulgings and the analytical surfaces were maintained, so

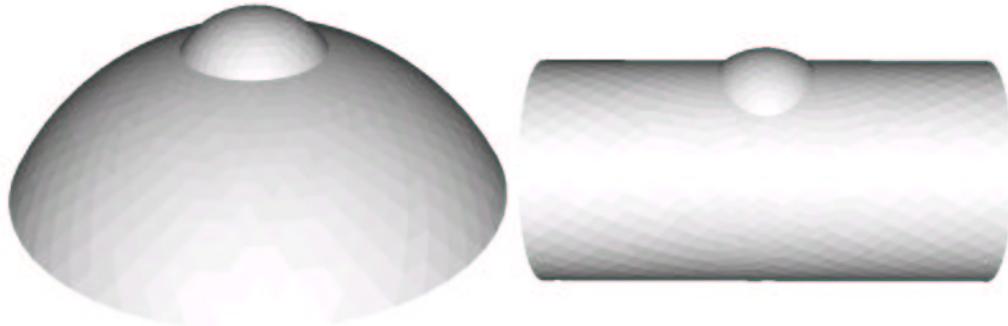


Figure 4.12: Geometries used for bi-Laplacian fairing algorithm validation.

to test algorithm robustness. As shown in Figures 4.13 and 4.14, the bulging region was selected interactively, and the algorithm was run without adjusting parameters or convergence tolerance with respect to what have been set on the basis of real-world cases. In the sphere subset case, experiments were performed selecting both the region around the bulging and the whole surface. Three different surface mesh densities were generated by specifying different number of nodes on bulging boundary. Table 4.1 shows the results of validation on such geometries expressed as maximum percent error of point positions of the faired surface with respect to the radius of the analytical geometry.

Dataset	Mesh density	Max error %
A	10	0.42
	20	0.18
	30	0.40
B	10	2.53
	20	0.77
	30	1.34
C	10	1.07
	20	0.56
	30	0.24

Table 4.1: Validation results for the bi-Laplacian fairing algorithm, expressed as maximum percent error of point positions of the faired surface with respect to the radius of the analytical geometry. Dataset A: sphere subset with bulging region selected. Dataset B: sphere subset with whole surface selected. Dataset C: cylinder with bulging region selected.

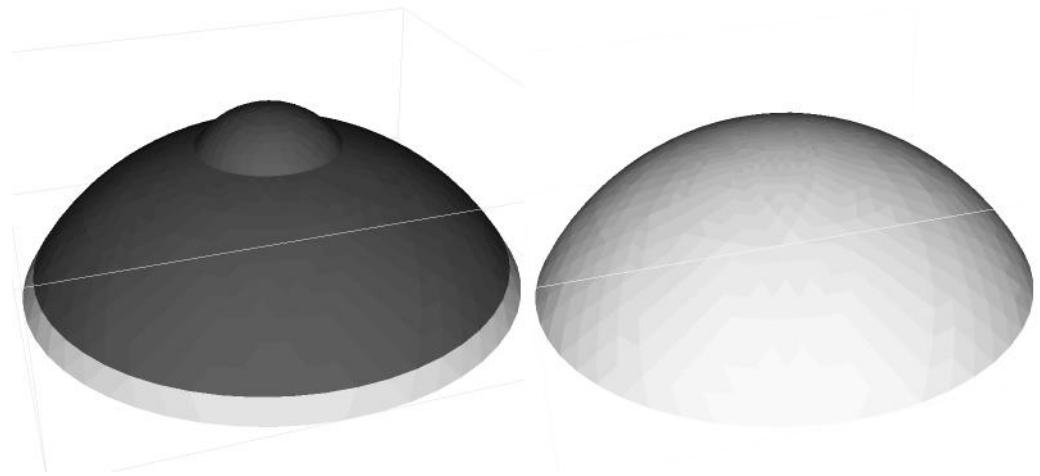


Figure 4.13: Bi-Laplacian fairing performed on the sphere subset dataset with denser mesh. Left: interactive selection of the faired region. Right: results of fairing.

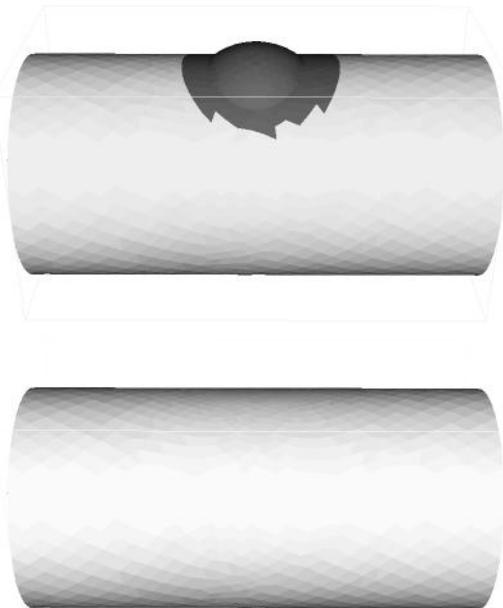


Figure 4.14: Bi-Laplacian fairing performed on the cylinder dataset with denser mesh. Left: interactive selection of the faired region. Right: results of fairing.



## Chapter 5

# Geometric characterization of vascular structures

### 5.1 Introduction

Given the influence of geometry on fluid dynamics and the great inter-individual anatomic variability of blood vessels, characterization of blood vessel geometry is an important step toward a better understanding of the factors involved in vascular pathology. This kind of analysis must be performed at different levels of detail. At the higher level, synthetic parameters quantifying geometric features, such as stenosis grade, are required in diagnostic imaging. At a middle level, vessel axis definition and local radius measurement are important to synthesize the intrinsic information contained in vessel surface, so to extend geometric description to populations and study how different vessel global configurations undergo different histories. At the finer level, local geometric characterization of vessel surface would help to document the progression of vascular disease and to look for typical local geometric features associated with disease initiation. Several works have been proposed for centerline extraction and radius measurement from angiographic images [36, 43, 136], or for local geometric characterization of shapes [114], but, to our knowledge, no consistent approach which covers all the levels from centerline definition to local surface characterization of generic vascular structures has been proposed.

Geometric analysis of vascular structures can be related to the problem of describing interconnecting  $2\frac{1}{2}$ D geometries. We refer to  $2\frac{1}{2}$ D geometries as to volumes which can be obtained by sweeping a 2D face of variable shape along a 1D path. Single  $2\frac{1}{2}$ D structures, such as single vessels without any branching, can be easily described as generalized cylinders by means of local cylindrical coordinate systems defined along vessel centerline. Although it is possible to extend generalized cylinder description to interconnecting  $2\frac{1}{2}$ D geometries by splitting them into single tubular pieces, in practice it is difficult to set up general rules to define such splitting for real vascular structures. Moreover, the results are dependent

on the chosen splitting strategy. One last issue regarding cylindrical systems is that vascular structures can exhibit geometric features which go beyond the definition of  $2\frac{1}{2}$ D geometry, in particular when dealing with diseased tracts presenting aneurysms, plaques and stenoses.

For these reasons we set up an original method to describe vessel structures which has no limitations regarding branching geometries, and which allows to obtain information at different levels of descriptions, from centerlines to local geometric features, in a consistent way. Moreover, the hypothesis of dealing with interconnecting  $2\frac{1}{2}$ D structures is used here, but more general shape description criteria can be easily included in order to deal with full 3D geometries.

Conversely to most centerline extraction approaches which directly deal with 3D images, our starting point is a polygonal surface representing vascular wall, obtained here from level sets evolution and eventual local fairing and editing. The reason of this choice is that this way we uncouple the reconstruction problem from the geometry characterization problem, that local surface characterization can be directly performed for a given surface mesh and that accuracy depends on local mesh resolution, rather than on image resolution.

## 5.2 Medial axis and Voronoi diagram

Boundary surface is not in general a good synthetic descriptor of the shape of a 3D object, although it univoquely identifies the object. For example, occupancy, symmetry and topology are not intrinsic properties of the surface of generic objects. A well known concept in shape description is that of medial axis, a geometric entity dual to the object surface (i.e. medial axis can be computed knowing the surface and vice-versa), represented as a non-manifold<sup>1</sup> surface (see Figure 5.1) located medially inside the object and containing information about object thickness (i.e. distance from boundary) at every medial point. In other words, the medial axis is a way of describing objects which uncouples geometric properties of location and thickness [114]. We decided to use this concept as the basis for the description of branching vessel structures starting from their boundary surface. In this phase the  $2\frac{1}{2}$ D nature of the constituents of vascular geometries will not be taken into account, since the concepts we are introducing are suited for the description of generic objects. Later, in the next section,  $2\frac{1}{2}$ D geometry characterization will be finally performed directly on the medial axis.

In this section, the definition of medial axis of continuous objects will be first introduced. Subsequently, the approximation of the medial axis for a set of points sampling the object surface will be presented, along with an efficient technique for its computation and the characterization of the result.

---

<sup>1</sup>A surface is said to be a 2-manifold if every point has a neighborhood which can be mapped onto a disk. In contrast, a surface is a non-manifold if this property is not satisfied for at least one point. In general, on a non-manifold surface a line can be shared by more than two adjacent surface patches, which typically occurs on the medial axis.

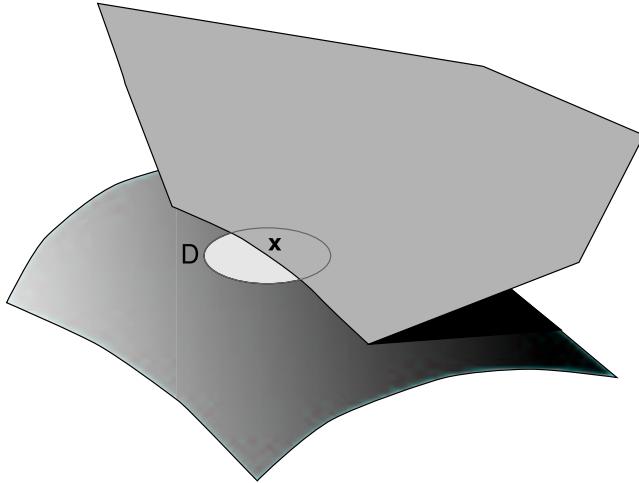


Figure 5.1: A generic non-manifold surface. The neighborhood of point  $\mathbf{x}$  cannot be mapped onto a disk.

### 5.2.1 Medial axis of continuous objects

The definition of medial axis  $\text{MA}(\Omega)$  of an object  $\Omega \subset \mathbb{R}^3$  is commonly given as the locus of centers of maximal spheres inscribed into  $\Omega$  [18, 28]. An inscribed sphere is maximal if it is not strictly contained in any other inscribed sphere (see Figure 5.2). A useful concept involved in the definition of medial axis is that of distance transform [20], which for each object point returns the distance to the nearest boundary point

$$\text{DT}(\mathbf{x}) = \min \{ \text{dist}(\mathbf{x}, \partial\Omega) \} \quad (5.1)$$

where  $\text{dist}(\mathbf{x}, \mathbf{y})$  is the Euclidean distance between  $\mathbf{x}$  and  $\mathbf{y}$ . The medial axis of an object  $\Omega \subset \mathbb{R}^3$  can be identified with the ridges of its distance transform, so that

$$\text{MA}(\Omega) = \{ \mathbf{x} \in \Omega \mid \exists \text{dir } \mathbf{n}, \exists \bar{\epsilon} > 0 : \forall \epsilon \in (0, \bar{\epsilon}), \text{DT}(\mathbf{x}) \geq \text{DT}(\mathbf{x} \pm \epsilon \mathbf{n}) \} \quad (5.2)$$

It can be demonstrated that this definition is equivalent to that referring to maximal spheres. Intuitively, definition in Equation 5.2 states that for each medial axis point there exists a direction along which the distance transform presents a local maximum in that point. Since the distance transform returns the radius of the sphere touching object boundary, this means that along direction  $\mathbf{n}$ , in the vicinity of  $\mathbf{x}$ , the sphere defined on  $\mathbf{x}$  is maximal. Definition in Equation 5.2 also shows that maximal spheres must have more than one contact point, i.e. medial axis points have more than one nearest point on  $\partial\Omega$ .

An important property of medial axis is homotopy, that is  $\Omega$  and  $\text{MA}(\Omega)$  present the same number of connected components, holes and loops [28]. As a consequence, if two pieces of the surface of  $\Omega$  are path-connected, their medial axes will be path-connected.

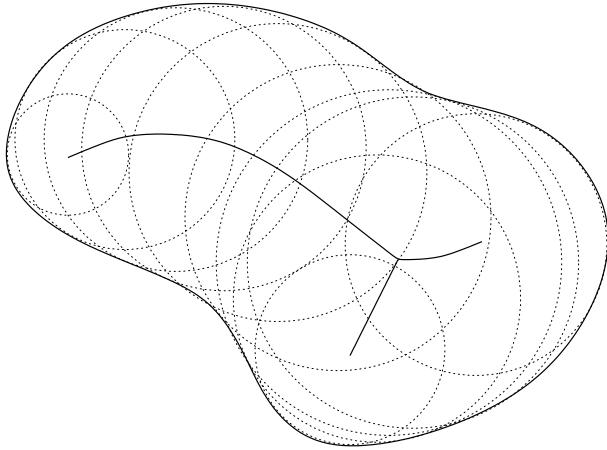


Figure 5.2: Medial axis of a 2D object as the locus of centers of maximal circles (dotted).

### 5.2.2 Voronoi diagram and Delaunay tessellation

The computation of the medial axis of a continuous object is a very complex problem, and an exact solution has been only found for few simple shapes. For generic surfaces, e.g. representing real anatomy, it is therefore necessary to find an approximation of the medial axis which properly balances computational expensiveness and accuracy. In our case, object surface is given in terms of a piecewise linear surface whose vertices are supposed to densely sample an unknown real surface. Although it is possible to compute the medial axis of a polygonal surface taking into account both the vertices and the piecewise linear interpolation between them, this approximation is computationally expensive and some aspects of the results still remain theoretically unexplored. Conversely, a common and theoretically well-established method to approximate the medial axis of an object is to derive it using mesh vertices only, taken as a dense set of disconnected points.

To this purpose, we define the Voronoi diagram<sup>2</sup> of a point set  $P$ , eventually sampling  $\partial\Omega$ , as [12]

$$\text{Vor}(P) = \bigcup_{p \in P} \partial V(p) \quad (5.3)$$

where  $V(p)$  is the Voronoi region associated with point  $p$ , defined as

$$V(p) = \{\mathbf{x} \in \mathbb{R}^3 \mid \text{dist}(\mathbf{p}, \mathbf{x}) \leq \text{dist}(\mathbf{p}, P)\} \quad (5.4)$$

where  $\mathbf{p}$  stands for the position of  $p$ . In other words,  $V(p)$  is the region of  $\mathbb{R}^3$  closer to  $p$  than to any other element of  $P$ . Therefore, points on  $\text{Vor}(P)$  have more than one nearest point in  $P$ , and Voronoi diagram lies on the ridges of the distance

---

<sup>2</sup>In the following we will refer to the 3D case for simplicity, but the same concepts can be generalized to  $n$  dimensions.

transform associated with the point set  $P$ , evidencing the close relation between the Voronoi diagram of  $P$  and the medial axis of  $\Omega$ . In 3D, Voronoi diagrams take the shape of a non-manifold surface made up of convex polygons. See Figure 5.3 for a 2D example.

The Voronoi diagram of a point set  $P$  is dual to the Delaunay tessellation of  $P$ ,  $\text{Del}(P)$ , which is a tetrahedrization of point set  $P$  which satisfies the property that the sphere circumscribed to each tetrahedron does not strictly contain any point of  $P$  [12]. The relationship between Voronoi diagram and Delaunay tetrahedra holds in that Voronoi vertices are the circumcenters of Delaunay tetrahedra, and the spheres circumscribed to Delaunay tetrahedra, called Voronoi spheres, are maximal spheres with respect to point set  $P$ . Furthermore, Voronoi vertex connectivity can be deduced from Delaunay tessellation connecting two circumcenters to form an edge if their associated tetrahedra share a face, and connecting a loop of circumcenters to form a polygon if their associated tetrahedra share an edge [12] (see Figure 5.3). This is actually the way in which Voronoi diagrams are constructed in this work. A method to build the Delaunay tessellation of a set of points is therefore needed at this point. Several algorithms exist to tetrahedrize a point set in 3D such that the Delaunay property is satisfied [42], among which three major methods are usually employed, namely *flipping*, *incremental*, and *random incremental* algorithms. Flipping algorithm starts building an initial tetrahedrization of  $P$  and then iteratively changes local vertex connectivity in order to make tetrahedra locally satisfy Delaunay property. This algorithm was created for 2D case, for which a flip is naturally defined as flip of the diagonal of the quadrilateral formed by two adjacent triangles. Furthermore, it has been demonstrated that the number of flips required in 2D is finite. Its extension to the 3D case requires a more complex definition of flipping, moreover there's no guarantee that the algorithm stops before success (this last limitation was solved by more advanced versions of the algorithm). Incremental algorithm works by first constructing a convex hull made up of a few tetrahedra bounding all the points in  $P$ , and then adding points of  $P$  one by one breaking the tetrahedron which contains the added point and replacing it with a locally Delaunay-satisfying set of tetrahedra. The algorithm is guaranteed to terminate. Furthermore, worst-case computational complexity was demonstrated to be  $O(n^2)$ ,  $n$  being the number of points in  $P$ , but numerical experiments show that for real cases complexity is slightly more than linear. Random incremental algorithm works as the previous algorithm, except that point incremental insertion is performed randomly, with each permutation equally likely. This lowers expected worst-case running time down to  $O(n^{\frac{3}{2}})$ . Thanks to the good trade-off between speed and robustness, the incremental approach has been used here.

It is important to note that the Delaunay tessellation of a point set is always a convex set of tetrahedra. Therefore, building the Delaunay tessellation of a set of points sampling the surface of a generic non-convex object does not produce a tetrahedrization of the inner part of the object only. From the dual point of view, the Voronoi diagram is not defined only inside the object, but in all space. Intuitively, this derives from the fact that sampling points have no connectivity, therefore no

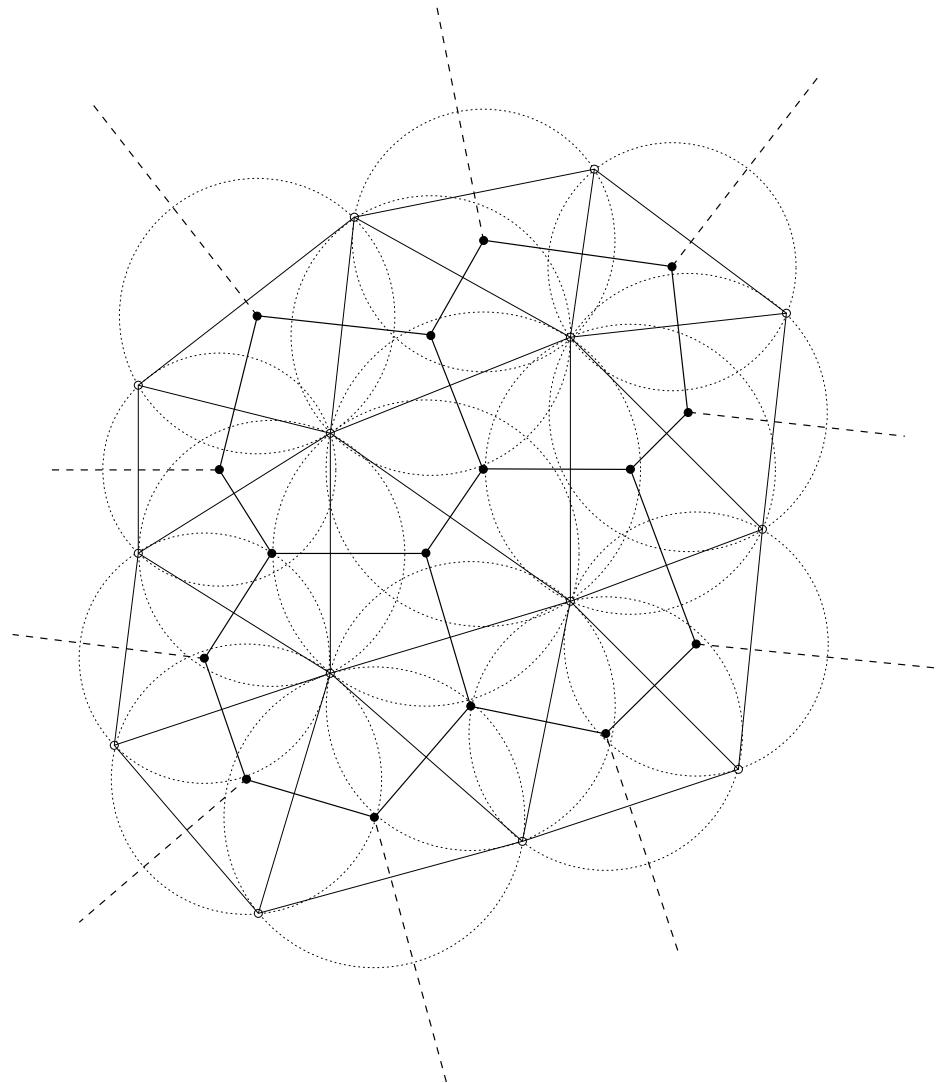


Figure 5.3: Delaunay tessellation (thin lines) of a 2D point set (empty circles), circumscribed circles (dotted lines), Voronoi vertices (filled circles), Voronoi diagram (continuous and dashed thick lines), embedded Voronoi diagram (continuous thick lines).

*inside* or *outside* is intrinsically defined. Therefore, in order to obtain medial axis approximation from the Voronoi diagram, a subset of the Voronoi diagram internal to the object must be selected, here referred as *embedded* Voronoi diagram. This operation is performed removing from the Delaunay tessellation the tetrahedra whose circumcenter falls outside the object, and then constructing only those Voronoi polygons whose vertex loops are complete (see Figure 5.3). From a practical point of view, to remove outer tetrahedra using surface information only, it is necessary to compute surface normals at surface mesh vertices, orientate them outward (this is done propagating the orientation from a outward-oriented seed neighborhood throughout the mesh), and remove a tetrahedron  $T_i$  if the following condition is not satisfied (see Figure 5.4)

$$(\mathbf{p}_j - \mathbf{c}_i) \cdot \mathbf{n}_j \geq 0 \quad \forall \mathbf{p}_j \in T_i \quad (5.5)$$

where  $\mathbf{c}_i$  is the circumcenter of  $T_i$  and  $\mathbf{n}_j$  are outward surface normals defined on tetrahedron vertices  $\mathbf{p}_j$ . The surface triangles of a generic mesh are not necessarily

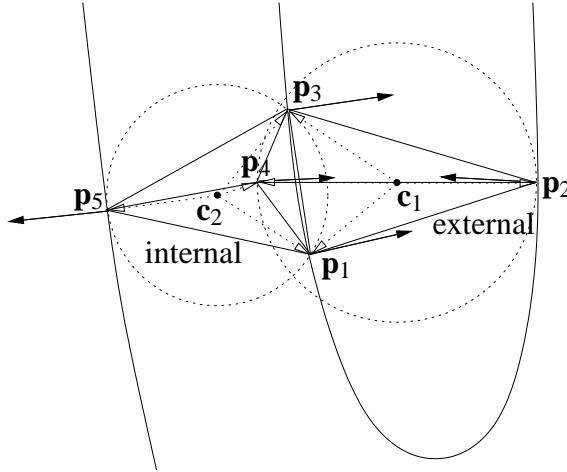


Figure 5.4: Selection of internal tetrahedra with the criterion in Equation 5.5 (see text for details). Surface normals are drawn as filled arrows, spheres circumscribed to Delaunay tetrahedra are dotted.

Delaunay tetrahedra faces, therefore the tetrahedron removal operation is not guaranteed to provide a set of tetrahedra whose boundary coincides with the original mesh. Experimentally, however, if the surface mesh doesn't present regions of low sampling density and its elements are not ill-shaped, the original surface mesh and Delaunay boundary surface differ for a small number of triangles, and the results obtained for Delaunay boundary surface are valid for the original surface mesh. However, this limitation can be overcome with the use of a Delaunay-conforming isosurface construction algorithm [11] in alternative to Marching Cubes, or with

the use of mesh filters which transform a generic surface mesh in a Delaunay-conforming one.

An example of what has been treated in this section is shown in Figure 5.5 for a model of abdominal aorta.

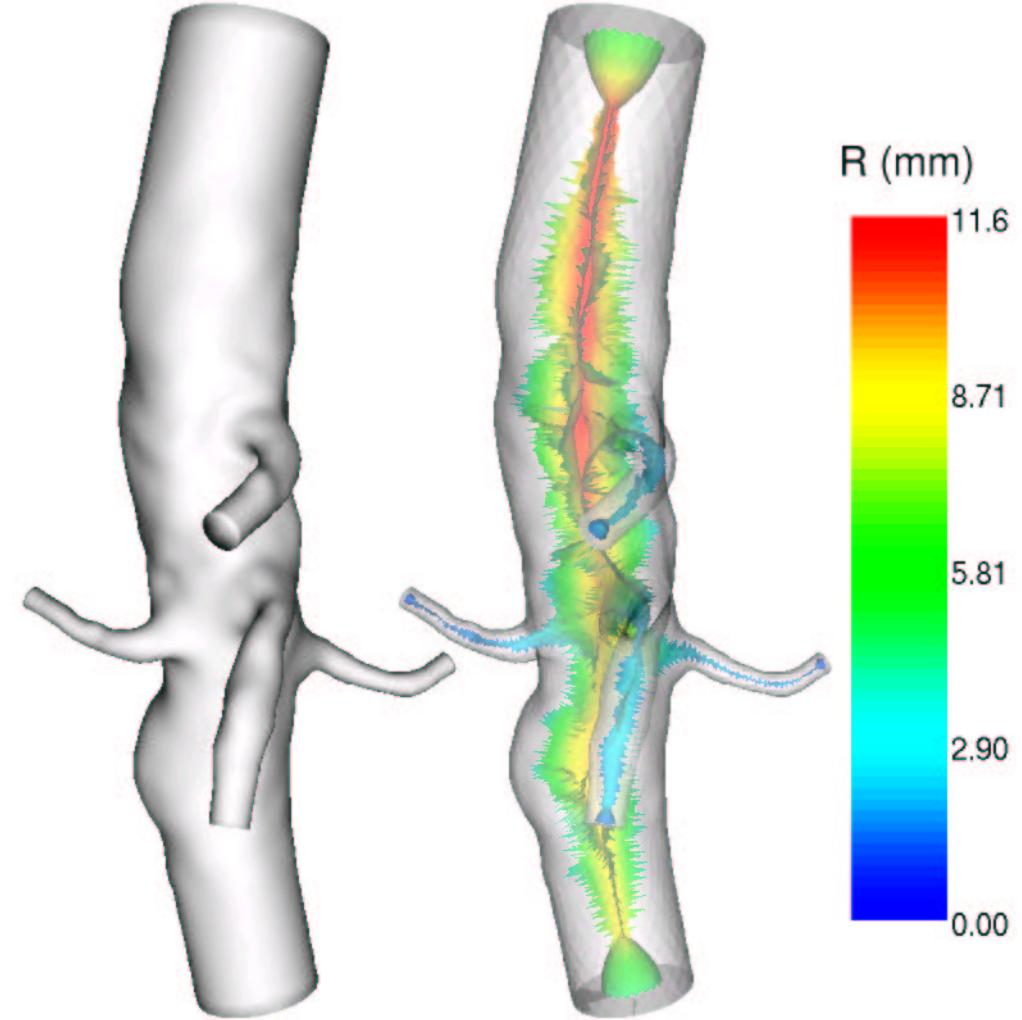


Figure 5.5: Left: abdominal aorta model surface. Right: embedded Voronoi diagram (colored surface) and internal Delaunay tetrahedra boundary (transparent surface). Colors map  $R$ , which is the radius of Voronoi spheres.

### 5.2.3 Poles and stable Voronoi diagram subsets

One problem arising when dealing with Voronoi diagrams, and with medial axes in general, is that they are very sensitive to boundary noise. Adding noise to the sur-

face of a regular shape results in the creation of medial axis branches extending in the direction of the surface. The reason of this behavior is that new branches in the vicinity of the surface host the centers of small maximal inscribed spheres fitting surface irregularities. Recently Belyaev et al. [15] presented interesting theoretical results on the effect of surface deformation on the medial axis.

Dealing with Voronoi diagrams, a slight change in the position of a single point can produce a strong effect on Voronoi diagram shape. Again, there is no simple rule to predict the effect of one point movement on Voronoi diagram geometry and topology, since new adjacency relationships among reconfigured Delaunay tetrahedra may take place, dual to a new organization of space partition described by the Voronoi diagram.

From the previous statements it follows that, when approximating the medial axis of an object using Voronoi diagram, even a change in surface point sampling density has an effect on Voronoi diagram structure. An important issue is therefore that of characterizing the behavior of the chosen medial axis approximation as point sampling density goes to infinity. Amenta et al. [2] have shown that in 3D the inner Voronoi diagram does not converge to the medial axis (although this is true in 2D). This is due to the fact that, as point density increases, neighboring points can be touched by small maximal spheres whose center is very near to the surface, circumscribed to very flat tetrahedra constructed among the neighboring points. It has been although demonstrated that a subset of the Voronoi vertices, the *poles*, converges to the medial axis. The poles of a point  $p$  are defined as the farthest inner and outer Voronoi vertices among the Voronoi vertices associated to  $p$  (i.e. among the circumcenters of the tetrahedra which have  $p$  as a vertex). Since for our means the inner poles only are of interest, each mesh vertex will be associated with an inner pole, indicated in the following as  $\text{pole}(p)$  (see Figure 5.6). It is worth to note that the direction of vector  $(\text{pole}(p) - \mathbf{p})$  approximates that of surface normal in  $p$ , because it is associated with the inscribed sphere which converges to the tangent sphere in  $p$ . We will make a wide use of poles later in this chapter. Since the set of poles is a disconnected subset of the Voronoi diagram, the use of disconnected poles to approximate the medial axis is limiting, because the information regarding medial axis topology is left unused. Several works have dealt with the problem of constructing approximation of both the geometry and the topology of the medial axis ensuring convergence to the continuous medial axis. Amenta et al. [3] set up an algorithm which allows to build a medial axis approximation, the *power shape*, whose vertices are the poles, with convergence guarantees. The power shape has the disadvantage of being thick (i.e. it is a volume rather than a non-manifold surface) and very sensible to noise. Bealyev et al. instead proposed to simply connect poles according to the connectivity of the associated surface points. Although this approach is convenient for some applications, it has the limitation of producing a two-sided self-intersecting manifold surface, which does not provide useful topological information. Based on the work of Amenta et al., Dey et al. [37] recently showed that a subset of Voronoi diagram converges to the medial axis, and presented a method to extract it from the Voronoi diagram based on local geometric

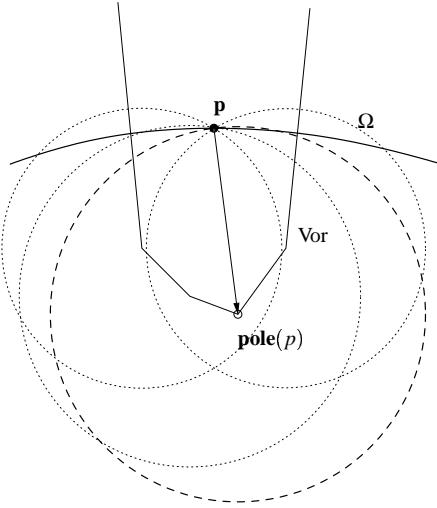


Figure 5.6: Schematic representation of the pole of a point  $\mathbf{p}$  sampling surface  $\Omega$ .  $\mathbf{pole}(p)$  is the farthest Voronoi vertex from  $\mathbf{p}$  belonging to the embedded Voronoi cell  $\text{Vor}(p)$ . The Voronoi sphere centered in  $\mathbf{pole}(p)$  is dashed, and approximates the maximal sphere inscribed in  $\Omega$  passing through  $\mathbf{p}$ . Voronoi spheres centered in the other vertices of  $\text{Vor}(p)$  are dotted.

criteria. In that work, the converging subset is defined as *stable*, i.e. not too sensitive to small perturbations in surface point distribution.

Independently from convergence properties of medial axis approximations, several works have been presented to extract stable subsets in order to deal with the sensitivity to noise which is inherent to medial axis itself. Usually the approach is that of pruning the unstable parts based on local criteria [83, 82] and user-defined thresholds. For example, Attali and Lachaud [11] proposed as local criterion a combination of two parameters defined on Voronoi vertices, thickness (i.e. the radius of the maximal sphere centered in one vertex) and bisector angle (i.e. the maximum angle made by two Voronoi edges incident in one vertex). This approach leads to good results, but suffers from dependence on scale and sampling point density, which leads to the need of defining proper thresholds depending on problem features. Conversely, the previously cited approach presented by Dey et al. [37] relies on scale and point density independent criteria, which make it possible to use the same thresholds for different geometries, and to provide theoretical guarantees. In this work we will not make use of Voronoi diagram simplification, instead we will deal with the whole embedded Voronoi diagram, as shown in the next section.

## 5.3 Centerline calculation

In the last section we presented a method to describe generic 3D objects. In particular it has been pointed out that it is necessary to identify stable medial axis subsets in order to derive usable information about the shape to be described. To do this, since our main goal is that of draw information about the geometry of blood vessels, we here use of the assumption of dealing with interconnecting  $2\frac{1}{2}$ D structures. The first information typically required when describing tubular structures is represented by centerlines<sup>3</sup>. A precise definition of centerline can be given as the line drawn from the two outermost sections of a tubular structure which locally maximizes the distance from vessel boundary. Referring to the medial axis stability problem, we can state that centerlines are the most stable medial entities for  $2\frac{1}{2}$ D structures.

Several methods have been presented for centerline extraction for medical imaging, with applications in angiography, virtual colonoscopy and bronchoscopy, among others. Often centerline computation is treated as a particular case of the calculation of the skeleton of a 3D object, which consists in both 1D and 2D elements. The approaches employed usually deal with 3D images rather than with piecewise linear surfaces, as in our case, and range from topological thinning, in which object voxels are removed from the segmented volume according to topological rules until the whole volume has been peeled and reduced to edgels (and surfels), to distance transform ridge finding, or combinations of the two approaches [8, 21, 44, 62, 92, 93, 101, 133]. Schirmacher et al. [102] proposed boundary surface shrinking based on distance transform gradient. Other approaches have been presented which directly deal with surface, as in our case. For example, the method presented in Chapter 3, although more suited to topological rather than geometric analysis of vessel structures, produces a good approximation of centerlines especially far from bifurcations, but without any theoretical guarantee. A surface-based approach was also proposed by Verroust et al. [126] relying on geodesic graphs, working also if only scatter points are available. Anyway, although the algorithm produces good results and is capable to automatically identify bifurcations, the method again lacks of theoretical guarantees.

Particular attention in this field has to be paid to the work of Deschamps et al. [36], who presented a method based on distance transform and wavefront propagation which produces centerlines provably satisfying their definition. Although that work deals with 3D digital images, it shares many concepts with our method.

### 5.3.1 Weighted geodesic formulation and the Eikonal equation

From the given definition of centerlines it follows that centerlines must lie on the ridges of the distance transform associated with the tubular surface, therefore on its medial axis. For this reason our approach is that of looking for centerlines on the

---

<sup>3</sup>Since the term *axis* refers to the notion of symmetry and better fits regular shapes, we will rather use the term *centerline* in case of generic tubular surfaces.

approximation of the medial axis of blood vessel surface given in terms of inner Voronoi diagram. To do that we formulate the centerline calculation problem in terms of energy minimization. We look for the path  $\mathbf{C} = \mathbf{C}(s)$  ( $s$  being arc length) traced from two points  $p_0$  and  $p_1$  which minimizes the following energy functional [36]

$$\mathcal{E}_{centerline}(\mathbf{C}) = \int_{0=\mathbf{C}^{-1}(p_0)}^{L=\mathbf{C}^{-1}(p_1)} F(\mathbf{C}(s))ds \quad (5.6)$$

where  $F(\mathbf{x})$  is a scalar field which is lower for more internal points, and  $\mathbf{C}$  is defined on the inner Voronoi diagram of  $\Omega$ . This formulation is very similar to that of snakes (see Chapter 2). Centerlines can in fact be viewed as deformable lines traced between two fixed points finding their stable configuration at a minimum of  $\mathcal{E}_{centerline}$ . Such minimization process is equivalent to solving the following evolutionary Euler-Lagrange equation (see Appendix A)

$$\frac{d\mathbf{C}}{dt} = -\nabla F(\mathbf{x}) \quad (5.7)$$

so that  $F(\mathbf{x})$  can also be seen as the scalar field whose gradient is the driving force attracting the snake to the center of the shape (for this reason  $F(\mathbf{x})$  is also called *centering potential*).

The centering potential we employ is a scalar field of the following simple form

$$F(\mathbf{x}) = \frac{1}{R(\mathbf{x})} \quad \forall \mathbf{x} \in \text{Vor}(P) \quad (5.8)$$

where  $P$  is the point set sampling the surface  $\partial\Omega$ , and  $R(\mathbf{x})$  is the scalar field defined on Voronoi diagram representing the values of the radius of the Voronoi spheres, which, we remind, are the maximal inscribed spheres with respect to the sampling point set  $P$ . Therefore  $R(\mathbf{x})$  is the value of the distance transform computed for the point set  $P$  evaluated over the Voronoi diagram. In defining  $F(\mathbf{x})$  as in Equation 5.8, it is necessary to ensure that  $R(\mathbf{x}) > 0$ , which is always true in absence of degeneracies of Voronoi diagram structure, avoided by careful numerical implementation. It is also worth to note that  $F(\mathbf{x}) \in C^0$  only (since  $DT(\mathbf{x}) \in C^0$ ), which means that particular treatment of  $\nabla F(\mathbf{x})$  should be provided on ridge points.

In order to gain more insight on the choice of such centering potential, let's consider the evolutionary form of the Euler-Lagrange equation associated to the minimization of energy functional  $\mathcal{E}_{centerline}$

$$\frac{d\mathbf{C}}{dt} = -\nabla \left( \frac{1}{DT(\mathbf{x})} \right) = \frac{\nabla DT(\mathbf{x})}{DT^2(\mathbf{x})} \quad \forall \mathbf{x} \in \Omega \setminus \partial\Omega \quad (5.9)$$

where this time the centering potential has been replaced with the distance transform of the boundary sampling points and the domain is a subset of  $\mathbb{R}^3$ . This evolution equation states that for  $\mathbf{C}$  to stop evolving, a ridge of  $DT(\mathbf{x})$  must be surely reached, otherwise  $\mathbf{C}$  will locally move toward the direction of increasing values of  $DT(\mathbf{x})$ . This ensures that the solution to the minimization problem lies

on the ridges of the distance transform, therefore on the Voronoi diagram. This confirms that the choice of our centering potential is consistent with restricting the search for the centerline on the inner Voronoi diagram.

Even if the snake approach is useful to understand the nature of the problem, we will not use snakes for the computation of centerlines. The reasons include the difficulty of dealing with the Voronoi diagram, a non-manifold surface, as the domain for line deformation, and the fact that a snake driven by the functional in Equation 5.6 is easily attracted toward local minima, making the results depend on initialization. We instead treat this problem as a weighted geodesic computation problem. In fact, minimization  $\mathcal{E}_{\text{centerline}}$  in Equation 5.6 can be also achieved by computing the shortest path on the Voronoi diagram domain between  $p_0$  and  $p_1$  according to the metric induced by  $F(\mathbf{x})$  [30] (setting  $F(\mathbf{x}) = 1$  yields classic geodesics computed on surface metric). With this approach it is possible to directly extract the global minimum of  $\mathcal{E}_{\text{centerline}}$ .

In the weighted geodesic approach, once a scalar field is defined on the domain, the aim is to find the path from  $p_0$  to  $p_1$  for which the integral of the scalar field along the path is minimal (for this reason the path is also called *minimal cost* or *minimal action* path). One way to solve this problem is that of defining the weighted geodesic distance of each point of the domain to point  $p_0$ , and then to trace the path back from  $p_1$  to  $p_0$  along the gradient of weighted geodesic distance field [30, 36]. We define weighted geodesic distance on the whole domain as

$$T(\mathbf{x}) = \min_{\mathbf{C} \in \Gamma(\mathbf{p}_0, \mathbf{x})} \left( \int_{0=\mathbf{C}^{-1}(\mathbf{p}_0)}^{\bar{s}=\mathbf{C}^{-1}(\mathbf{x})} F(\mathbf{C}(s)) ds \right) \quad (5.10)$$

where  $\Gamma(\mathbf{p}_0, \mathbf{x})$  is the set of all paths from  $p_0$  to  $\mathbf{x}$  which can be constructed on the domain (i.e. the Voronoi diagram). Since the scalar field  $T(\mathbf{x})$  represents weighted geodesic distances to point  $p_0$ , the level sets of  $T(\mathbf{x})$  are the regions of equal weighted geodesic distance to  $p_0$ , and weighted geodesic paths are orthogonal to level sets of  $T(\mathbf{x})$  (therefore, they are always tangent to  $\nabla T(\mathbf{x})$ ). This means that the distance needed to move along the geodesic in such a way to make  $T(\mathbf{x})$  vary of  $dT$  is  $\frac{dT}{F}$ . In other words, the value of the directional derivative in the direction of geodesics is  $dT \left( \frac{dT}{F} \right)^{-1}$ , so that  $T(\mathbf{x})$  must satisfy the following equation over the whole domain

$$|\nabla T(\mathbf{x})| = F(\mathbf{x}) \quad (5.11)$$

which is a nonlinear hyperbolic partial differential equation which takes the name of Eikonal equation (with  $F(\mathbf{x}) > 0$ ). As shown at the end of this section, obtaining  $T(\mathbf{x})$  by solving Equation 5.11 is much more convenient than using the variational formulation in Equation 5.10, especially in our case, where the domain is a generic non-manifold surface.

As previously introduced, once the field of weighted geodesic distances has been found, the weighted geodesic path of interest is calculated constructing a path from  $p_1$  oriented as  $\nabla T(\mathbf{x})$  in every point. This means solving the following ordinary differential equation

$$\frac{d\tilde{\mathbf{C}}(\tilde{s})}{d\tilde{s}} = -\nabla T(\mathbf{x}) \quad (5.12)$$

and then obtaining  $\mathbf{C}(s)$  which minimizes the energy  $\mathcal{E}_{centerline}$  in Equation 5.6 by arc length parameterization of  $\tilde{\mathbf{C}}(-\tilde{s})$  with  $\mathbf{C}(0) = \mathbf{p}_0$  and  $\mathbf{C}(L) = \mathbf{p}_1$ .

With this approach, centerlines satisfying a criterion of minimal energy and lying on medial axis are computed. As previously reported, Deschamps et al. [36] proposed a method based on the solution of the Eikonal equation over a 3D image with  $F(\mathbf{x})$  depending on an approximation of  $DT(\mathbf{x})$ , even if in that work no reference was made to the concept of medial axis. Conversely to theirs, our method is suited for piecewise linear surfaces, and it is the basis for precise local surface characterization, as shown in the next sections. Moreover, the theoretical approach of describing a surface through the relation with its stable medial axis subsets can be extended to general shapes, as shown at the end of this chapter.

### 5.3.2 Numerical solution

In the previous sections we showed how an approximation of the medial axis for a piecewise linear surface can be obtained by a subset of the Voronoi diagram of its mesh vertices. We then showed that Voronoi diagram is the domain where we look for centerlines following a energy minimization approach. The procedure described for the computation of centerlines requires to compute the solution of a partial differential equation (Equation 5.11) and an ordinary differential equation (Equation 5.12) on the Voronoi diagram, which we remind is a nonmanifold surface made up of convex polygons. We will now present two methods to efficiently approximate those two equations on the Voronoi diagram, finally yielding an efficient and robust tool for the computation of centerlines and for subsequent local surface characterization.

#### The fast marching method on polygonal non-manifolds

Numerical approximation of nonlinear hyperbolic Equation 5.11 must take into account the fact that its solution  $T(\mathbf{x})$  can be non-smooth. Consider at this extent the 1D version of Equation 5.11,

$$\left| \frac{dT}{dx} \right| = 1 \quad (5.13)$$

where  $F(x) = 1$  for simplicity, with  $x \in [0, 1]$  and boundary conditions  $T(0) = 0$  and  $T(1) = 0$ . The analytical solution of Equation 5.13 for  $T(x) \geq 0$  is expressed

by

$$\begin{cases} T(x) = x & x \in [0, 0.5] \\ T(x) = 1 - x & x \in (0.5, 1] \end{cases} \quad (5.14)$$

Basically this is the distance function from points  $x = 0$  and  $x = 1$  over the interval  $[0, 1]$ . This solution presents a cusp for  $x = 0.5$ , where  $T(x)$  is non-differentiable. A similar problem was encountered in Chapter 2 in the context of level sets. There the focus was on Hamilton-Jacobi equations, which are time-dependent nonlinear hyperbolic equations which are tightly linked to the Eikonal equation, the latter being a static version of Hamilton-Jacobi equation<sup>4</sup>. As in that case, the hyperbolic nature of Equation 5.11 leads to the need of a particular treatment of odd-order derivative terms, such as for the approximation of gradient modulus.

Since  $T(\mathbf{x})$  can present non-differentiable ridges, central finite differences produce instabilities in the solution no matter how dense is the discretization, because they average information on both sides of the ridges. To better understand this concept, we again refer to the 1D case. Central finite difference expression for the first derivative of  $T(x_i)$  in 1D is given by

$$\frac{dT}{dx} \Big|_{x_i} \approx \frac{T_{i+1} - T_{i-1}}{2h} \quad (5.15)$$

where  $h$  is grid spacing. For grid points located at a distance lower than  $h$  from  $x = 0.5$ , the above expression gives incorrect results. For example, calculating the left-hand value of Equation 5.13 for  $x_i = 0.5$  using central finite differences, taking values from the analytical solution in Equation 5.14, results  $\frac{dT}{dx} \Big|_{x_i} \approx 0.5$ , which does not satisfy Equation 5.13.

Conversely, employing upwind finite differences, contributions are correctly evaluated separately on both sides of the ridges, and the solution can be accurately modeled in regions of non-differentiability. Always referring to the 1D case, upwind finite difference approximation of Equation 5.13 is given by

$$\max\left(-\frac{T_{i+1} - T_i}{h}, \frac{T_i - T_{i-1}}{h}, 0\right) = 1 \quad (5.16)$$

In this expression, first derivatives are computed from one-sided contributions. Left-hand side of Equation 5.16 was constructed taking into account the fact that the solution must grow moving away from boundary points. Therefore, the contributions only come from points with smaller values of  $T$ . This time, the left-hand side of Equation 5.16 for  $x_i = 0.5$  correctly yields  $\frac{dT}{dx} \Big|_{x_i} \approx 1$ , satisfying the Eikonal equation. The upwind finite difference approximation of the Eikonal equation over

<sup>4</sup>The general Hamilton-Jacobi equation has the form  $\frac{\partial T(\mathbf{x}, t)}{\partial t} + \mathcal{H}(\mathbf{x}, \nabla T(\mathbf{x}, t)) = 0$ . By dropping time derivative and choosing  $\mathcal{H}(\mathbf{x}, \nabla T(\mathbf{x}, t)) = |\nabla T(\mathbf{x})| - F(\mathbf{x})$  we obtain the Eikonal equation.

a 3D Cartesian grid of spacing  $h$  is

$$\left[ \sum_{l=1}^3 \max(D_{ijk}^{-x_l} T, -D_{ijk}^{+x_l} T, 0)^2 \right]^{\frac{1}{2}} = F_{ijk} \quad (5.17)$$

where  $D_{ijk}^{-x_1} T = \frac{T_{i,j,k} - T_{i-1,j,k}}{h}$  and  $D_{ijk}^{+x_1} T = \frac{T_{i+1,j,k} - T_{i,j,k}}{h}$  (expressions for  $x_2$  and  $x_3$  are straightforward) are the upwind approximations of first-order derivatives.

Based on the approximation given in Equation 5.17, an efficient computational technique, the fast marching method, has been proposed by Sethian [106] to solve the Eikonal equation. This method takes advantage from the aforementioned observation that information in Equation 5.17 always propagates from smaller to larger values of  $T$ , so that the solution can be constructed proceeding downwind from the boundary, here referred to as seed points. Since no contribution can flow from points with larger to points with smaller values of  $T$ , the point with the lowest value of  $T$  among the update set can be *frozen*, which means that it will not be considered for further updates. Solution update is therefore performed only for a narrow band of points around the frozen points. Based on this strategy, grid points can be classified into three groups, *accepted* points (initially the seed points only), *considered* points (the narrow band of points adjacent to the accepted ones) and *far* points (points beyond the band of considered points) (see Figure 5.7). The algorithm is

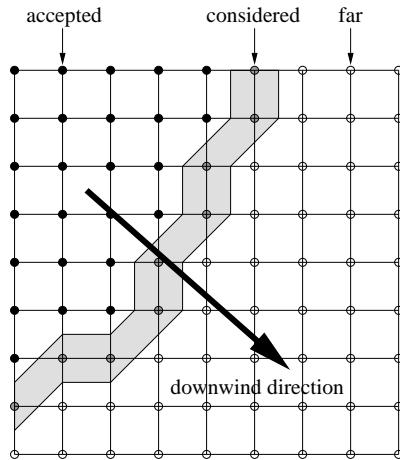


Figure 5.7: Representation of the classification of grid points in the fast marching method.

summarized as following

```

Let TRIAL be the CONSIDERED point with the lowest value of T;
Tag as CONSIDERED all the neighbors of TRIAL that are not ACCEPTED;
Tag TRIAL as ACCEPTED;
Recompute the value of T for the CONSIDERED neighbors of TRIAL

```

by solving the quadratic update equation.

The above procedure is repeated until all the points have been tagged as *accepted*. The most expensive operation in the algorithm is the selection of the *considered* point having the smallest  $T$ , which in general requires a sweep through all *considered* points before each freezing operation, resulting in overall  $O(N^2)$  complexity,  $N$  being grid point number. However, computational complexity is lowered to  $O(N \log N)$  employing a min-heap data structure for *considered* points management.

Min-heap data structure is a complete binary tree for which the value at any node is less than or equal to the values at its children. Therefore, the root of the mean heap is the smallest valued element, which can be accessed in constant time. The insertion of one node in the min-heap requires a sweep from the root (i.e. the smallest valued element) to the lower level, in the worst case, which results in a computational complexity of  $O(\log N)$ . The same cost is required for the update of the  $T$  value of a *considered* point, and for the removal of the root, when the point with the lowest value of  $T$  is freezed (i.e. removed from the set of *considered* points). The binary tree can be efficiently represented by an array in which the children of a node located at the array index  $k$  are stored in  $2k$  and  $2k + 1$ , and consequently the parent of  $k$  is stored in  $k/2$ .

So far we dealt with the fast marching method for orthogonal structured grids. Let's now move on to the case of unstructured manifold domains, developed by Sethian's group in [57, 107]. Although the four steps of the fast marching method remain substantially unchanged, the update step, involving the quadratic Equation 5.17, must be adapted to unstructured domains and particular care must be taken to ensure that information always flows downwind. For a manifold piecewise linear surface, the finite difference approximation of gradient  $\nabla T(x_i)$  at mesh vertex  $x_i$  given  $T(x_i)$  and  $T(x_j)$ , where  $x_j$ ,  $j = 1..n$ , are the  $n$  neighbor vertices of  $x_i$ , can be expressed by a linear combination of the  $n$  directional derivatives  $v_j(x_i)$  with respect to the directions  $\mathbf{e}_j = \frac{\mathbf{x}_i - \mathbf{x}_j}{|\mathbf{x}_i - \mathbf{x}_j|}$ , so that

$$\mathbf{E} \nabla T(x_i) = \mathbf{v}(x_i) \quad (5.18)$$

where  $\mathbf{E}$  is the matrix whose rows are directions  $\mathbf{e}_j$ , and  $\mathbf{v}(x_i)$  is the column vector of directional derivative values. Solving for  $\nabla T(x_i)$  and substituting in the Eikonal equation (Equation 5.11) yields

$$\mathbf{v}^T(x_i)(\mathbf{E}\mathbf{E}^T)^{-1}\mathbf{v}(x_i) = F^2(x_i) \quad (5.19)$$

which expresses the Eikonal equation in terms of the directional derivatives with respect to vertex-to-neighbor directions (for triangular element meshes vertex-to-neighbor directions coincide with mesh edge directions). Discretization is finally obtained expressing directional derivatives  $v_j(x_i)$  with a finite-difference approach. In our case we will employ the first-order approximation  $v_j(x_i) \approx a_j T(x_i) + b_j$ , with

$a_j = \frac{1}{|\mathbf{x}_i - \mathbf{x}_j|}$ , and  $b_j = -\frac{T(x_j)}{|\mathbf{x}_i - \mathbf{x}_j|}$ , resulting in  $v_j(x_i) \approx \frac{T(x_i) - T(x_j)}{|\mathbf{x}_i - \mathbf{x}_j|}$ . The general expression for the Eikonal equation discretized over an unstructured manifold grid with first-order finite differences is therefore the quadratic equation

$$(\mathbf{a}^T \mathbf{Q} \mathbf{a}) T^2(x) + (2 \mathbf{a}^T \mathbf{Q} \mathbf{b}) T(x) + (\mathbf{b}^T \mathbf{Q} \mathbf{b}) = F^2 \quad (5.20)$$

where  $\mathbf{Q} = (\mathbf{E} \mathbf{E}^T)^{-1}$ . Since the gradient term has been expressed, in Equation 5.18, as the linear combination of the directional derivatives of all its neighbors, the latter equation lacks an upwinding criterion, necessary for correct approximation of the Eikonal equation. Such criterion can be translated from orthogonal to unstructured grids by decomposing the update of a point  $x_i$  into the contributions computed from each neighboring triangle  $x_j x_i x_{j+1}$ , and accepting only those contributions for which the computed gradient  $\nabla T(x_i)$  comes from inside triangle  $x_j x_i x_{j+1}$ . This condition is expressed by the two inequalities

$$\begin{aligned} a_j T(x_i) + b_j &\geq (\mathbf{e}_j \cdot \mathbf{e}_{j+1}) (a_{j+1} T(x_i) + b_{j+1}) \\ a_{j+1} T(x_i) + b_{j+1} &\geq (\mathbf{e}_{j+1} \cdot \mathbf{e}_j) (a_j T(x_i) + b_j) \end{aligned} \quad (5.21)$$

A triangle satisfying the above inequalities is called *defining* triangle for point  $x_i$  (see Figure 5.8). The new value of  $T(x_i)$  is finally chosen to be the lowest among

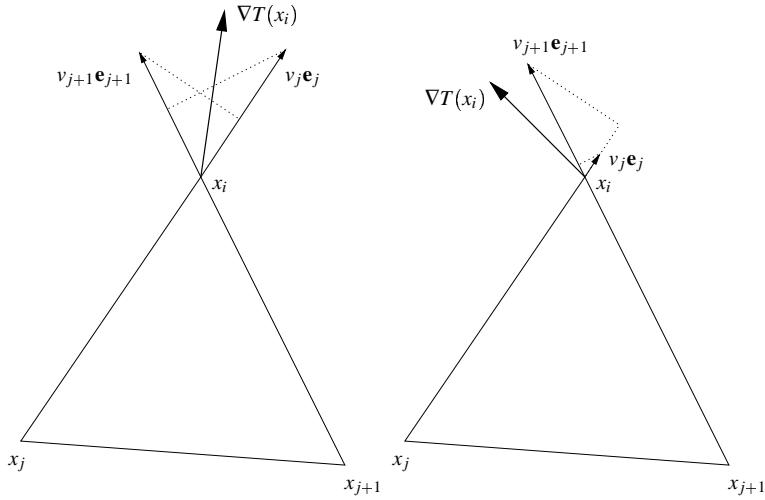


Figure 5.8: Left: triangle satisfying the upwinding criterion in Equation 5.21 for vertex  $x_i$ . Right: triangle not satisfying the first of the two conditions in Equation 5.21, for which  $\nabla T(x_i)$  does not come from within the triangle.

the values computed from all the defining triangles for  $x_i$ , similarly to the orthogonal grid case in Equation 5.17, where the maximum values of directional derivatives among the two orientations of each axis were chosen.

Let's now finally extend the fast marching method to the non-manifold surface of the Voronoi diagram, as presented by us in [6, 5]. We remind that on a non-manifold surface it is not possible in general to map a neighborhood of a point onto a disk. This means that Equation 5.18 cannot be employed to compute an approximation of  $\nabla T(x_i)$  over the whole neighborhood. In the Voronoi diagram, which is made up of convex polygons, more than two polygons can share a common edge. Therefore the neighborhood of a Voronoi vertex is in general made up of points belonging to a number of incident polygons. Since the update performed in the fast marching method exclusively proceeds downwind and it is based on upwind derivatives, we can simply compute the update from each incident polygon separately and finally choose the minimum value of  $T(x_i)$  obtained, with the same criterion used to implement the upwind criterion for unstructured manifold domains. Moreover, the polygons are convex, which in practice means that every point can be connected to any other by a straight line not intersecting the boundary of the polygon. Our strategy is therefore to use Equation 5.18 to approximate  $\nabla T(x_i)$  over each polygon of the neighborhood. The upwinding criterion is then accounted in the following way. Since the polygons are convex, it is possible to build a triangle with the update point and any other two points of each polygon. Therefore, each time a point has to be updated, all triangles with the *accepted* points of every incident polygon are constructed, and the update which yields the minimum value of  $T$  computed for the triangles which satisfy Equation 5.21 is chosen. The

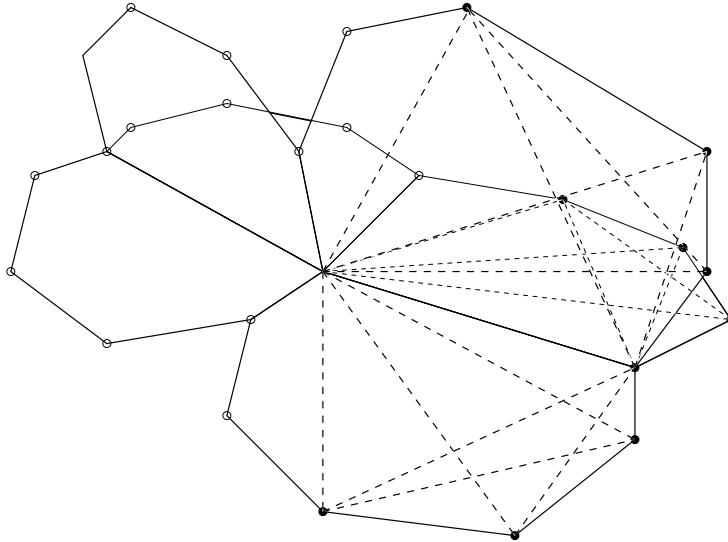


Figure 5.9: Update of a point from its non-manifold neighborhood. The filled circles represent *accepted* points. The triangles from which the update is performed are constructed with the dotted lines, traced between *accepted* points and the updating point.

resulting algorithm is reported in the following

```

Let TRIAL be the CONSIDERED point with the lowest value of T;
Tag as CONSIDERED all the points of the polygons
    sharing TRIAL that are not ACCEPTED;
Tag TRIAL as ACCEPTED;
For each CONSIDERED point (name it UPDATING) belonging
    to a polygon incident to TRIAL:
    For each polygon incident to UPDATING:
        For each triangle constructed with UPDATING and
            any other two ACCEPTED points of the current polygon:
        {
            Compute the value of T for UPDATING by solving the
                quadratic update equation;
            If the upwinding condition is satisfied and T is lower
                than the current value of UPDATING:
                Set computed T as the current value of UPDATING;
        }
    }
```

## Backtracing

As previously introduced, once the Eikonal equation is solved over the whole Voronoi diagram with the pole associated to the barycenter of an inlet section as the seed point, centerlines are obtained by backtracing a path along the direction of maximum descent of  $T(\mathbf{x})$  (Equation 5.12) starting from each of the poles associated to the barycenter of the remaining sections. High-order schemes are employed for the solution of the ordinary differential Equation 5.12 on manifolds, in order to minimize numerical errors which accumulate at each step during the integration. However, the extension of such schemes to non-manifold domains is not trivial, because the gradient cannot be expressed as a linear combination of directional derivatives computed by evaluating the integrand in the neighborhood. We instead adopted a simple first-order scheme tailored to non-manifold surfaces made up of convex polygons. One advantage of our method, besides being fast, is that centerline vertices are defined on Voronoi polygon boundaries only.

Given a point  $\tilde{\mathbf{x}} = \mathbf{C}(\tilde{s})$  located on the boundary of a Voronoi polygon, consider Voronoi polygons sharing the edge on which  $\tilde{\mathbf{x}}$  is defined. The point on polygon edges which yields the maximum descent of  $T(\mathbf{x})$  from  $\tilde{\mathbf{x}}$  is set to be the new centerline point. Since the polygon is convex, the centerline is ensured to lay on the Voronoi diagram. If the distance of the maximum descent point to a Voronoi vertex is lower than a given threshold, the new centerline point is set to coincide with the Voronoi vertex. This last adjustment is important, because if a point is near to a vertex but still results to lie on an incident edge, only the polygons sharing the edge are taken into account as neighbor polygons, while all the polygons sharing the vertex should be accounted as neighbors.

### 5.3.3 Centerlines and vessel radius measurements

The resulting centerline is a piecewise linear line defined on the Voronoi diagram, whose vertices lie on Voronoi polygon boundaries [6, 5]. Values of Voronoi sphere radius  $R(\mathbf{x})$  are therefore defined on centerlines, so that centerline points are associated with maximal inscribed spheres, as shown in Figure 5.10. Since centerlines were constructed to lie on local maxima of distance from the boundary, there is a tight connection between maximal sphere radius and minimum projection diameter used in clinical evaluation [136]. In fact, classic angiographic vessel diameter evaluation is performed considering the minimum diameter obtained by measurements on different projections. If the vessel is convex in the neighborhood of the section of interest, our method provides the measure of the minimum vessel diameter with respect to the projection angle, as shown in Figure 5.11, plotted against centerline abscissa in Figure 5.12. The availability of a robust method for centerline computation and diameter measurement as the one presented here allows to characterize blood vessel geometry in a synthetic way, therefore giving the opportunity of performing a study on a population of models. Since it has been shown that planarity, tortuosity and branching angles have a major influence on complex blood flow patterns, such a study may reveal if particular vessel configurations are involved in vascular pathology.

As previously anticipated, besides providing useful synthetic characterization tools, the method developed so far is the basis for more detailed surface characterization, as shown in the following.

## 5.4 Relating surface points to centerlines

Local surface characterization for  $2\frac{1}{2}$ D geometries can be performed by relating surface points to the centerlines. As an immediate application, this allows to compute a surface map of the distance of each surface point to the centerlines. Such result is useful, for example, for analyzing and comparing surface morphology, and documenting changes in vessel wall geometry over time under pathological conditions.

A common approach to compute the distance to centerlines is to employ minimum Euclidean distance. This approach works well only in the case of a nearly cylindrical shape with no tight bendings, bifurcations or abrupt changes in vessel diameter. In the cases this is not satisfied, such as in Figure 5.13, measures can be affected by severe inaccuracies and, in presence of bifurcations, minimum distance can lead to the assignment of surface points to the wrong branch. This simple approach can be improved by taking into account surface normal orientation. In fact, if a vessel has a nearly circular section shape, surface normals point toward the centerline, so that the distance from centerlines can be obtained balancing the minimization of surface point-centerline distance and the maximization of the inner product of the distance vector with the inward surface normal. Also this approach is not robust when applied to vessels having an irregular section shape

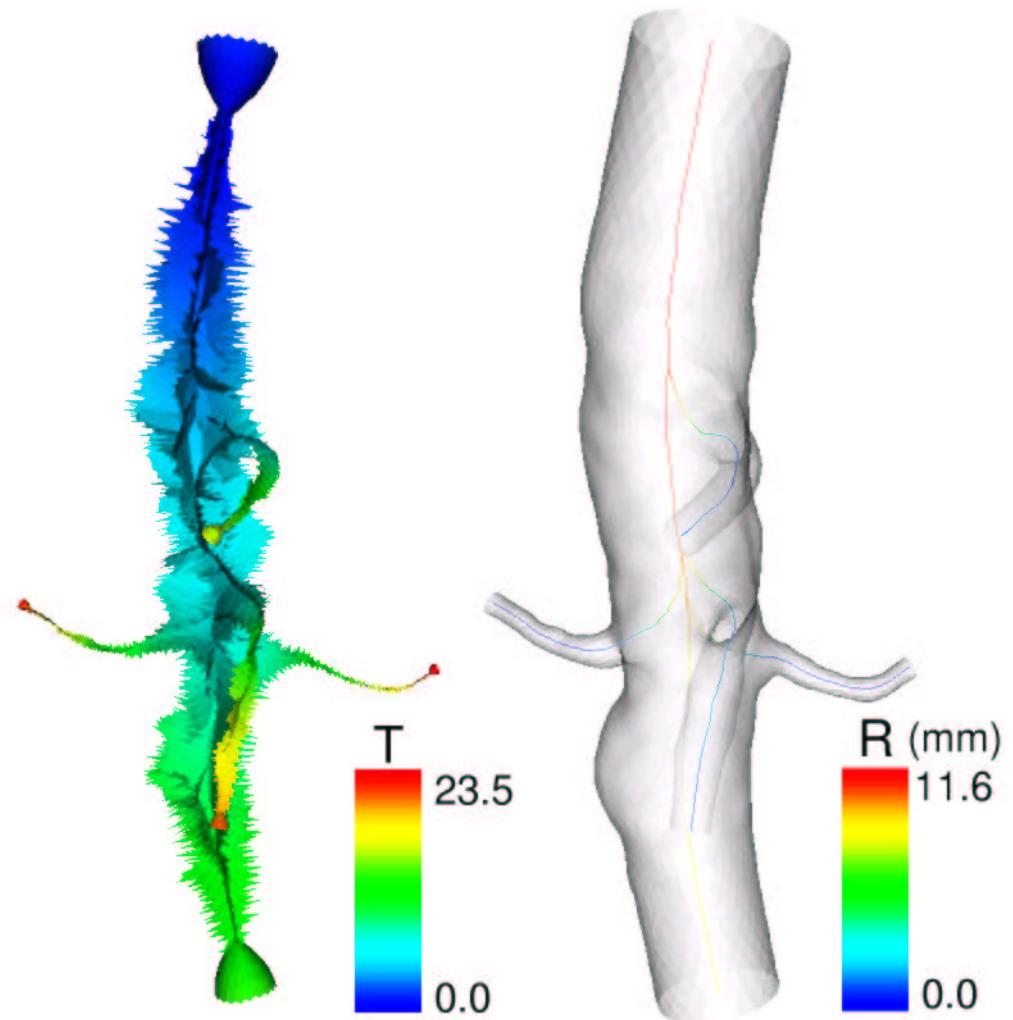


Figure 5.10: Left: solution of the Eikonal equation over the embedded Voronoi diagram for the abdominal aorta model presented in Figure 5.5, with seed point at the aorta inlet. Right: centerlines obtained after backtracing from model outlets (namely aorta outlet, renal, superior mesenteric and celiac arteries). Colors on centerlines are the values of maximal inscribed sphere radius.

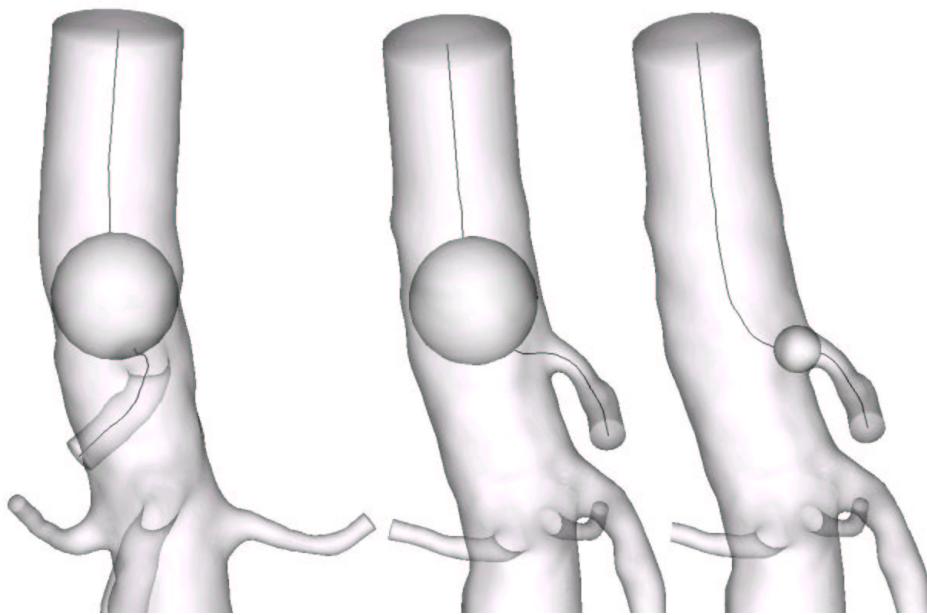


Figure 5.11: Maximal inscribed spheres along a centerline of the abdominal aorta model. The first two images depict the same sphere from two different viewpoints, evidencing the tight connection with the minimum projection diameter measurement method employed in the clinical practice.

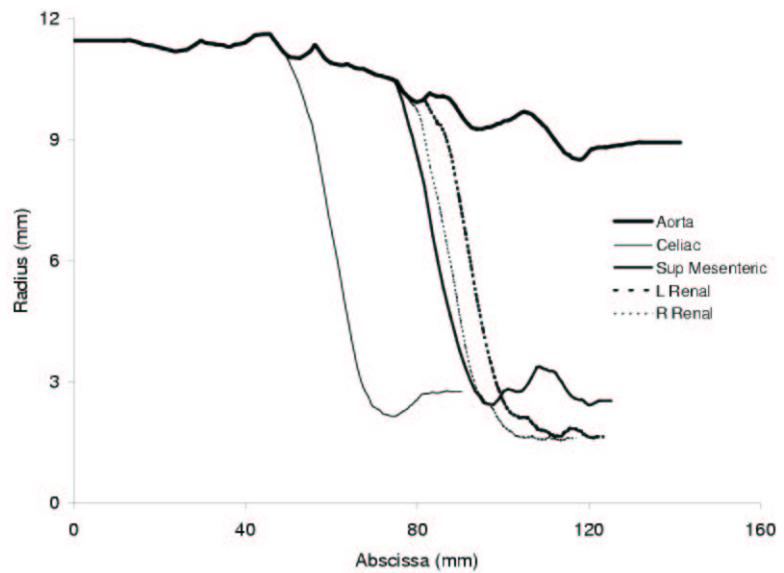


Figure 5.12: Plot of maximal inscribed sphere radius versus centerline abscissa for the abdominal aorta model.

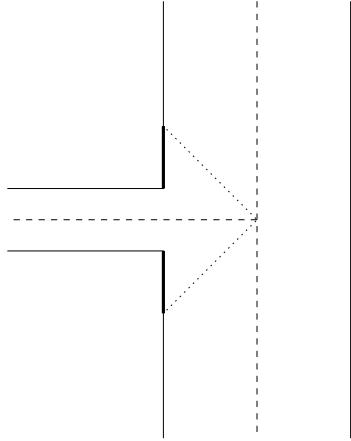


Figure 5.13: Schematic example of an error caused by employing minimum Euclidean distance to compute the distance of surface points to centerlines. A portion of large vessel boundary (indicated by thick lines) is incorrectly related to small vessel centerline.

and in bifurcation regions.

The main reason of failure of previous methods is that centerlines are not fully representative of a generic interconnecting  $2\frac{1}{2}$ D geometry. Recalling what has been presented at the beginning of the previous section, centerlines were interpreted as the most *stable* subsets of the medial axis for interconnecting  $2\frac{1}{2}$ D geometries. Moving from surface points in the direction of inward normals means moving toward the medial axis, because maximal inscribed spheres are tangent to the surface. In previous sections, dealing with the Voronoi diagram as the approximation of the medial axis, *poles* have been introduced, defined as the farthest inner Voronoi vertex associated to a surface point. It has also been noted that the direction of vectors  $\text{pole}(p) - \mathbf{p}$  approximates inward normal. This property derives from the fact that poles are the subset of Voronoi vertices which converge to the medial axis when surface sampling point density tends to infinity, therefore they converge to the center of maximal balls inscribed in the continuous surface. Thanks to these important properties, the distance  $|\text{pole}(p) - \mathbf{p}|$  is also referred to as to *thickness*.

Based on these observations, the problem of relating surface points to centerlines was solved by first finding the pole associated to each surface point, and then the centerline point which is closest to the pole. This latter step is robustly performed by measuring distances on Voronoi diagram surface, therefore taking advantage of the topological information inherent in the Voronoi diagram structure.

### 5.4.1 Geodesic distance to centerlines

Identifying the pole associated to a surface point is trivial once the inner Voronoi diagram has been constructed. Therefore, we will now focus on the relation between poles and centerlines.

The computation of the minimum distance of poles to centerlines measured on the Voronoi diagram is a geodesic distance computation problem, which is formalized with an integral formulation analogous to Equation 5.10, with  $F(\mathbf{x}) = 1$ . As shown in the previous section, geodesic distances are obtained by solving the following Eikonal equation, where  $F(\mathbf{x}) = 1$

$$|\nabla \tilde{T}(\mathbf{x})| = 1 \quad \forall \mathbf{x} \in \text{Vor}_E(P) \quad (5.22)$$

where  $\tilde{T}(\mathbf{x})$  is geodesic distance field (the tilde has been introduced to avoid confusion with the previous Eikonal equation solution),  $\text{Vor}_E(P)$  is the embedded Voronoi diagram, and  $P$  is the set of surface points. The boundary conditions are here expressed by zero values on centerline points. The solution to this problem is found once again applying the fast marching method for polygonal non-manifolds (see Figure 5.14 left).

The measure of geodesic distances from surface points to centerlines is then expressed by [6, 5]

$$D_g(p) = |\mathbf{pole}(p) - \mathbf{p}| + \tilde{T}(\mathbf{pole}(p)) \quad (5.23)$$

as shown in Figure 5.14 right. Since a pole is the farthest Voronoi vertex in the inward direction,  $|\mathbf{pole}(p) - \mathbf{p}|$  is the length of the line which connects a surface point with the Voronoi vertex having the lowest value of  $\tilde{T}(\mathbf{x})$  in its Voronoi cell. In other words,  $(\mathbf{pole}(p) - \mathbf{p})$  is the most direct path from  $\mathbf{p}$  to the central paths in the metric expressed by  $\tilde{T}(\mathbf{x})$ . Although this quantity overestimates the Euclidean distance of surface points to the centerlines, except when the poles lie on centerlines (such as in cylinders, for which the second term of Equation 5.23 is zero, it turns out to be a robust measure especially when the shape of vessel sections is irregular.

### 5.4.2 Euclidean distance to centerlines

Geodesic distance computation, besides overestimating Euclidean distance, doesn't directly provide a correspondence between surface points and centerlines. In other words, it is still not defined which centerline point is (geodesically) nearest to each surface point. If such a relationship is found, calculating the Euclidean distance  $D_e$  from surface points to centerlines is straightforward.

The correspondence can be retrieved constructing geodesic paths on the Voronoi diagram from surface points (more correctly, from the associated poles) to the centerlines. Analogously to what presented in the previous section, this is done tracing a path from each pole following the steepest descent of the geodesic distance field

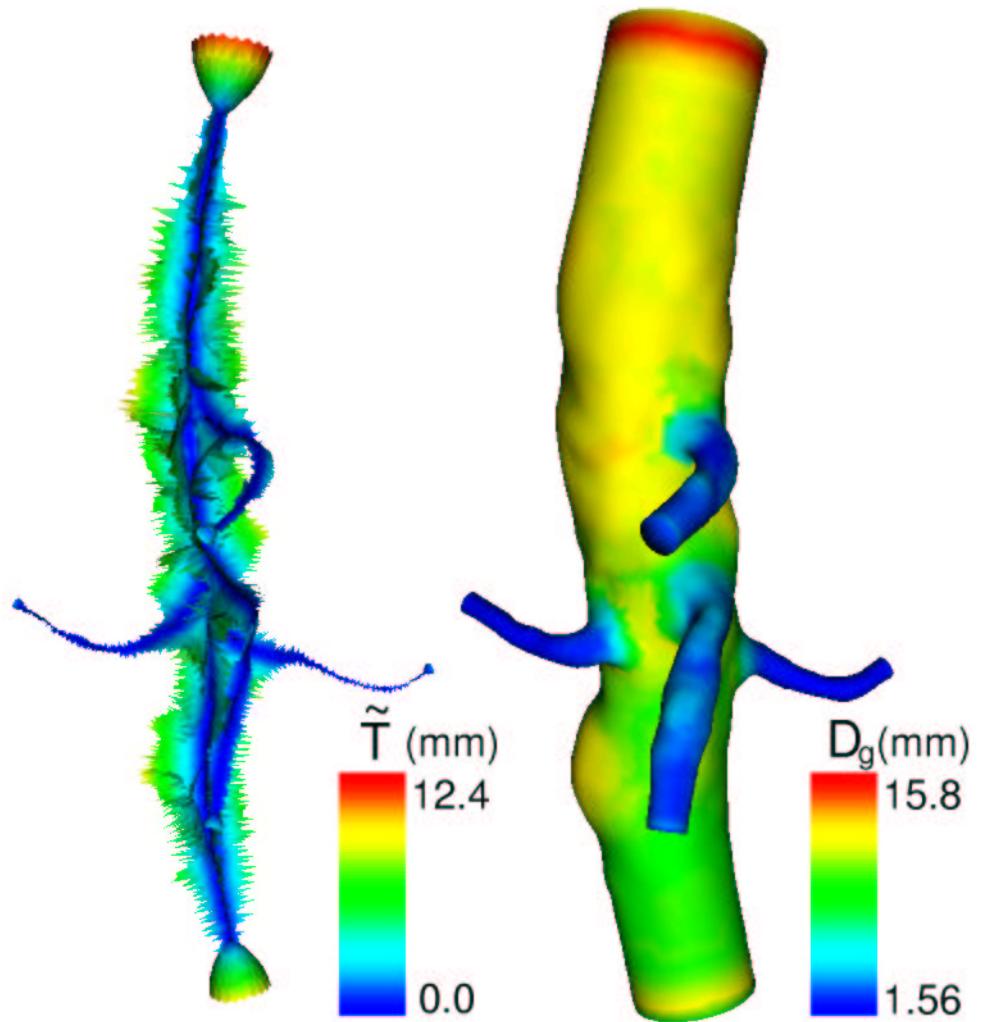


Figure 5.14: Left: solution of the Eikonal equation over the embedded Voronoi diagram with zero boundary condition on the centerlines. Right: geodesic distance of surface points to centerlines.

$\tilde{T}(\mathbf{x})$ , therefore solving Equation 5.12, until centerlines are encountered. The endpoint  $\mathbf{c}(p)$  of the path starting in  $\mathbf{pole}(p)$  is actually the geodesically nearest central path point to the selected pole, therefore it is also the geodesically nearest central path point to the surface point(s) associated with that pole. Thus, we can write

$$D_e(p) = |\mathbf{c}(p) - \mathbf{p}| \quad (5.24)$$

Figure 5.15 shows a result of Euclidean distance computation and a comparison between geodesic and Euclidean distance. Disclosing the relationship between surface points and centerlines constitutes an important result in vessel description, because it allows to characterize surface points according to their belonging to different vessel branches, or to the position of the associated points along the centerlines.

## 5.5 Normalized tangency deviation

In the previous section the problem of relating surface points to centerlines was decomposed into relating surface points to poles, and poles to centerlines. In other words, a path was created moving from surface points along surface normals to the medial axis, and from there to centerlines moving on the medial axis. This procedure suggests to compare the two different contributions to the geodesic distance  $D_g$ .

### 5.5.1 Definition

At the beginning of this chapter, we stated that the representation of an object given by the medial axis is capable to uncouple location and thickness, where location is represented by the position of the center of maximal spheres and thickness by their radius. As previously mentioned, the quantity  $|\mathbf{pole}(p) - \mathbf{p}|$  is used to approximate thickness in the discrete setting. By computing  $\tilde{T}(\mathbf{x})$  we gave a measure of the displacement of poles with respect to centerlines, which depends on object shape. We can now define a new quantity associated to each surface point  $p$ , named by us *normalized tangency deviation* [6, 5], as

$$\text{NTD}(p) = \frac{\tilde{T}(\mathbf{pole}(p))}{D_g(p)} = \frac{\tilde{T}(\mathbf{pole}(p))}{|\mathbf{pole}(p) - \mathbf{p}| + \tilde{T}(\mathbf{pole}(p))} \quad (5.25)$$

which is a normalized measure of how much pole displacement from centerlines weights on the overall geodesic distance of  $p$  to centerlines (see Figure 5.16). Obviously  $0 \leq \text{NTD}(p) \leq 1$ . The justification for the use of the terms *tangency deviation* comes from the following observation. The union of maximal spheres defined on a centerline yields a cylinder with variable radius which lies entirely inside the vessel surface. The contact points of the inscribed cylinder with the vessel surface are those points whose poles lie on the centerline. Therefore, surface points which are tangent to the inscribed cylinder have associated pole location displacement

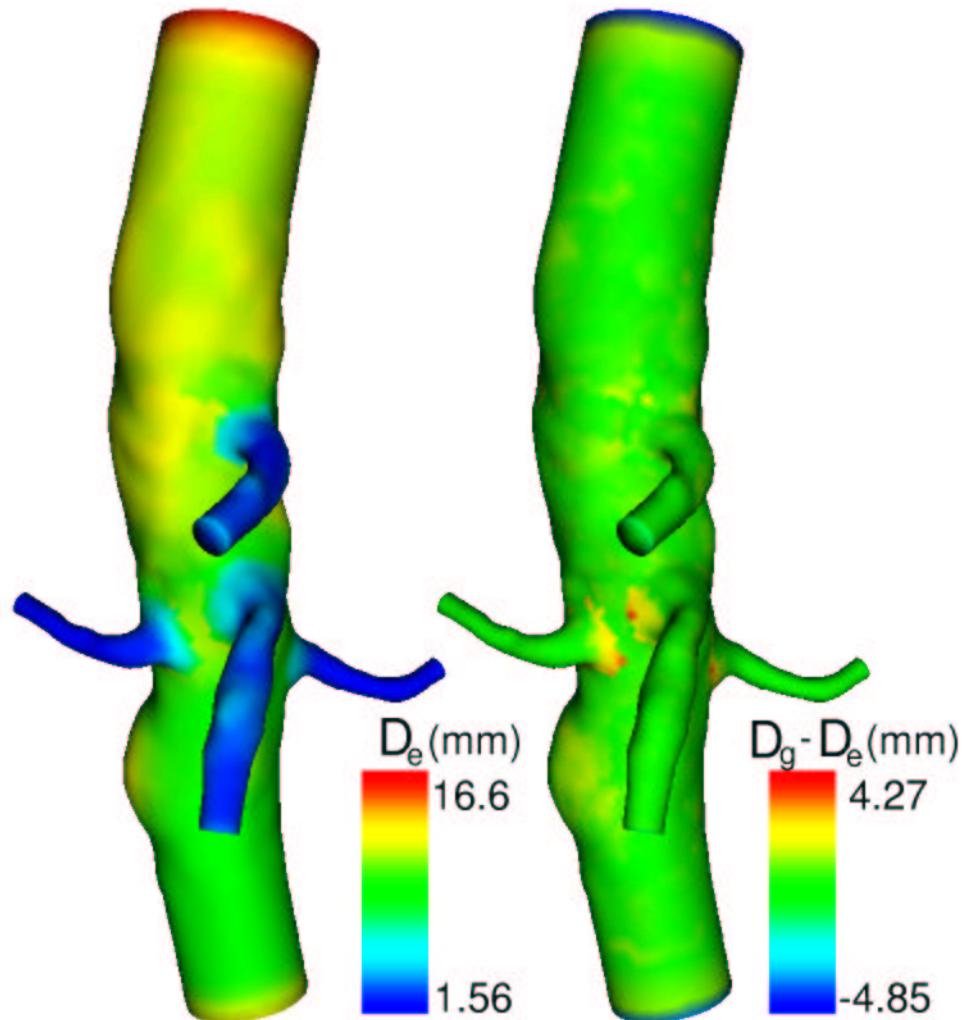


Figure 5.15: Left: Euclidean distance to centerlines computed on the abdominal aorta model surface. Right: difference between geodesic and Euclidean distance on the abdominal aorta model surface, showing that major differences are localized near bifurcations. Negative values of difference are caused by boundary effects.

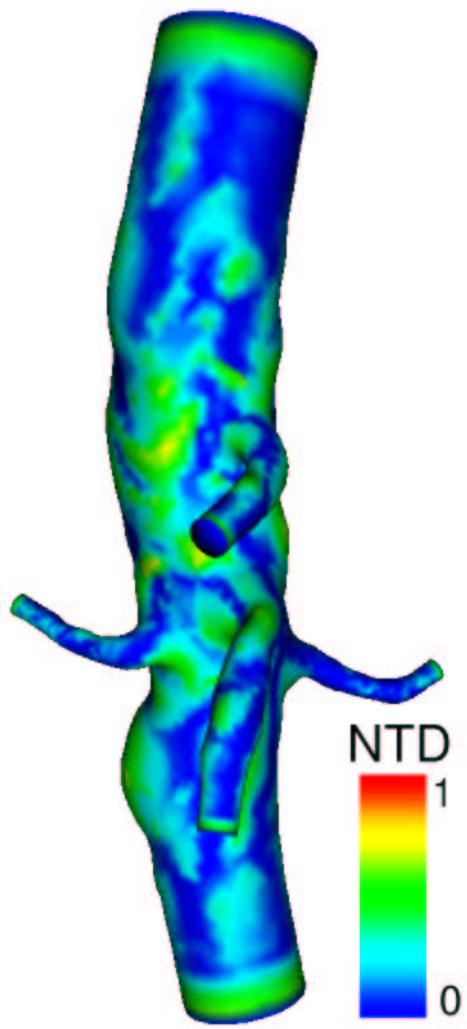


Figure 5.16: Distribution of normalized tangency deviation over the abdominal aorta model surface.

equal to zero, so that  $\text{NTD}(p) = 0$ . Conversely, if a pole is far from the centerline on the Voronoi diagram, it means that the surface point is located on a region which is not well represented by the inscribed cylinder, so that the inscribed cylinder will be far from being tangent to the surface in that point. In this case  $\text{NTD}(p) > 0$ . At the limit,  $\text{NTD}(p) \rightarrow 1$  when a surface point coincides with its pole, which is possible only at sharp corners and if point sampling density goes to infinity. Therefore  $\text{NTD}(p)$  is a normalized measure of how much a surface deviates from being tangent to the inscribed cylinder constructed by the envelope of the maximal spheres defined on centerlines.

Normalized tangency deviation is suited to characterize surface features. Dealing with blood vessel models, this quantity can be used to document the evolution of vessel surface morphology in serial acquisitions. Moreover, since  $\text{NTD}(p)$  is normalized with  $D_g(p)$ , normalized tangency deviation does not depend on vessel size, but only on surface shape. For this reason, it can be also employed to characterize and compare surface morphology in vessels of different scales in the same model, or in a population of models regardless their absolute size. This is a major improvement with respect to surface characterization performed on the basis of mean curvature, because mean curvature obviously depend on vessel radius.

### 5.5.2 Analytical expression for elliptical base cylinders

To better understand the meaning of normalized tangency deviation, the analytical expression for such quantity of an elliptical base straight cylinder is reported in the following. The choice of a straight cylinder with constant section along the centerline allows us to limit the calculation to a generic 2D section perpendicular to the cylinder centerline. We will therefore consider an ellipse on a 2D plane with center in  $(0,0)$ , semimajor (semiminor) axis  $a$  (resp.  $b$ ) oriented along  $x$  (resp.  $y$ ) axis. The parametric equation for such ellipse is

$$\begin{cases} x = a \cos t \\ y = b \sin t \end{cases} \quad (5.26)$$

with  $t \in [0, 2\pi]$ . On such plane, cylinder centerline is a point in the center of the ellipse. Maximal inscribed spheres, which have centers on the 2D plane of the ellipse, are circles inscribed into the ellipse, referred here as maximal inscribed circles.

Let's first derive the expression of the medial axis of the ellipse. Maximal inscribed circles must touch the ellipse in at least two points. Given the symmetry of the ellipse, the contact points will be symmetric with respect to the  $x$  axis. Therefore, the locus of centers of maximal inscribed circles is the intersection of the lines normal to the ellipse and passing through  $(a \cos t, b \sin t)$  and  $(a \cos t, -b \sin t)$ . The normal to the ellipse is the vector

$$\mathbf{n} = \frac{1}{\sqrt{b^2 \cos^2 t + a^2 \sin^2 t}} \begin{bmatrix} -b \cos t \\ -a \sin t \end{bmatrix} \quad (5.27)$$

Therefore, the line  $y = mx + q$  passing through point  $(a \cos t, b \sin t)$  and having  $m = n_y/n_x$  has equation

$$y = \frac{a}{b} \tan t x - \frac{a^2 - b^2}{b} \sin t \quad (5.28)$$

with  $t \neq \frac{\pi}{2}, \frac{3}{2}\pi$ . The intersection of such line with the  $x$ -symmetric one, for  $t' = 2\pi - t$ , yields the following equation for the locus of centers of maximal inscribed circles

$$\begin{cases} x_0 = \frac{a^2 - b^2}{a} \cos t \\ y_0 = 0 \end{cases} \quad (5.29)$$

where the condition  $t \neq 0, \pi$  has been used. The limit for  $t \rightarrow \frac{\pi}{2}$  yields the point  $(0, 0)$ , which is the center of the ellipse, while the limit for  $t \rightarrow 0$  yields the point  $(\frac{a^2 - b^2}{a}, 0)$ , which is the center of curvature of the ellipse in  $(a, 0)$ , coherently with the fact that the maximal inscribed circle is bitangent in that point (see Figure 5.17).

Let's now compute the value of NTD as a function of  $t$ . The term corresponding to  $\tilde{T}(\mathbf{x})$  is here expressed by  $\frac{a^2 - b^2}{a} \cos t$ , being the geodesic distance of a point of the medial axis from the centerline. The term corresponding to  $|\text{pole}(p) - \mathbf{p}|$  is expressed by the distance from  $(x, y)$ , point on the ellipse, to  $(x_0, y_0)$ , corresponding medial axis point (since we are dealing with a continuous surface, poles coincide with medial axis points), and takes the value  $b \sqrt{\frac{b^2}{a^2} \cos^2 t + \sin^2 t}$ . The expression for normalized tangency deviation over the ellipse as a function of  $t$  is therefore

$$\text{NTD}(t) = \frac{|e^2 \cos t|}{|e^2 \cos t| + \sqrt{(1 - e^2)(1 - e^2 \cos^2 t)}} \quad (5.30)$$

where terms have been rearranged as a function of ellipse eccentricity,  $e^2 = 1 - (\frac{b}{a})^2$ , which univoquely characterizes ellipse shape. Equation 5.30 evidences how normalize tangency deviation does not depend on object scale, but is a function of a shape-related quantity only. The limit values for normalized tangency deviation of the ellipse are  $\text{NTD}(\frac{\pi}{2}) = 0$ , at the contact point of the elliptical base cylinder with the maximal inscribed cylinder, and  $\text{NTD}(0) = e^2$ . A 2D plot of NTD as a function of ellipse eccentricity and parameter  $t$  is shown in Figure 5.18.

### 5.5.3 Generalization to arbitrary shapes

Although we derived the concept of normalized tangency deviation for interconnecting  $2\frac{1}{2}\text{D}$  geometries, the same concept can be easily extended to the characterization of surfaces of objects having arbitrary shape.

We have previously introduced that centerlines can be seen as the most stable medial axis subsets for  $2\frac{1}{2}\text{D}$  geometries. From this point of view, normalized tangency deviation can be extended to all the cases in which a stable subset is extracted from the medial axis [11, 37]. Normalized tangency deviation, defined as in

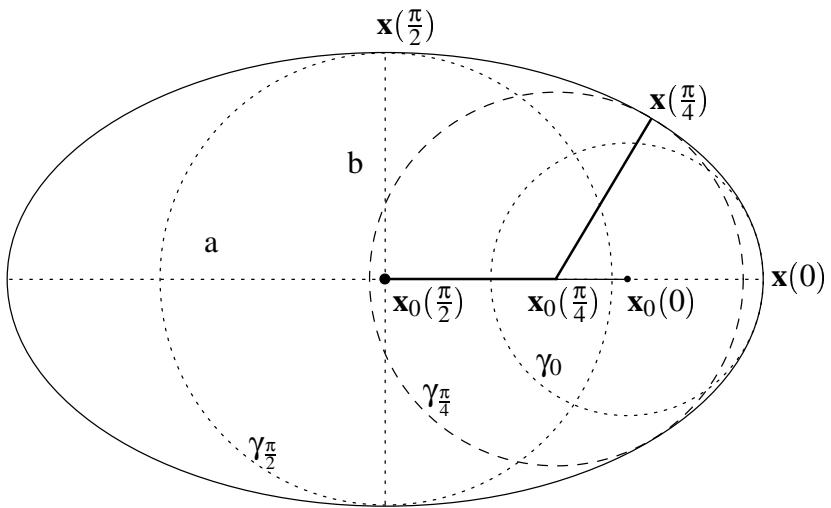


Figure 5.17: Normalized tangency deviation calculation for an ellipse with semi-major and semiminor axes  $a$  and  $b$ , respectively. Ellipse points  $\mathbf{x}$ , maximal inscribed circles  $\gamma$  and their centers  $\mathbf{x}_0$  are a function of ellipse parameter  $t$ . Half of the medial axis is represented as the line  $\mathbf{x}_0(0) - \mathbf{x}_0(\frac{\pi}{2})$ . Point  $\mathbf{x}_0(0)$  is the center of the circle  $\gamma_0$  which is bitangent in  $\mathbf{x}(0)$ , therefore it is the center of curvature of the ellipse in point  $\mathbf{x}(0)$ . Normalized tangency deviation is calculated as  $\text{NTD}(t) = \frac{|\mathbf{x}_0(t) - \mathbf{x}_0(\frac{\pi}{2})|}{|\mathbf{x}_0(t) - \mathbf{x}_0(\frac{\pi}{2})| + |\mathbf{x}(t) - \mathbf{x}_0(t)|}$  (see text for results).

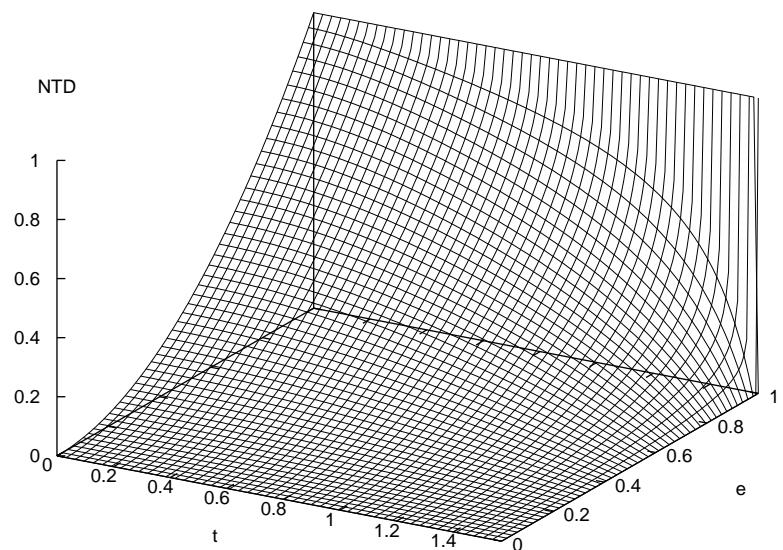


Figure 5.18: Plot of normalized tangency deviation against ellipse parameter  $t \in [0, \frac{\pi}{2}]$  and eccentricity  $e$ .

Aspect ratio	Error % [ $\mu(\sigma)$ ]	
	Centerline	Radius
5/5	0.01(0.02)	0.03(0.06)
4/5	0.24(0.19)	0.33(0.04)
3/5	0.84(0.78)	0.57(0.04)
2/5	2.17(1.34)	0.96(0.17)

Table 5.1: Validation results for elliptical base cylinders of different aspect ratio (semiminor to semimajor ratio). Percent errors of centerline position and maximal inscribed ball radius along centerline are normalized against cylinder maximum and minimum radius respectively. The number of vertices of the surface mesh vertex was 2975 for the four cylinders.

Equation 5.25, will then measure how the surface is far from being tangent to the envelope of the maximal spheres defined on the stable medial axis subset. In this case  $\tilde{T}(\mathbf{x})$  is computed with zero boundary conditions on the whole stable subset.

## 5.6 Implementation

The techniques described in this chapter were implemented in C++ using VTK libraries. Besides basic data structures and input/output operations, class `vtkDelaunay3D` was used for Delaunay tessellation of model surface points, and class `vtkPolyDataNormals` was used to generate consistently oriented surface normals to the surface, for external Delaunay tetrahedra removal.

## 5.7 Validation

The techniques presented in this chapter have been validated on synthetic objects of known geometry. Circular and elliptical base cylinders were first considered, testing the accuracy of centerline computation algorithm with different aspect ratios of the base ellipses (see Table 5.1), and with different sampling point densities (see Table 5.2). The results show that the centerline computation algorithm performance decreases when the section has a low aspect ratio. This is due to the fact that in these cases  $F(\mathbf{x})$  is near to be constant over the medial axis around the centerline, so that line position is more sensitive to numerical artifacts. This effect can be avoided by adding a term to  $E_{centerline}$  which penalizes centerline length [30]. The correctness of the interpretation is confirmed by the fact that maximal inscribed ball radius is accurately computed even in low aspect ratio models. As to dependence from point sampling density, the results show how the centerline computation algorithm is very robust when the section is circular, while for lower aspect ratio elliptical sections a good performance relies on an adequate surface sampling density. It must be noted, however, how the density required for accurate results

	Vertex no edge(tot)	Error % [ $\mu(\sigma)$ ] Centerline      Radius		Time (s)
A	40(2975)	0.01(0.02)	0.03(0.06)	12.34
	30(1693)	0.02(0.03)	0.10(0.15)	5.86
	20(757)	0.02(0.04)	0.25(0.35)	1.97
	10(198)	0.44(0.33)	1.38(1.20)	0.31
B	40(3628)	0.84(0.78)	0.57(0.04)	6.20
	30(2073)	0.74(0.72)	1.01(0.08)	3.18
	20(934)	2.88(1.95)	2.08(0.19)	1.21
	10(239)	4.42(4.68)*	7.48(0.79)	0.26

Table 5.2: Validation results for circular (A) and elliptical (aspect ratio 3/5) (B) base cylinders of different surface mesh density. Percent errors of centerline position and maximal inscribed ball radius along centerline are normalized against cylinder maximum and minimum radius respectively. \*This value is strongly influenced by inaccuracy near the extremities of the model, due to boundary effects; not considering centerline extremities leads to a value of 2.66(1.00).

is not high (30 points on section edge, which roughly means one point every 12 degrees along the section edge). High maximal inscribed ball radius measurement error for elliptical base cylinders of low mesh density is caused by the fact that in this case Voronoi balls are not tangent to the surface, but rather intersect it between sampling points, thus overestimating real inscribed ball radius.

Two configurations of interconnecting circular base cylinders which are representative of vascular branching types (Y and T bifurcations). Of particular interest is the accuracy of radius calculation in T bifurcations with respect to nearest centerline point calculation performed directly in  $\mathbb{R}^3$ . As shown in Figure 5.19, performing measurements over the Voronoi diagram using the inherent topological information is much more accurate in characterizing the bifurcation region. Boundary effects near outermost sections of the model (see Figure 5.19 left) are due to the fact that points in that regions share a common pole, which is also the pole of outermost section barycenter, so that direction  $(\text{pole}(p) - \mathbf{p})$  does not approximate surface normal.

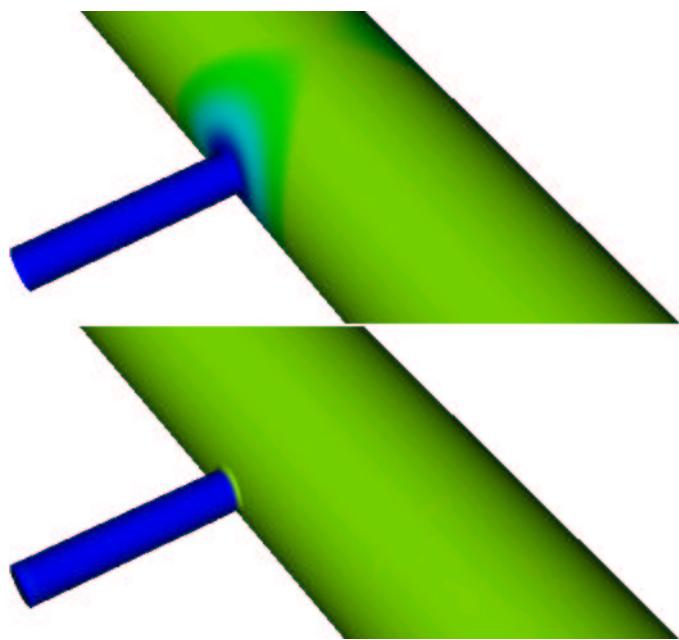


Figure 5.19: Upper: minimum distance from centerlines calculated in  $\mathbb{R}^3$  instead of on  $\text{Vor}_E$ , showing that the shaded region near the bifurcation is incorrectly assigned to the small cylinder. Lower: calculation of  $D_g$  for two cylinders arranged in a T junction, correctly discerning between the two cylinders.

# Chapter 6

## Mesh generation

### 6.1 Introduction

In the previous chapters the problem of accurately reconstructing 3D models of vascular segments from volumetric angiographic images was solved. Furthermore, accurate methods for geometric analysis of the reconstructed models were proposed, motivated either by the need of synthetically describe and compare vascular tracts both from different subjects and in time, and by the primary role of blood vessel geometry on blood flow. At this point, we introduce the problem of directly investigating local hemodynamics in the vessels of interest. As documented in Chapter 1, a reliable instrument for blood flow modeling at the patient-specific level is computational fluid dynamics (CFD), which consists in modeling blood flow by numerically solving the partial differential equations of fluid dynamics given the geometry of the domain and boundary conditions<sup>1</sup>.

Since flow field is defined in 3D, domain definition must be given in terms of a subset of  $\mathbb{R}^3$ . In particular, the numerical methods usually employed for CFD analysis, such as finite volumes and finite elements, require the discretization of the domain volume into small elements whose vertices, called nodes, are the points where the solution, velocity vectors and pressure values, is computed. The method employed to obtain the discretized domain, or mesh, along with the choice of the kind and shape of elements used in the discretization play a key role in the computation of accurate solutions. Indeed, thanks to the availability of robust CFD solvers, mesh generation usually turns out to be the most time-consuming and operator-dependent procedure, and it can heavily affect the quality of the solution. In general, the mesh must be dense enough to reproduce solution features, but mesh density must be balanced with the computational resources available. With this trade-off in mind, algorithms must be developed in order to generate meshes with properly distributed nodes and well-shaped elements [16].

For blood vessel modeling, in particular, mesh generation must rely on algo-

---

<sup>1</sup>Some sketches about the problem of fluid dynamics and the numerical methods used to solve it are given in the next chapter.

rithms which yield well-shaped elements for interconnecting generalized cylinders whose surface geometry and size have a great inter-subject variability. Along with this geometric flexibility requirement, and somehow in contrast with it, the nature of the problem modeled, viscous fluid flow inside vessels with no-slip boundary conditions on vessel walls, requires the creation of layers of thin elements adjacent to the vessel wall, the boundary layers, which are necessary to properly model the high solution gradients expected in that region. The use of boundary layers is made even more important when wall shear stress-related quantities have to be computed [90], since shear stress is a function of the gradient of velocity field, as shown in the next chapter. Adaptive refinement methods can also be employed, for which the mesh is made finer based on the residuals of the solution computed on a coarser grid.

In this chapter, we present two different approaches for mesh generation for blood vessel models, which differ from the choice of the element type and for the complexity of the geometries which can be discretized. Both approaches provide boundary layer generation. The starting point is a surface representing the vessel wall, such as that obtained with the techniques described in the previous chapters. Some of the meshing operations are performed with Gambit (Fluent Inc.) [54] commercial software, as it will be explicitly noted in the text. Since in the following we will refer to different finite element types, in Appendix B a brief description of the element types employed throughout the chapter.

## 6.2 Hexahedral meshing of arterial bifurcations

Hexahedral elements are usually employed in CFD when it is possible to orientate them along the expected direction of flow. In these cases numerical solutions are characterized by quick convergence and good accuracy [53]. As pointed out in the previous section, the major disadvantage of the use of hexahedral elements is the low flexibility of the geometries which can be discretized. In fact, given an object whose surface is meshed with quadrilaterals, no general rule exists to effectively perform a discretization which always yields well-shaped elements, although important results have been given on this topic [39].

For generalized cylinders, an effective approach for hexahedral mesh generation is represented by sweeping [17]. By definition  $2\frac{1}{2}$ D geometries are defined by the sweeping of a 2D face of variable shape along a 1D path. Based on this definition, sweeping meshing methods work by defining a node distribution on one outermost section face, called *source face* or *endcap*, and to sweep that distribution along the direction defined by the side surface, also called *linking surface*. Generation of internal points is the crucial part of this kind of algorithms, since naive implementation may give raise to poor quality or even self-intersecting elements.

Mesh generation tools implemented in commercial applications are usually tailored to deal with CAD models. Such models are usually defined by composition of shapes obtained by analytical functions, and their description is given by a hier-

archical organization of vertices, edges, faces and volumes. Taking advantage from such topological and geometric information, sweeping meshing tools are capable of finding (if exists) a decomposition of a CAD model into  $2\frac{1}{2}$ D subparts and mesh them with a sweeping approach in a conformal way (i.e. no node is located on the face of a neighbor element). The application of such tools to mesh generation of blood vessel models containing bifurcations is usually not successful, because the criteria used to partition the models into generalized cylinders fail to identify vessel branches. This is mainly due to the fact that blood vessels, unlike most mechanical parts, branch smoothly, therefore a general splitting criterion is not intrinsically defined.

In this section we present an approach based on automatically splitting of the models to discretize into generalized cylinders (this phase is implemented outside Gambit) and generating the mesh for each branch by sweeping (using Gambit algorithms), as presented by us in [4]. The application of the method presented is limited to models containing a single bifurcation, such as models of the carotid or iliac bifurcation. This limitation derives both from the difficulty of splitting a generic complex model in a compatible way, and from the inability of the meshing tool employed to perform sweeping when node distribution from both endcaps is prescribed.

### 6.2.1 Model decomposition

In this section we address the problem of finding a splitting criterion for a bifurcation such that the resulting branches are generalized cylinders. We will treat the case of a single Y bifurcation, for which the two child branches have a similar diameter, such as the carotid or the iliac bifurcations. In a previous work, Long et al. [65] addressed the problem of generating hexahedral meshes for arterial bifurcations, and solved it by splitting the models with a manual procedure. Their decomposition strategy does not give rise to generalized cylinders, but rather splits the vessels longitudinally along the axis. The resulting element quality is good along the vessels, but skewed elements are present around the bifurcation apex. Furthermore, boundary layers are not explicitly defined.

In Chapter 3 we presented a fast algorithm for topological analysis of vascular structures containing bifurcations. We recall that the algorithm consists in visiting the surface with a sphere, slightly greater than vessel section, which iteratively moves on the barycenter of its intersections with the vessel surface. The movement of the center of the sphere generates a line which is assumed to be vessel centerline. When the sphere generates more than two intersections with the surface, a bifurcation is identified, and the barycenter of the surface enclosed inside the sphere is assumed to be the point where the afferent branches meet. We made use of the capability of this algorithm to automatically identify bifurcation points and centerline orientations to find a splitting criterion for our bifurcation models.

The most straightforward way of splitting a Y bifurcation into three generalized cylinders is to perform three cuts, perpendicularly to the bifurcation plane, from the

center of the bifurcation to the three branching apices. This way the generalized cylinder endcaps joining at the bifurcation are made of two semi-disks of approximately equal size, and each two branches share a semi disk. The major problem of this approach is that two incident cutting planes may form an acute angle, so that the shape of the associated endcap doesn't allow to correctly parameterize the side wall of the generalized cylinder. As a result, a large number of elements generated during sweeping are result to be distorted.

In order to apply the splitting strategy just introduced while avoiding the generation of skewed elements, we propose to perform a triangular-based prismatic cut around the bifurcation center perpendicularly to the bifurcation plane prior to split the branches. Bifurcation plane is identified as the plane passing through the bifurcation barycenter and whose orientation minimizes the square distance from afferent centerline endpoints. The bifurcation plane is used as a reference for the construction of all cutting planes. The prismatic cut is performed with three planes normal to the bifurcation plane and to each afferent branch centerline, at a distance from bifurcation barycenter set as a fraction of the radius of the sphere enclosing the bifurcation. The intersections of each two planes together with the midpoint of the shortest segment between the respective enclosing sphere-model intersection profiles are then used to identify three more cutting planes. Such planes perform a cut which approximates bifurcation apices, therefore decomposing the model into generalized cylinders. The prismatic volume generated is also generalized cylinder, therefore it can be meshed by sweeping using hexahedral elements. The branch endcaps at the bifurcation are now made up of three parts, two semi-disks separated by a rectangle which is one side of the triangular-based prism. Again, each two branches endcaps share a semi-disk. Moreover, the three side faces of the prism have the same nodal distribution, so that the nodal distribution on the endcaps must be the same for all three branches. See Figure 6.1 for an example of result. The geometric construction just described reduces the generation of skewed elements because acute angles between incident cutting planes are transferred to the triangular base of the prism, which is a source face for sweeping along the prism. Therefore, skewed elements are not swept along the branches, but, if generated, they remain confined into the prism.

### 6.2.2 Hexahedral mesh generation with the Cooper algorithm

Once the model has been decomposed into generalized cylinders, single  $2\frac{1}{2}$ D pieces must be meshed by sweeping. The usual procedure to do this is to first discretize edges, that in our case coincide with the splitting lines, then source faces, and finally the volume. The splitting scheme adopted allows to determine cross-sectional node distribution by specifying node count  $n_a, n_b, n_c, n_d$  on the edges  $a, b, c, d$  in Figure 6.2. In order to obtain symmetric node distribution with respect to two orthogonal axes on the sections along the generalized cylinder, the following relations

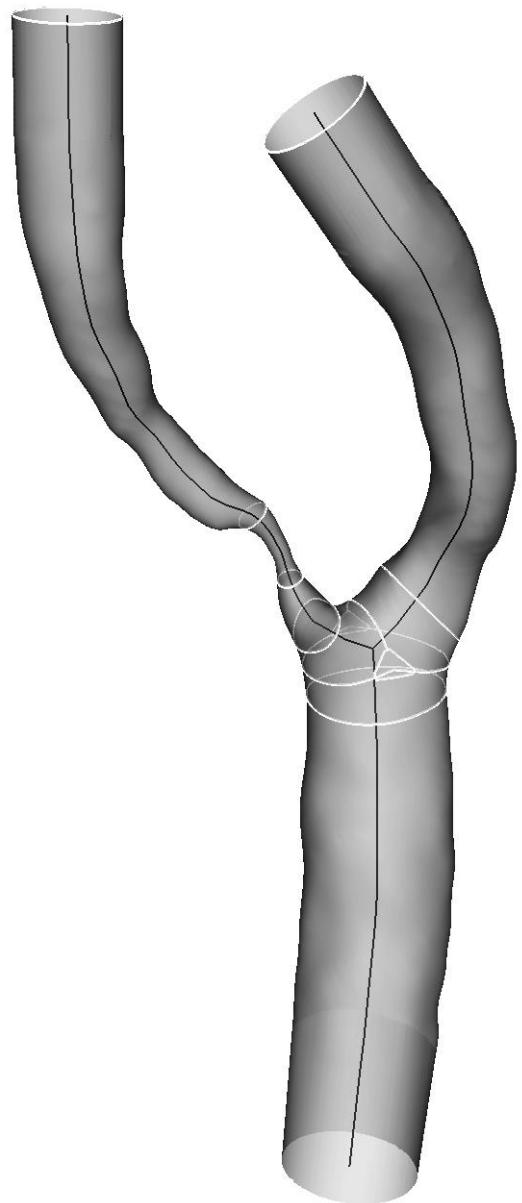


Figure 6.1: Automatic decomposition of a carotid bifurcation model based on approximate centerlines computed with the algorithm presented in Chapter 3. Notice that the stenotic region is automatically identified and isolated for subsequent adequate meshing.

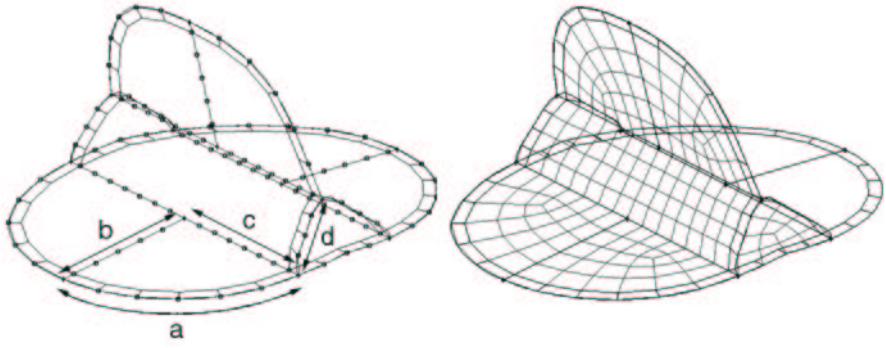


Figure 6.2: Edge meshing scheme on the decomposition edges of the bifurcation region.

among edge node counts must hold

$$\begin{aligned} n_a &= n_b \\ n_b + \frac{n_d}{2} &= n_c \end{aligned} \quad (6.1)$$

Therefore, only two of the four node counts are free. Before proceeding with source face surface meshing, boundary layers must be defined. This operation constrains a specified number of layers of volume elements at a specified distance from the boundary wall to have a given thickness [54]. Boundary layers can be only specified by absolute thickness. This constitutes a heavy limitation for automatic meshing, because vessels of different sizes, or tracts of different sizes in the same vessel caused by a stenosis, would require boundary layers adapted to the different sizes. This limitation is overcome by sweeping algorithms, because the cross-sectional node distribution is maintained along the sweep direction. In this case boundary layers have only to be defined for the source faces.

Source faces are meshed using simple mapping for quadrilateral side face of the triangular based prism on each branch endcap, and tri-primitive scheme for the rest of the source faces. Tri-primitive scheme is a simple scheme which allows to discretize a triangular patch into quadrilaterals (see Figure 6.2). Note how the composition of four disk-quarters and one rectangle on each branch source made following edge node count relations in Equation 6.1 yields a globally symmetric node distribution with respect to two orthogonal axes.

The sweeping algorithm used for subsequent volume meshing is the Cooper algorithm [17]. This algorithm begins by first projecting the edge discretization of the source face onto the opposite endcap on the basis of side surface parameterization. Secondly, the side face of the generalized cylinder is discretized with quadrilaterals by transfinite interpolation. Transfinite interpolation is a fast technique to map the points of a reference domain onto a physical domain once the description of its boundary is given. In our case, the side surface of the generalized

cylinder is mapped onto a reference rectangle for which one parameter coordinate is periodic. A cut can be induced in the side surface of the generalized cylinder based on surface parameterization such that it is mapped onto the reference rectangle whose parameter coordinate are  $(s, t)$ , with  $s \in [0, 2\pi]$  and  $t \in [0, 1]$ . By defining the set of blending functions  $\phi_0(s), \phi_1(s), \theta_0(t), \theta_1(t)$ , satisfying  $\phi_0(0) = 1$  and  $\phi_0(1) = 0$ ,  $\phi_1(0) = 0$  and  $\phi_1(1) = 1$ ,  $\theta_0(0) = 1$  and  $\theta_0(\pi) = 0$ ,  $\theta_0(0) = 0$  and  $\theta_0(\pi) = 0$ , transfinite interpolation in point  $\mathbf{x}$  is expressed by the *boolean sum*

$$(P_s \oplus P_t)[\mathbf{x}] = P_s[\mathbf{x}] + P_t[\mathbf{x}] - P_s P_t[\mathbf{x}] \quad (6.2)$$

where

$$\begin{aligned} P_s[\mathbf{x}](s, t) &= \phi_0(s) \mathbf{x}(0, t) + \phi_1(s) \mathbf{x}(1, t) \\ P_t[\mathbf{x}](s, t) &= \theta_0(t) \mathbf{x}(s, 0) + \theta_1(t) \mathbf{x}(s, \pi) \\ P_s P_t[\mathbf{x}](s, t) &= \phi_0(s) \theta_0(t) \mathbf{x}(0, 0) + \phi_1(s) \theta_0(t) \mathbf{x}(1, 0) + \\ &\quad \phi_0(s) \theta_1(t) \mathbf{x}(0, \pi) + \phi_1(s) \theta_1(t) \mathbf{x}(1, \pi) \end{aligned} \quad (6.3)$$

The simplest blending functions satisfying the above conditions for the side surface of a generalized cylinder are

$$\begin{aligned} \phi_0(s) &= \frac{1}{2} (1 + \cos(s)) \\ \phi_1(s) &= \frac{1}{2} (1 - \cos(s)) \\ \theta_0(t) &= 1 - t \\ \theta_1(t) &= t \end{aligned} \quad (6.4)$$

where the blending functions for  $s$  are periodic. One of the limitations of transfinite interpolation is that it propagates boundary singularities (i.e. sharp corners) onto the surface. This is the reason why particular attention is paid to avoiding distortion of source endcaps at the bifurcation, reason why the triangular-based prism cut at the center of the bifurcation is adopted.

The last step of mesh generation is the creation of internal nodes and elements, which is the core of the Cooper algorithm. Being a proprietary algorithm, the exact procedure is not well documented. The author refers to the method used as to a “*least square weighted residual method* which maintains the overall shape and quality of the mesh even for drastic distortion of the geometry” [17]. From our experience, the algorithm results to be very effective in preserving cross-sectional node distribution even in regions of high stenosis grade [4], as shown in Figure 6.3. Boundary layer elements maintain the same aspect ratio with respect to the rest of cross-sectional elements, which is a great advantage of using the Cooper algorithm in this context. Therefore, the meshes generated by the described approach meet the requirements of being well-aligned with the expected direction of flow and presenting scale-dependent boundary layers. The resulting volume elements are linear hexahedra of high quality. Figure 6.4 shows the final hexahedral mesh for

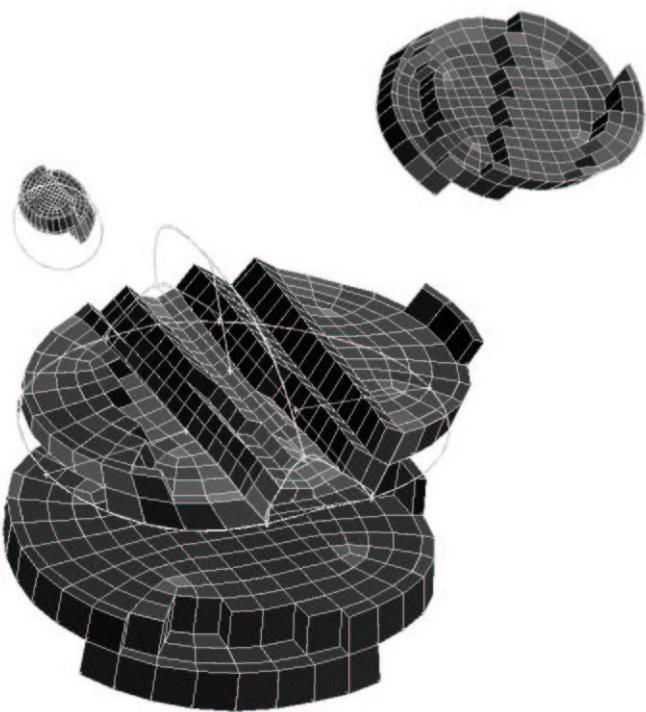


Figure 6.3: Visualization of the volume mesh resulting from Cooper scheme meshing of the single subparts. Cross-sectional node distribution is well-preserved along the branches, even in the highly stenotic tract, with boundary layer thickness correctly scaling with vessel diameter, and well-shaped elements are obtained in the branching region around the triangular-based prism cut.

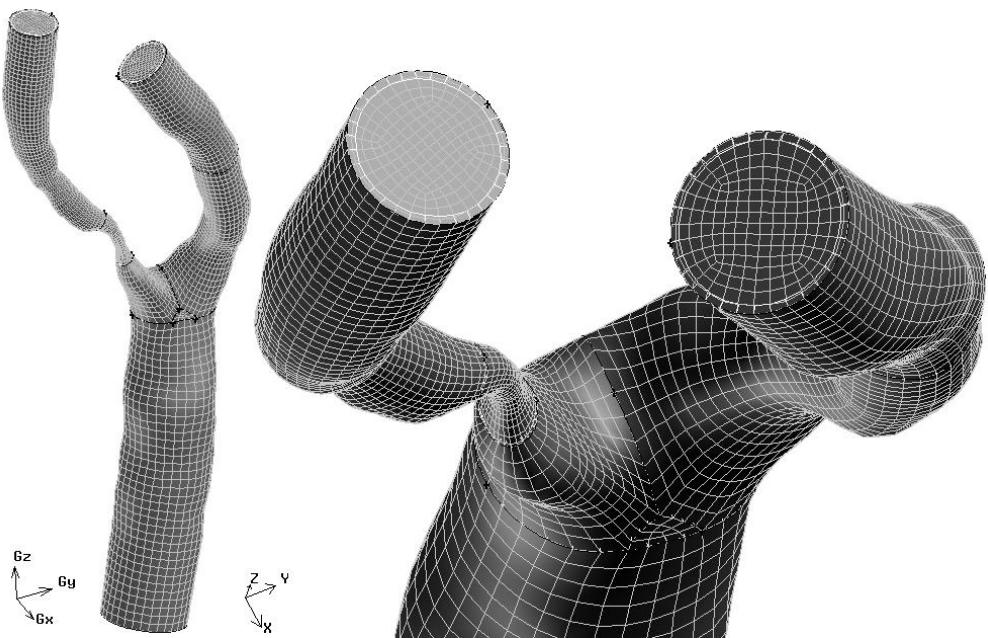


Figure 6.4: Left: hexahedral mesh for the stenotic carotid bifurcation model. Right: particular of the discretization obtained at the bifurcation apex.

the stenotic carotid bifurcation considered in this section, while Figure 6.5 shows the hexahedral mesh for the controlateral carotid bifurcation model obtained from the same acquisition.

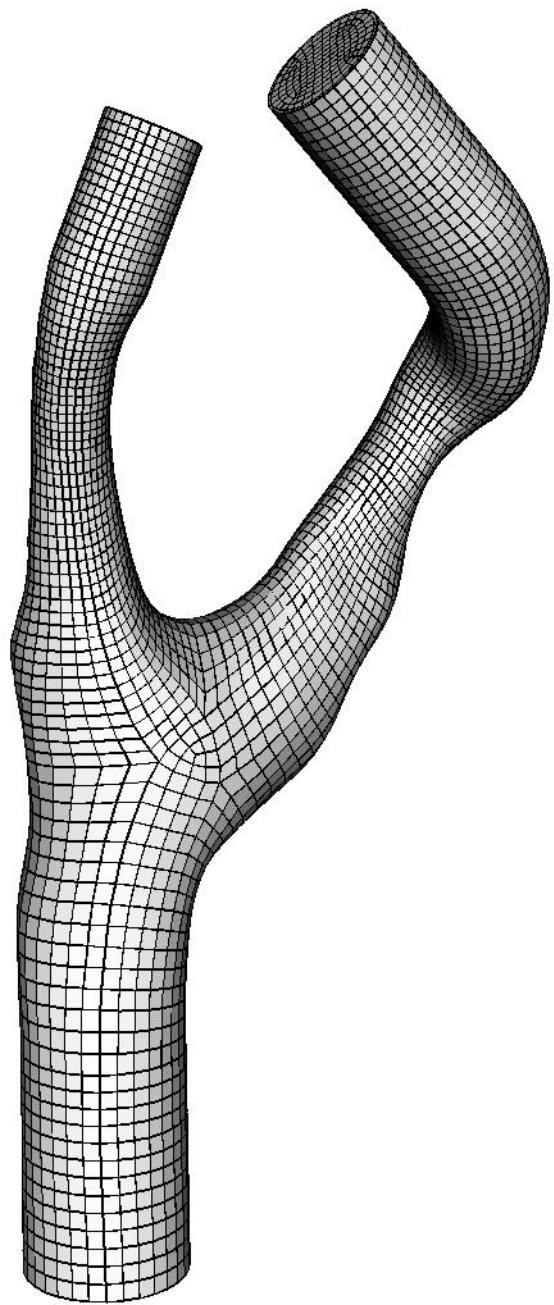


Figure 6.5: Hexahedral mesh for the carotid bifurcation contralateral to that in Figure 6.4 presenting different geometric features.

## 6.3 Hybrid meshing of complex arterial segments with adaptive boundary layers

In the previous section an effective method to generate computational meshes made up of hexahedral elements for arterial bifurcations was presented. The major limitation of the above method was the difficult extension of the application to models containing more than a single bifurcation. When the geometry and the topology of blood vessel models increase in complexity, the definition of bifurcation and splitting planes can be not straightforward, and the application of the sweeping algorithm to branches whose both endcaps have constrained node distribution is frequently unsuccessful. Even if much work has been done on single bifurcation models, such as the carotid or the iliac bifurcation, which are interesting cases of atherosclerotic plaque preferential sites, a more general meshing technique is needed to be provided. The lower flexibility of hexahedral elements gives rise to one last limitation of the previous method, that is the difficulty of locally refining the mesh, preserving conformity, where the CFD solution requires it, for example in regions of complex flow or high solution gradients.

Tetrahedral elements, being simplices in 3D, are more flexible than hexahedra, since their use in volume discretization needs not to meet any particular topological requirement. On the other hand, the simple application of a tetrahedral meshing scheme to blood vessel models for CFD computation presents some serious disadvantages, the first of which is the difficulty of defining boundary layers. Even if local refinement or the generation of flat tetrahedral elements can be prescribed near the wall, tetrahedral shape functions are inherently less accurate than that of hexahedral elements in catching the high gradients which develop normal to the vessel wall. Moreover, not making use of sweeping and preservation of cross-sectional node distribution, boundary layer thickness must be specified with its absolute value.

A good compromise between flexibility and accuracy is represented by hybrid meshes, which employ wedge elements for boundary layers and tetrahedral elements for the inner volume. Wedge elements are created by inward extruding surface triangular elements. The choice of the extrusion direction has a great influence on the quality of the final mesh, also considering the importance of boundary layer elements on CFD results.

In the following we will show how it is possible to produce high quality hybrid meshes in which boundary layer thickness automatically adapts to vessel size, even in bifurcation regions, with no need of model decomposition (as presented by us in [6]). The method described here is based on geometric analysis techniques presented in Chapter 5. For reasons related to CFD computation accuracy and stability, which will be made clear in the next chapter, we will employ tetrahedral and wedge elements with quadratic interpolation.

### 6.3.1 Boundary layer generation with adaptive thickness wedges

In Chapter 5 we presented a method to compute a measure of the distance of surface points to centerlines. In particular, the geodesic distance of surface points to centerlines computed on the embedded Voronoi diagram was calculated by first solving the Eikonal Equation (5.11), whose solution was indicated with  $\tilde{T}(\mathbf{x})$ ,  $\forall \mathbf{x} \in \text{Vor}_I(P)$ , where  $P$  is the point set sampling the boundary of the model. The geodesic distance of a surface point  $p$  to centerlines was then defined as  $D_g(p) = |\mathbf{pole}(p) - \mathbf{p}| + \tilde{T}(\mathbf{pole}(p))$ , where  $\mathbf{pole}(p)$  is the farthest inner Voronoi vertex associated to point  $p$ .

The availability of  $D_g$  allows to define boundary layer thickness as a fraction of vessel size, without the need of using sweeping algorithms or to split the model. In fact, a kind of sweeping is performed on the Voronoi diagram when calculating centerlines (see Chapter 5). This way, if triangular surface mesh is extruded inward of a fraction of  $D_g$ , one or more layers of wedges can be created. The quality of such elements depend on the choice of warping directions for surface points. Therefore, two problems have to be solved. First, providing a well shaped triangular surface mesh, second, defining proper warping directions.

Surface meshes generated by Marching Cubes algorithm and subsequent smoothing and fairing procedures are usually not adequate for CFD analysis. Marching Cubes algorithm generates surface meshes whose density depends on the resolution of the source images, independently from surface features, therefore meshes are usually overresolved. Moreover, the presence of bad shaped triangular elements can be frequently observed. To overcome this limitations, edge collapsing and diagonal swapping techniques have been proposed [26], along with smoothing filters, in part introduced in Chapter 4, which improve triangle shape and density. Although these tools are widely used with success in the field of modeling and visualization, we prefer to create a completely new mesh based on model surface using Gambit advancing front surface meshing algorithm<sup>2</sup>. The resulting surface mesh is made up of high quality triangular elements, whose density can be controlled by edge discretization density. A more sophisticated approach would be to perform remeshing directly on the background mesh represented by model surface, as proposed in [64, 100]. This way,  $D_g$  could also control surface mesh density, higher in smaller vessels. Actually this constitutes a future direction.

After good surface discretization is obtained, the centerline computation method proposed in Chapter (5) is applied and  $D_g$  calculated for each surface mesh vertex. At this point only warping directions remain to be defined. For this task we recall the concept of poles. As reminded above,  $\mathbf{pole}(p)$  is the farthest inner Voronoi vertex among the Voronoi vertices of  $p$ . It has been pointed out in Chapter (5) that the direction  $(\mathbf{pole}(p) - \mathbf{p})$  approximates surface normal, because  $\mathbf{pole}(p)$  approaches to the continuous medial axis when point sampling density increases. From a slightly different point of view,  $(\mathbf{pole}(p) - \mathbf{p})$  is the straight line along

---

<sup>2</sup>Since the 3D version of the advancing front algorithm will be presented later in this chapter for tetrahedral inner volume meshing, we will not give a description of this approach at this point.

which the distance from neighboring surface points to  $p$  is maximized. This last assertion is easily depicted by letting a sphere touching  $p$  grow from  $p$  by moving its center  $c$  inside the object. The path of  $c$  from  $p$  to  $\text{pole}(p)$  is the longest straight path which ensures that the sphere stays empty. In fact, since Voronoi cells are convex (see Chapter 5) and both  $p$  and  $\text{pole}(p)$  belong to  $\text{Vor}(p)$  by definition, the path  $(\text{pole}(p) - \mathbf{p})$  is completely contained into  $\text{Vor}(p)$ , and the sphere is empty along the path. Since a sphere is completely defined by its center and one point on its surface, our sphere, defined by  $c$  and  $p$ , will touch the neighboring points only in  $\text{pole}(p)$ , otherwise  $c$  could not be moved any further (i.e. Voronoi cell boundary would be reached).

From these considerations it comes out that convenient warping directions are represented by  $(\text{pole}(p) - \mathbf{p})$  for each surface mesh vertex, because surface triangle vertices stay away from each other during warping. Of course, no guarantee is provided to avoid triangle self-intersection beyond the poles, therefore if boundary layer absolute thickness is greater than the smallest point-to-pole distance (also called object *thickness* in point  $p$ ), more subtle warping direction definition should be adopted. From our experience, however, model features and required boundary layer thicknesses, usually up to  $0.25 * D_g$ , provide warped meshes of high quality (see Figure 6.6). A possible improvement would be that of considering piecewise linear warp directions, defined along the geodesic paths from surface points to centerlines (computed in Chapter 5 for the calculation of the Euclidean distance  $D_e$ ).

In order to produce quadratic 18-node wedge elements, surface linear triangles, used in the calculations performed so far, have to be converted into quadratic 6-node triangles. The additional nodes are then warped interpolating warping directions defined on triangle vertices, thus creating 18-node wedges (see Figure 6.7). To perform linear to quadratic triangle conversion with an adequate geometric accuracy, we adopted a subdivision approach. Subdivision schemes are employed in multiresolution modeling to obtain a fine mesh from a coarser mesh of the same object. When triangular surface meshes are subdivided, each triangle gives rise to four smaller triangles by inserting three nodes which split triangle edges. If node placement is performed accurately, the surface resulting from infinite subdivisions can result to be  $C_1$  or even  $C_2$ . Several subdivision schemes for triangle meshes exist, among which Loop's scheme and Butterfly scheme are the most popular ones [140]. They differ in that Loop's scheme is an approximating scheme (i.e. the vertices of the coarser mesh do not necessarily maintain their position in the finer one) but it is capable to yield  $C_2$  continuous surfaces, while Butterfly scheme is an interpolating scheme, which yields  $C_1$  surfaces but node positions of the coarser mesh are maintained in the finer one. For our means we employ Butterfly node placement scheme, as shown in Figure 6.8, in order to ensure that original mesh positions are not modified.

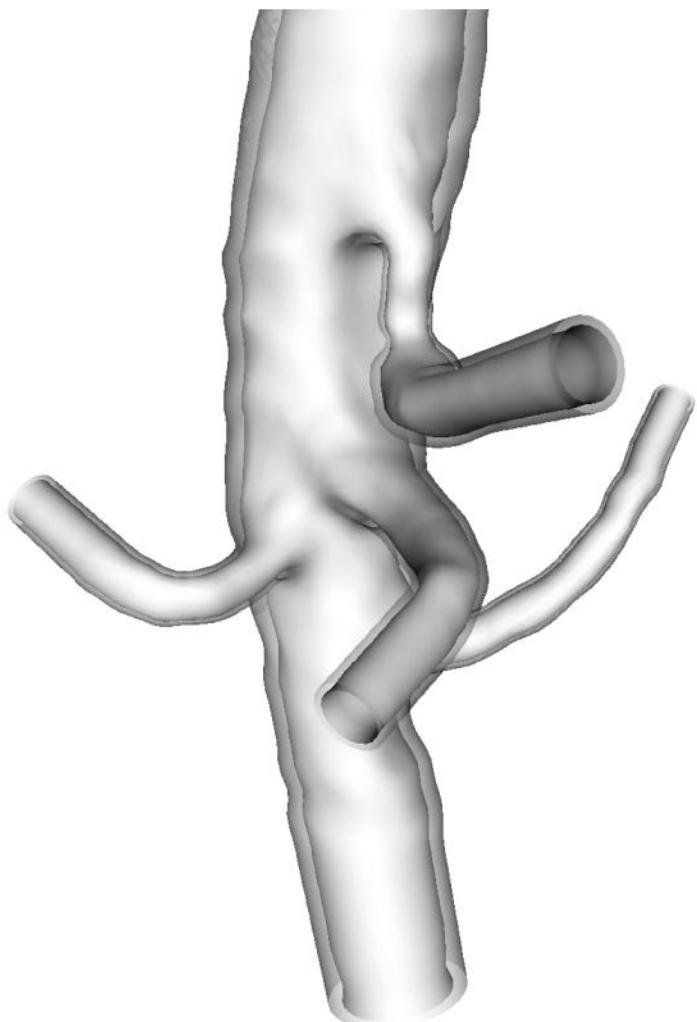


Figure 6.6: Warping of surface mesh based on geodesic distance to centerlines for the abdominal aorta model.

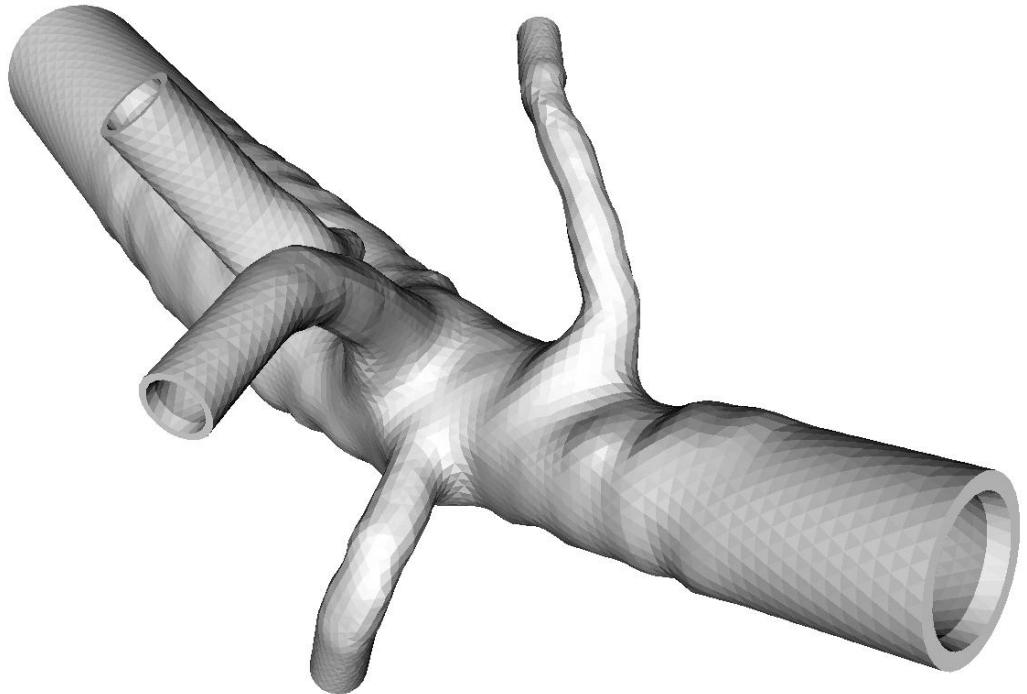


Figure 6.7: Adaptive thickness boundary layer for the abdominal aorta model.

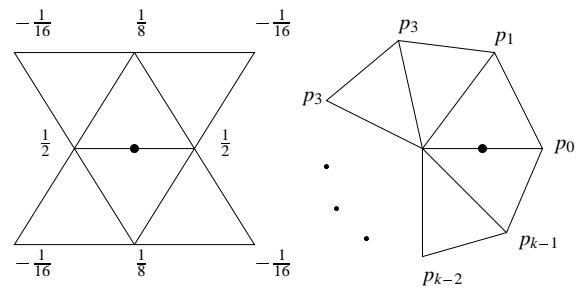


Figure 6.8: Butterfly subdivision scheme for regular (6-neighbors) vertices (left) and for extraordinary ( $\neq$  6-neighbors) vertices, with  $p_i = \frac{1}{k} \left( \frac{1}{4} + \cos \frac{2i\pi}{k} + \frac{1}{2} \cos \frac{4i\pi}{k} \right)$  for  $k > 5$ ,  $p_0 = \frac{5}{12}$ ,  $p_{1,2} = -\frac{1}{12}$  for  $k = 3$ ,  $p_0 = \frac{3}{8}$ ,  $p_2 = -\frac{1}{8}$ ,  $p_{1,3} = 0$  for  $k = 4$

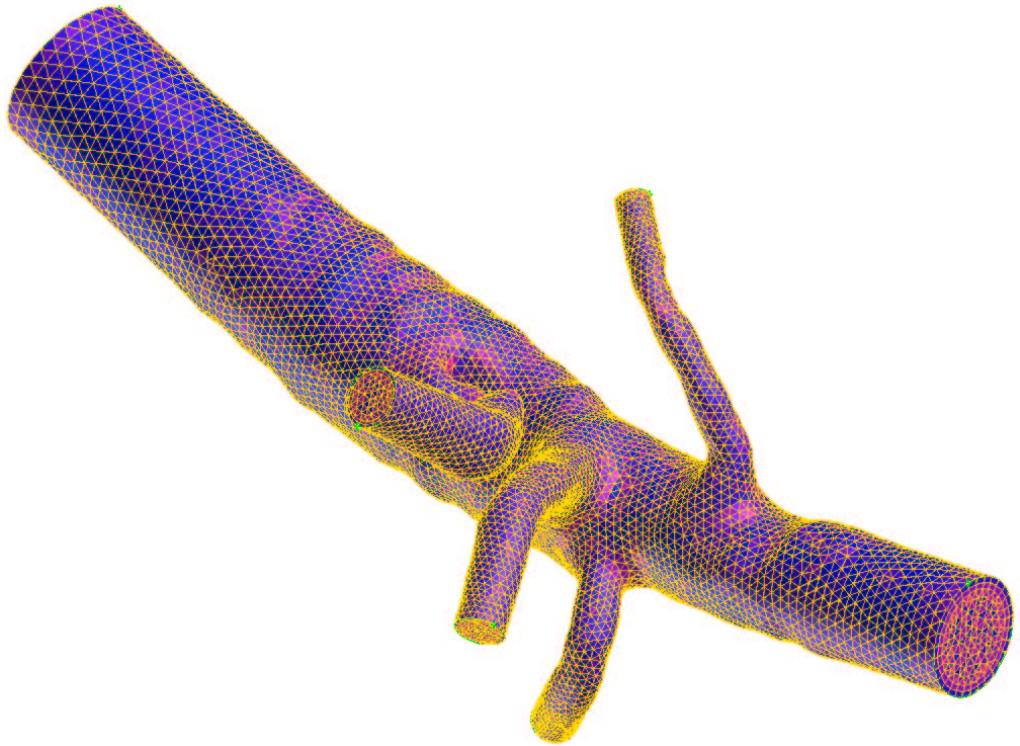


Figure 6.9: Final hybrid wedge-tetrahedra mesh for the abdominal aorta model.

### 6.3.2 Tetrahedral volume meshing by the advancing front approach

After the creation of boundary layers of wedges having thickness automatically adapted to vessel size, the last step is to fill the inner volume with quadratic 10-node tetrahedra (see Figure 6.9 and Figure 6.10). For this mean we use the advancing front tetrahedral meshing scheme implemented in Gambit [54]. Advancing front schemes work by adding elements from the boundary (the initial front) to the interior of the domain based on local criteria [16]. In our case the initial front is represented by the warped surface mesh. Each triangle of the front is assigned a priority value. Elements are added by selecting the triangle with the highest priority, removing it from the front, forming a new tetrahedron by inserting a new point in an appropriate position in the vicinity of the selected triangle, and updating the front. Although the base concept of this approach is simple, several issue have been arisen on priority definition and point insertion, for example to properly handle front collision. A number of tests must be performed on the neighborhood when a new point has to be inserted. Gambit employs a meshing scheme in which Delaunay criterion is used to ensure that the new point is properly inserted. Even if the theoretical guarantees provided by the advancing front method are not as sound as other unstructured meshing methods, such as Delaunay tessellation, which can

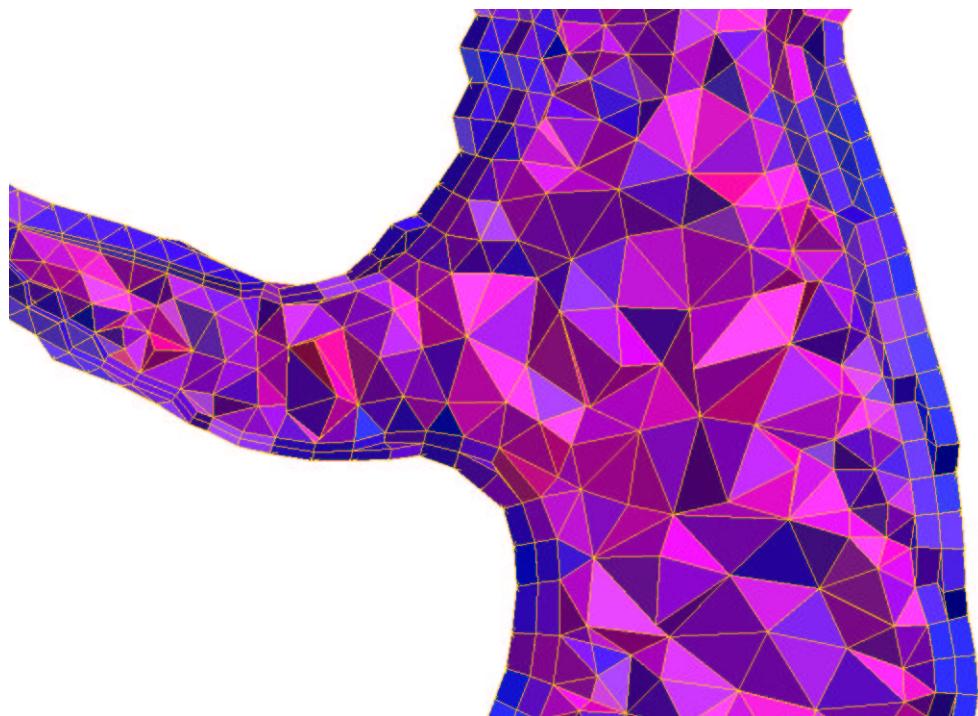


Figure 6.10: Detail of the hybrid mesh (10-node tetrahedra and 18-node wedges) generated for the abdominal aorta model at the level of the left renal bifurcation, showing the good grading of boundary layer thickness.

for example provide a lower bound for the minimum angle in elements, from our experience meshes generated with advancing front schemes result to have good element grading and high quality elements, at least for the class of models employed in this work.

## 6.4 Implementation

The techniques presented in this chapter have been implemented in C++ using VTK libraries. For hexahedral mesh generation, the code developed in Chapter 3 is employed in centerline and bifurcation identification phase. In the hybrid mesh generation, the code developed in Chapter 5 is employed in the boundary layer generation phase. In order to easily import and export surface and volume meshes between our code and Gambit, FDNEUT format reader and writer VTK classes were implemented.

## 6.5 Validation

There are many criteria to characterize the quality of mesh elements, among which we report the distribution of *equiangle skew*, defined as

$$Q_{EAS} = \max \left\{ \frac{\theta_{\max} - \theta_{eq}}{180 - \theta_{eq}}, \frac{\theta_{eq} - \theta_{\min}}{\theta_{eq}} \right\} \quad (6.5)$$

where  $\theta_{\max}$  and  $\theta_{\min}$  are the maximum and minimum angles in degrees between the edges of the element, and  $\theta_{eq}$  is the characteristic angle of each element type. For hexahedral elements  $\theta_{eq} = 90$ , while for tetrahedral elements  $\theta_{eq} = 60$ .  $Q_{EAS}$  values are in the range  $[0, 1]$ , higher quality being associated to lower values. In general, as a rule of thumb, high quality meshes have  $Q_{EAS} < 0.4$ .

In Table 6.1 the distribution of equiangle skew for the hexahedral meshes of two carotid bifurcation models (shown in Figures 6.4 and 6.5) generated with the model decomposition method is reported.

Table 6.2 shows the distribution of equiangle skew for the hybrid mesh of the abdominal aorta shown in Figure 6.9.

$Q_{EAS}$	A		B	
	N	%	N	%
0.0–0.25	31202	82.91	23703	79.65
0.25–0.5	5950	15.81	5917	19.88
0.5–0.75	475	1.26	140	0.47
0.75–1.0	5	0.01	0.00	0.00
0.0–1.0	37632	100.00	29760	100.00

Table 6.1: Mesh quality evaluation for the hexahedral meshes of carotid bifurcation models in Figure 6.4 (A in the table) and Figure 6.5 (B in the table).  $Q_{EAS}$ : equiangle skew (see text for details). Maximum  $Q_{EAS}$  is 0.7767 for model A, and 0.6094 for model B.

$Q_{EAS}$	Global		Wedges		Tetrahedra	
	N	%	N	%	N	%
0.0–0.25	34475	37.25	27186	29.38	7289	7.88
0.25–0.5	48603	52.52	4775	5.16	43828	47.36
0.5–0.75	9415	10.17	133	0.14	9282	10.03
0.75–1.0	53	0.06	2	0.00	51	0.06
0.0–1.0	92546	100.00	32096	34.68	60450	65.32

Table 6.2: Mesh quality evaluation for the hybrid meshes of the abdominal aorta model in Figure 6.9. The result for the totality of the elements as well as separately for wedges and tetrahedra are given.  $Q_{EAS}$ : equiangle skew (see text for details). Maximum  $Q_{EAS}$  is 0.897 (tetrahedron).



## Chapter 7

# Computational hemodynamics

### 7.1 Introduction

In the previous chapters we presented a set of techniques to reconstruct accurate 3D models of vascular structures and to geometrically characterize them at various levels of detail. Moreover, we faced the problem of how a reconstructed model can be modified in order to consistently remove unwanted features or locally edit its surface to make it conform to application-specific requirements. Finally, we presented two alternative methods to discretize model surfaces and volumes in order to generate meshes for the numerical approximation of partial differential equations such that arising in blood flow modeling.

We will now introduce the fluid dynamics problem into the process of patient-specific modeling of blood vessels. Since it is far beyond the scope of this work to give a detailed description of the physical, mathematical and numerical aspects of computational fluid dynamics, we will limit to the essential description of the equations involved and the numerical methods here employed, referring to the literature for further details [14, 52].

In the following we will treat the case of rigid walls only, although advances in modeling techniques and the availability of computational resources will likely allow to introduce the fluid-structure interaction problem as a standard in the near future.

### 7.2 Navier-Stokes equations

We consider blood as an homogeneous, incompressible, constant-density, viscous fluid. Moreover, the effect of gravity is neglected, and no thermal effects are considered. Vascular walls are modeled as non-permeable, rigid walls. Under these assumptions, the fluid dynamics problem is expressed by the conservation of the two physical properties of mass and momentum, giving rise to the following system

of partial differential equations:

$$\begin{cases} u_{j,j} = 0 \\ \rho \frac{\partial u_i}{\partial t} + \rho u_j u_{i,j} = -p_{,i} + \tau_{ij,j} \end{cases} \quad (7.1)$$

where the first equation describes mass conservation, and the second equation momentum conservation (the index summation convention has been used) in the domain  $\Omega$ . In Equation 7.1,  $u_i$  represents velocity, while  $p$  and  $\tau_{ij}$  are pressure and viscous stress tensors respectively, which are the isotropic and deviatoric parts of the stress tensor  $\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$ . The system is completed by the following initial conditions

$$u_i(x_j, 0) = u_i^0(x_j) \quad x_j \in \Omega \quad (7.2)$$

and boundary conditions

$$\begin{aligned} u_i(x_j, t) &= \bar{u}(x_j) & x_j \in \Gamma_D, t > 0 \\ \sigma_i(x_j, t) &= \sigma_{ij} n_i(x_j, t) = \bar{\sigma}_i(x_j) & x_j \in \Gamma_N, t > 0 \end{aligned} \quad (7.3)$$

where  $n_j(x_j)$  is the outward oriented normal to the boundary, and  $\Gamma_D$  and  $\Gamma_N$  constitute a partition of the domain boundary  $\partial\Omega$  (therefore  $\Gamma_D \cap \Gamma_N = \emptyset$  and  $\Gamma_D \cup \Gamma_N = \partial\Omega$ ).

For a viscous isotropic incompressible fluid the constitutive relation between  $\tau_{ij}$  and the strain rate tensor  $d_{ij} = \frac{1}{2}(u_{i,j} + u_{j,i})$  is of the type

$$\tau_{ij} = 2\mu d_{ij} \quad (7.4)$$

where  $\mu$  is fluid viscosity. Introducing such relationship into the momentum conservation equation yields the *stress divergence* form of the momentum conservation equation

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j u_{i,j} = -p_{,i} + [\mu(u_{i,j} + u_{j,i})]_{,j} \quad (7.5)$$

If viscosity is assumed constant, using the incompressibility constraint (mass conservation) leads to the *Navier-Stokes* form of the momentum conservation equation

$$\rho \frac{\partial u_i}{\partial t} + \rho u_j u_{i,j} = -p_{,i} + \mu u_{i,j,j} \quad (7.6)$$

It is possible to recognize in the last equation a momentum time-dependence term  $\rho \frac{\partial u_i}{\partial t}$ , a momentum advection term  $\rho u_j u_{i,j}$  and a momentum diffusion term  $\mu u_{i,j,j}$ . The role of the pressure term is peculiar for an incompressible fluid, since it is not related to density by an equation of state, and it only appears through its gradient in the momentum conservation equation. Pressure can be interpreted as a Lagrange multiplier for the saddle point problem of finding  $u_i$  which minimizes the energy associated to momentum conservation under the constraint of incompressibility.

## 7.3 Constitutive laws for blood rheology

As pointed out in the previous section, the relationship between the deviatoric stress tensor and the strain rate tensor models the rheologic behavior of the fluid. Again, a comprehensive description of blood rheologic models is beyond the scope of this work. In the following we will focus on the two models employed later in this work.

Briefly, blood is a suspension of aggregatable particles, among which red blood cells play a major role, from a rheologic point of view. Therefore, it is a viscoelastic fluid, with viscosity and elastic modulus depending on shear rate and time [121, 89]. Its behavior is of the *shear-thinning* type, with apparent viscosity decreasing with increasing values of shear rate, due to the disruption of red blood cells aggregates.

### 7.3.1 Newton's law

The simplest constitutive equation is the Newton law, which consists in a linear relationship between stress and strain, namely

$$\tau_{ij} = 2\mu d_{ij} \quad (7.7)$$

where viscosity  $\mu$  is constant, therefore independent from kinematic quantities. The value of blood viscosity for the Newtonian model depends on hematocrit and plasma protein concentration. In case these data are available from blood sample analysis, blood viscosity can be determined by the following empirical relationships [56, 95], as in [38]

$$\begin{aligned} \mu &= \mu_{rel} \mu_{pl} \\ \mu_{rel} &= (1 - 0.5H_t k)^{-2} \\ \log(k) &= 1.3435 - 2.083H_t + 2.711H_t^2 - 0.6479H_t^3 \\ \mu_{pl} &= 0.204 + 0.177C_p \end{aligned} \quad (7.8)$$

where plasma protein concentration  $C_p$  is expressed in  $g/dl$  and plasma viscosity  $\mu_{pl}$  in  $cPoise$ . If blood sample data are not available, blood viscosity for Newtonian fluid models can be assumed equal to the physiologic value of  $3.5 cPoise$ . Modeling blood as a Newtonian fluid has been commonly performed in the literature, and yields better results for larger vessels. Perktold et al. [89] pointed out how the errors deriving from employing a Newtonian model for blood yield non essential differences in flow characteristics and wall shear stress distributions.

### 7.3.2 Carreau's law

When more accurate modeling of blood rheology is required, such as in domains in which particularly low shear rates are expected, more complex shear-thinning

constitutive equations must be taken into account. In general, rheologic models which are expressed in the following form

$$\tau_{ij} = 2\mu(\dot{\gamma}) d_{ij} \quad (7.9)$$

where viscosity is a function of the shear rate

$$\dot{\gamma} = \sqrt{2d_{ij}d_{ij}} \quad (7.10)$$

take the name of *generalized* Newtonian models. A popular generalized Newtonian model is represented by the power law model [52], expressed by

$$\mu = \mu_0 K \dot{\gamma}^{n-1} \quad (7.11)$$

where  $K$  is a constant. The power-law models the shear-thinning behavior for  $n < 1$ , but yields unphysical behavior for low and high shear rates, so that more sophisticated models are preferred, such as Casson's [89].

One of the most used rheologic models for blood is Carreau's law [52], expressed by

$$\mu = \mu_\infty + (\mu_0 - \mu_\infty)(1 + K^2\dot{\gamma}^2)^{\frac{n-1}{2}} \quad (7.12)$$

where  $\mu_0$  and  $\mu_\infty$  are low and high shear rate asymptotic values, and parameters  $K$  and  $n$  control the transition region. Parameter  $\mu_\infty$  can be obtained with the same method used for viscosity in the Newtonian model. The other parameters can be derived by fitting experimental data, such as the ones reported by Brooks [23]. For example, for  $H_t = 35.1\%$ ,  $C_p = 7.0 \text{ g/dL}$ , from Equations 7.8 it results  $\mu_{infty} = 3.3 \text{ cPoise}$ , while  $\mu_0 = 16 \text{ cPoise}$ ,  $K = 1 \text{ s}$ ,  $n = 0.4$  are obtained by fitting Brooks' data [38].

## 7.4 Boundary conditions

In the present work, much emphasis was placed on modeling the geometry of the patient-specific vascular domain. However, the fluid dynamic problem is not defined without the specification of appropriate boundary conditions, which should account for the unmodeled part of the circulation. In recent works, it has been pointed out that the major determinant of blood flow and wall shear stress *patterns* is domain geometry, and that patterns rather than absolute values are required in this phase to lead to conclusions about the likely cause or preferred sites of vascular disease [110]. Nevertheless, it has also been noted [75] how inlet velocity profiles and flow waveform shapes have secondary but non-negligible effects on wall shear stress distributions. These effects have been observed to be more pronounced when the flow is characterized by higher Reynolds and Womersley numbers. We remind that Reynolds number is defined as

$$Re = \frac{\rho D U}{\mu} \quad (7.13)$$

where  $D$  and  $U$  are the characteristic spatial dimension and velocity modulus. The Reynolds number represents the ratio of inertial to viscous forces, and characterizes the laminar or turbulent flow regime. The Womersley number, defined as

$$Wo = \frac{D}{2} \sqrt{\frac{\omega}{\mu}} \quad (7.14)$$

where  $\omega$  is the characteristic pulsation of the flow, represents the ratio between transient inertial to viscous forces. In a recent work, Straatman et al. [113] shows how a finite Womersley number has a destabilizing effect on laminar to turbulent transition in internal flow at fixed Reynolds numbers.

As shown by Equation 7.3, boundary conditions can be expressed as velocity profiles or surface traction force during the cardiac cycle on the inflow and outflow sections, while the typical boundary condition employed for the wall is zero velocity. Unfortunately, detailed information about the inflow and outflow velocity distributions over the boundary is hardly available with current acquisition techniques (see below). For this reason, velocity profiles are usually deduced from centerline or section-averaged velocity measurements by means of analytic solutions for Newtonian fluids in circular-section rigid straight vessels under steady and pulsatile flow conditions. Such velocity profiles are expressed by Poiseuille equation for steady flow conditions and Womersley equation for pulsatile flow conditions [131] when either centerline velocity, pressure or flow rate is known. In order to minimize the effect of velocity distributions over inlet and outlet sections on the computed flow field, cylindrical extensions are added to the model, as shown in Chapter 4. Formaggia [41] presented a method to impose so-called *defective* boundary conditions of prescribed mean pressure and prescribed flux on the inflow or outflow of the domain while maintaining the problem well-posed, allowing to prescribe boundary conditions more closely resembling non invasively acquired data.

As a rule of thumb, velocity profiles are imposed at all the boundary sections except one, in which traction-free boundary condition is used. However, the choice of boundary conditions, if not imposed by the availability of experimental data, has to be carefully performed on a per case basis. For rigid wall models of carotid bifurcation, Perktold [89] imposed velocity on the common carotid inlet, and proposed a two-step approach for the outlets. In the first step, traction-free boundary condition was imposed on the external outlet and fully-developed flow with such to obtain prescribed flow-division was imposed on the internal outlet. In the second step, the velocity profiles obtained on the external outlet were imposed on the external outlet, while traction-free boundary conditions were imposed on the internal outlet. Conversely, Milner et al. [75] imposed prescribed velocity distribution on the inlet and on the external carotid outlet, and zero traction on the internal carotid outlet. If rigid wall assumption is relaxed and fluid-structure interaction is accounted for, as in [87], normal traction force is assumed equal to (minus) pressure acquired, for example, by applanation tonometry, while tangential traction force is set to zero, in the hypothesis that the magnitude of tangential stress is much lower than that of

pressure. This is the case of Zhao et al. [139], who imposed pressure waveform on the common carotid inlet, and fully developed mass flow on the outlets. As to simulations of the abdominal aorta, Taylor et al. [119] imposed velocity distributions at the inlet and outlets with exception of the two iliac outlets, where zero traction was imposed.

#### 7.4.1 Average boundary conditions

The most straightforward way to impose boundary conditions on the inlets and outlets of reconstructed arterial domains is to derive them from averaged physiologic conditions published in the literature. Although this approach is obviously limiting with respect to the patient-specific perspective employed in the present work, some observations have to be made. The normal *in vivo* fluctuations of hemodynamic conditions, ranging from physiologic variations in heart beat and flow rates, to acute changes due to physical activity and posture, [110] can make the input flow waveform vary significantly in morphology and magnitude. Therefore, the use of standard boundary conditions is useful even at a patient-specific level to investigate the role of geometry on flow patterns with average input conditions, since, as pointed out above, major interest is placed onto blood flow and wall shear stress patterns, rather than absolute values. A detailed study on variability of the hemodynamic results on input flow waveforms in patient-specific models constitutes an important future development. Ethier et al. [40] performed such a study on an idealized model of end-to-side anastomosis, comparing characteristic coronary and peripheral (iliac and femoral) waveforms and evidencing that markedly higher spatial and temporal shear stress gradients were produced by peripheral flow waveforms. Moore et al. [78], with an *in vitro* approach, compared the effects of simulated exercise and postprandial conditions on flow patterns inside a model of abdominal aorta, showing that increased flow rates have an effect on flow reversal during the cardiac cycle. For the purpose of modeling the physiologic waveform variability in the common carotid artery, Holdsworth et al. [48] derived an archetypal peak velocity waveform on the basis of echo-Doppler measurements on 17 normal subjects, and proposed a method to generate realistic pseudo-random sequences of cardiac cycles.

Particular care should be paid to patient-specific modeling of pathologic arterial tracts, affected by stenosis or aneurysms. In this case, physiologic averaged boundary conditions cannot be used without prior considerations on the effect of such pathologic conditions on local and global hemodynamics. As an example, we consider the case of the carotid bifurcation. It has been shown by Archie et al. [9] that stenosis grade has little influence on internal carotid blood flow rate for internal carotid stenosis grades up to 60%, due to cerebral perfusion autoregulation. This suggests that average boundary conditions could be employed as a first approximation for simulations in models which present mild stenoses. For highly stenotic models, finer considerations should be taken into account.

### 7.4.2 Patient-specific boundary conditions

The possibility of performing velocity measurements inside large arteries *in vivo* and non-invasively constitutes an important resource for the definition of patient-specific boundary conditions. As pointed out above, the acquired flow waveforms have to be interpreted as a sample of a variety of conditions which depend on physical activity and posture. Moreover, clinical examinations are typically performed in the supine position, e.g. for technological reasons, which may have an influence on the resulting measurements. However, the prescription of boundary conditions acquired from the same subjects for which geometric modeling is performed constitutes a necessary step toward the study patient-specific local hemodynamics at a clinical level, because it accounts for the inter-subject variable unmodeled part of the circulation. Coupling such acquired data to mathematical models which simulate intra-subject physiologic variability will contribute to complete such clinical framework.

We now briefly review two popular non-invasive techniques employed to acquire velocity information for the prescription of patient-specific boundary conditions.

#### Acquisition of boundary conditions by Doppler ultrasound

Thanks to the temporal resolution ( $< 15ms$ ) and the non-invasiveness of this technique, Doppler ultrasound is actually the best mean to characterize velocity waveforms in the clinical routine, also considering its low-expensiveness if compared with the other 3D imaging modalities. Moreover, flexibility in scanner geometry allows the acquisition of data with patients not necessarily in the supine position.

Doppler acquisition is usually performed by placing the sample volume in the center of the lumen of interest, visualized by B-mode, in a region in which the vessel is straight and its section is circular, in order to allow the assumption of fully developed flow. The operator adjusts the exact position of the beam such that the peak velocity in the Doppler spectrum is maximized. In order to minimize acquisition and movement artifacts a number of cycles are acquired for subsequent averaging. Data are collected as images [38, 97] or frequency signals [48] and peak blood velocity extracted as the envelope of peak frequencies of the Doppler spatial velocity spectra. For this purpose several methods have been proposed, either based on image processing or frequency signal processing [48]. R-wave identification from ECG signal or feature detection from peak velocity waveform are then used to split peak velocity waveform into cardiac cycles. Averaging the single cardiac cycle waveforms finally yields the desired peak velocity waveforms, which are then used as input for Womersley analytical solution, which yield the velocity distributions over a circular section of given diameter.

Disadvantages of Doppler ultrasound are linked to the physical limitations of the use of ultrasounds in deep vessels, particularly when the subject is overweight. Moreover, data from intracranial vessels cannot be acquired by Doppler ultrasound.

Intracranial ultrasound, which we do not describe here, can be used in these cases, which provides an approximate measure of flow rates.

### **Acquisition of boundary conditions by phase-contrast magnetic resonance**

Phase-contrast magnetic resonance angiography [85] can be employed to provide images of one or more velocity components in several time steps of the cardiac cycle. To do that, the timing of phase contrast acquisition sequences is triggered with the heart beat, so that velocity distribution for a complete cardiac cycle is reconstructed from several frames acquired from different cardiac cycles [75]. A phase-contrast MR scan produces a magnitude image, which depicts anatomy, and a phase image, which contains velocity information. After acquisition, a segmentation step is performed on the magnitude image to identify the lumen in order to derive velocity information from the phase image inside the same region. Velocity values are eventually corrected by subtracting the average signal from stationary tissue. Although in principle interpolated velocity vector distribution over the boundary sections could be imposed, the presence of artifacts and the spatial resolution of the technique (which is payed with scan time) make it more feasible to average the information over the imaged sections and to impose the resulting flow rate as the boundary condition (e.g. by imposing the corresponding Womersley velocity profile).

Even if phase-contrast MR is going to constitute a useful velocimetry tool at the clinical level, some considerations have to be made. In order to reduce artifacts, imaged sections must be located in regions of fully developed flow, since complex flow patterns arising in correspondence of bifurcations and bendings can lead to incorrect velocity measurements [111]. Misalignment between the imaging plane and the plane normal to flow direction can lead to errors in measured velocity values and distribution. Therefore, particular care must be taken to orientate the imaging plane in complex arterial tracts. The time spent for acquiring a velocity component at a section throughout the cardiac cycle is in the order of several minutes.

#### **7.4.3 Boundary conditions from multiscale modeling of circulation**

A promising approach for the imposition of boundary conditions to reconstructed arterial tracts is represented by the coupling of 3D models and lumped parameter models representing the systemic and pulmonary circulation by the electric analogy. The values of the RLC components of the lumped parameter model can be estimated, to a certain extent, from physical data of the subject (sex, age, weight).

This approach is particularly attractive because it is in principle capable to account for the effect of local pathological conditions, such as stenoses or aneurysms, or surgical procedures [61], on the whole circulatory system, providing meaningful boundary conditions for the 3D problem. For example, as noted above, the mechanisms with which internal carotid stenosis affects blood flow rate are complex,

due to the compensatory phenomena taking place at the intracranial circulation. Multiscale modeling can account for such phenomena with a proper modeling of intracranial circulation [25]. It must be pointed out that the application of lumped parameter models to subjects affected by systemic pathologic conditions, which are not localized as congenital malformations in pediatric surgery [61], requires the development of ad hoc models and parameter estimation methods.

## 7.5 Finite-element approximation of Navier-Stokes equations

Once the problem has been defined with appropriate rheologic parameters and boundary conditions, the Navier-Stokes equations must be solved numerically. Several methods for the approximation of Navier-Stokes equations exist, the most popular being finite volumes (FV) [84] and finite elements (FE) [94]. Once again, proper treatment of numerical approximation of Navier-Stokes equations is beyond the scope of this work. We will only give a brief sketch of the method we employed in this work, the finite element method.

### 7.5.1 Weak formulation of Navier-Stokes equations

The finite element method is based on Galerkin approximation method, for which the differential problem is turned into its weak (variational) form. The standard procedure is to multiply the partial differential equation terms by arbitrary test functions  $\mathbf{v}$  and  $q$  defined in proper test spaces  $V$  and  $Q$ , integrate the equation over the whole domain and apply the Green-Gauss theorem to shift one order of differentiation from the unknown function to the test function in second-order terms (i.e. momentum diffusion term in Navier-Stokes equation). Boundary conditions are expressed through the choice of the test spaces for Dirichlet conditions (velocity profiles), and as additional terms of the integral equation for Neumann conditions (normal stress). The fluid-dynamics problem, in its weak formulation, translates into *look for the functions  $\mathbf{u} \in V$  and  $q \in Q$  such that the integral equations*

$$\begin{cases} b(\mathbf{u}, q) = 0 \\ a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) = (f, \mathbf{v}) \end{cases} \quad (7.15)$$

*are satisfied*  $\forall \mathbf{v} \in V$  and  $\forall q \in Q$ , where  $a(., .)$  and  $b(., .)$  are proper integral expressions. This formulation, although equivalent to the original, contains less regularity requirements for the solution, since the order of differentiation is lower than in the differential formulation. Further details on this topic can be found in [94].

### 7.5.2 Spatial discretization

Once the problem has been formulated in its integral form (Equation 7.15), it is still an infinite dimensional problem, the continuum being represented not by  $\mathbb{R}^3$ ,

as in the strong formulation, but rather by functional spaces  $V$  and  $Q$ . Galerkin method consists in reducing the dimensionality of the problem from infinite to finite by looking for an approximate solution  $\mathbf{u}_h, p_h$  of the weak problem in finite functional subspaces  $V_h, Q_h$  of the original spaces  $V, Q$ . Since  $V_h$  and  $Q_h$  are finite, every function defined in  $V_h$  or  $Q_h$  can be expressed as a linear combination of finite numbers  $N_h$  and  $M_h$  of basis functions  $\varphi_j(x)$  and  $\psi_k(x)$

$$\begin{aligned}\mathbf{u}_h(x) &= \sum_{j=1}^{N_h} \mathbf{u}_j \varphi_j(x) \\ p_h(x) &= \sum_{k=1}^{M_h} p_k \psi_k(x)\end{aligned}\quad (7.16)$$

where the coefficients  $\mathbf{u}_j$  and  $p_k$  are the unknowns of the discrete problem, the *nodal values* of the solution. Note that now only basis functions depend on the spatial coordinate. Therefore, by substituting the unknown functions with their discretized versions into the integral Navier-Stokes equations yields

$$\begin{cases} b(\mathbf{u}_h, q_h) = 0 \\ a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, p_h) = (f, \mathbf{v}_h) \end{cases}\quad (7.17)$$

$\forall \mathbf{v}_h \in V_h$  and  $\forall q_h \in Q_h$ . Since it is possible to perform exact integration of the integral terms which now contain the basis functions only<sup>1</sup>, Equation 7.17 is an algebraic nonlinear system in the  $N_h + M_h$  unknowns  $\mathbf{u}_j$  and  $q_k$ . An important property of Galerkin method is that errors between the continuous and the discretized solutions are orthogonal to the discrete subspaces. Therefore, for a given set of subspaces, Galerkin method yields the best approximation of the continuous solutions<sup>2</sup>.

The finite element method is finally derived by defining the finite functional subspaces into which  $\mathbf{u}_h$  and  $p_h$  are looked for as the space of continuous piecewise polynomial functions (the degree of polynomials determines the order of accuracy of the method). Therefore, in the finite element method the physical domain has to be discretized into elements whose nodes host the nodal values  $\mathbf{u}_j$  and  $q_k$ , and whose interpolation functions are defined by basis functions  $\varphi_j(x)$  and  $\psi_k(x)$ . It is important to note that the subspaces  $V_h$  and  $Q_h$  must be chosen in a compatible way (Babuska-Brezzi condition [94]), in order to avoid that a  $q_h$  exists such that for every  $\mathbf{v}_h$ ,  $b(\mathbf{v}_h, q_h) = 0$ , thus introducing spurious pressure modes which are transparent to the discretization employed. As an example, tetrahedral elements

<sup>1</sup>Generalized Galerkin methods employ approximate integration (e.g. Gauss integration) of the integrals involving basis functions, yielding a problem in which discretized integral expressions  $a_h(,)$  and  $b_h(,)$  appear.

<sup>2</sup>Several techniques exist to improve convergence of Galerkin approximation, for example in regions where convection dominates over diffusion. In this case formulations which properly introduce numerical diffusion in an upwinding fashion are employed. We refer to the literature [94] for further details.

with linear basis functions for both velocity and pressure do not satisfy the above requirement<sup>3</sup>.

In Chapter 6 two methods for discretization of the domain of vascular tracts models were presented. As to 8-node hexahedral elements, employing linear basis functions for velocity and constant basis functions for pressure satisfies the Babuska-Brezzi condition. We then proposed an hybrid meshing technique which generated 10-node tetrahedral elements and 18-node wedge elements. In this case, employing quadratic basis functions for velocity and linear basis functions for pressure satisfies the Babuska-Brezzi condition.

### 7.5.3 Solution technique

The nonlinear algebraic system of equations in the  $N_h + M_h$  unknowns  $\mathbf{v}_h$  and  $q_h$  resulting from finite-element discretization (Equation 7.17) is finally solved to produce the desired solutions. A number of solution strategies have been developed in the literature. For this purpose we use the commercial solver FIDAP [52] with a pressure-projection segregated solution strategy. The segregated strategy consists in decomposing the system matrix in sub-matrices associated with one conservation equation, whose solution is performed sequentially. The main difficulty in segregated strategy for Navier-Stokes problem is the treatment of pressure, since it doesn't appear in both conservation equations. In the pressure projection scheme, for each iteration, pressure approximation is first derived from the momentum equation using the velocity field available from the previous iteration, then velocity components are computed from the momentum equation using the available field variables, and finally the computed velocity field is adjusted in such a way that mass conservation equation (incompressibility constraint) is satisfied.

Finally a scheme for time integration is due to complete the Navier-Stokes approximation framework. Implicit time integration schemes, in which the update is performed considering the unknowns at the new time step, such as first-order backward Euler and second-order trapezoid, guarantee stability no matter the time step size. Even if computationally more expensive, they are in general preferred to explicit integration schemes which, although simple and fast, are only conditionally stable, imposing a limit on time step size. Furthermore, time step in implicit schemes can be adaptively set on the basis of local time truncation error.

## 7.6 Post processing

Once CFD computation has been carried out, velocity and pressure fields are available at the  $N_h$  and  $M_k$  nodes, respectively. Apart from direct visualization of such fields by means of vector and contour plots, post processing techniques are needed

---

<sup>3</sup>Corrections to the classic finite element formulation allow to circumvent this problem employing a least squares stabilization technique [53].

for both more effective velocity field visualization or computation of wall shear stress (WSS) distribution or related quantities.

### 7.6.1 Wall shear stress and derived quantities

As pointed out in the Introduction, WSS and derived quantities which account for reversal and temporal and spatial gradients are involved in endothelial cell regulation and atherogenic mechanisms. Since solution to Navier-Stokes equations provides velocity and pressure fields, WSS distribution must be computed in a post-processing step.

Computation of stress is first performed by evaluating the components of the stress tensor, using the relation  $\sigma_{ij} = -p\delta_{ij} + \mu(u_{i,j} + u_{j,i})$ . Wall traction vector is then extracted from the stress tensor by product with the surface normal,  $\sigma_i = \sigma_{ij}n_j$ . WSS is finally computed extracting the tangential component of the traction vector  $\tau_i = \sigma_i - (\sigma_j n_j)n_i$ .

It has been observed in endothelial cell cultures exposed to laminar flow, that endothelial cells tend to align with the mean direction of flow, given by the direction of

$$\bar{\tau}_i = \frac{1}{T} \int_0^T \tau_i dt \quad (7.18)$$

Even if WSS magnitude values and their average over the cardiac cycle can characterize normal and atherosclerosis-prone regions [67], the observation that endothelial cells respond to spatial and temporal variations of WSS has led to the formulation of WSS-derived indicators of disturbed flow.

One of the first indexes which accounted for disturbed flow conditions was the one proposed by Ku et al. [58], the oscillatory shear index (OSI), as a measure of how much WSS changes sign with respect to the mean direction of flow over the cardiac cycle. We here report the formulation given by Taylor et al. [119]

$$OSI = \frac{1}{2} \left( 1 - \frac{\bar{\tau}}{\tilde{\tau}} \right) \quad (7.19)$$

where

$$\bar{\tau} = |\bar{\tau}_i| \quad (7.20)$$

and

$$\tilde{\tau} = \frac{1}{T} \int_0^T |\tau_i| dt \quad (7.21)$$

OSI can vary from 0, when WSS is always directed along its average direction, to 0.5, which indicates that WSS direction is such that the vector sum over the cardiac cycle is zero. OSI obviously accounts only for temporal WSS variation. Even if Hyun et al. [51] demonstrated that OSI is dependent of the type of input flow waveform, it is still a very commonly computed parameter [75], thanks to its simplicity and to the results obtained by Ku et al. [58], which found positive correlation at the carotid bifurcation between OSI, calculated from an averaged

experimental model, and vascular wall thickness measured on cadaver specimens. A typical site at which OSI is significantly different from zero is the carotid bulb, one of the preferred sites for atherogenic processes.

In order to account for spatial variations of WSS, Lei et al. [63] proposed wall shear stress gradient (WSSG) magnitude averaged over the cardiac cycle

$$\overline{|\text{WSSG}|} = \frac{1}{T} \int_0^T |\text{WSSG}| dt \quad (7.22)$$

where

$$|\text{WSSG}| = \left[ \left( \frac{\partial \tau_m}{\partial m} \right)^2 + \left( \frac{\partial \tau_n}{\partial n} \right)^2 \right]^{\frac{1}{2}} \quad (7.23)$$

where  $m$  is the mean direction of flow, that is the direction of  $\bar{\tau}_i$ , and  $n$  is the direction normal to  $m$  over the surface.

Another WSS-derived indicator of disturbed flow is WSS angle deviation (WS-SAD), proposed by Hyun et al. [51], defined as

$$\text{WSSAD} = C \arccos \left( \frac{\bar{\tau}_i \bar{\tau}_i^{neigh}}{|\bar{\tau}_i| |\bar{\tau}_i^{neigh}|} \right) \quad (7.24)$$

where

$$C = \begin{cases} 1 & \text{if } \bar{u}n_i \geq 0 \\ 0 & \text{if } \bar{u}n_i < 0 \end{cases} \quad (7.25)$$

where  $\bar{u}$  is near-wall velocity vector averaged over the cardiac cycle, and  $\tau^{neigh}$  is wall shear stress in a neighbor cell (we remind that the index summation notation is being used). Final WSSAD is the mean of the results from neighbor cells. WSSAD is an attempt to account for both wall shear stress spatial variation and particle deposition phenomena, which depend on the near wall velocity field. Hyun et al. [51] showed good agreement between WSSAD distribution and high particle deposition regions. In 3D WSSAD ranges from 0 to  $\pi$ , greater values being associated with more particle deposition-prone near-wall flow conditions.

We implemented OSI computation by exporting wall stress data from FIDAP post processor and importing them into C++ code using VTK libraries. Integration was performed with extended Simpson's quadrature formula [91]

$$\int_0^T f(t) dt = \Delta t \left[ \frac{1}{3} f_0 + \frac{4}{3} f_1 + \frac{2}{3} f_2 + \frac{4}{3} f_3 + \frac{2}{3} f_4 + \dots + \frac{4}{3} f_{T-2} + \frac{2}{3} f_{T-1} + \frac{1}{3} f_T \right] \quad (7.26)$$

Preliminary implementations of WSSG and WSSAD computation have been made, although they will not be presented here.

### 7.6.2 Streaklines and Lagrangian particle tracking

Visualization of periodic flow inside 3D domains by means of vector field plots, although it provides a detailed representation of primary and secondary flow patterns,

requires the definition of cut planes over which vector visualization is performed. Patient-specific models of arterial tracts are rarely planar, so that such planes must be defined interactively in several locations. Moreover, vector visualization may not be adequately understood by physicians, who are used to observe contrast filling and wash-out in traditional X-ray angiography, also considered that transient effects require several frames to be displayed.

Such problems can find a simple and effective solution in Lagrangian techniques. Massless, neutrally buoyant, non-diffusive particles are seeded at the model inlet, and their trajectory computed in the pulsatile flow environment by solving the ordinary differential equations [109]

$$\frac{dx_i}{dt} = u_i(x_i, t) \quad (7.27)$$

with initial condition  $x_i(t_0) = x_i^0$ , where  $u_i$  is the computed velocity field. Equation 7.27 can be solved by means of standard ordinary differential equation integration methods, such as fourth-order Runge-Kutta method [91]

$$\begin{aligned} y_{n+1} &= y_n + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + O(h^5) \\ k_1 &= h f(x_n, y_n) \\ k_2 &= h f\left(x_n + \frac{h}{2}, y_n + \frac{k_1}{2}\right) \\ k_3 &= h f\left(x_n + \frac{h}{2}, y_n + \frac{k_2}{2}\right) \\ k_4 &= h f(x_n + h, y_n + k_3) \end{aligned} \quad (7.28)$$

which requires four function evaluations. Integration step is expressed as a fraction of characteristic length of the cell of the underlying domain mesh. Stepsize can be also made adaptive as in the embedded fourth-order Runge-Kutta method [91], where six function evaluations are used to build fourth and fifth order approximations of  $y(x+h)$ , so that their difference provides an estimate of the truncation error, and the step size is adjusted in order to keep the error under a desired threshold.

Although FIDAP post processor has Lagrangian particle tracking capabilities, performing such computation in pulsatile CFD analyses requires to queue the result of several cardiac cycles, which is impractical for the amount of disk space required [52]. For this reason, and in order to have the possibility to provide a basis for extending the computation to more advanced Lagrangian techniques, we implemented Lagrangian particle tracking in C++ using VTK libraries [105]. Class `vtkStreamer` is available in VTK distribution, but it only supports streamline generation (up to version 4.1), which are equivalent to Lagrangian particle trajectories computed by integrating Equation 7.27 in a steady-state velocity field<sup>4</sup>. Therefore

---

<sup>4</sup>We will here refer to *streamlines* as to Lagrangian particle trajectories computed from Equation 7.27 in which the velocity field is time-dependent.

we extended streamline computation to time-varying datasets, taking advantage from the highly modular implementation of VTK, which allowed to reuse many of the components employed in streamline computation. For example, we used Runge-Kutta integration methods provided by classes `vtkRungeKutta4` (fourth order) and `vtkRungeKutta45` (embedded fourth order with adaptive stepsize). Nodal velocity vectors of a complete cardiac cycle are first imported from FIDAP solution (usually simulation is run for three cardiac cycles and the last cycle is exported, in order to minimize transient effects).

Seed points are specified on the inlet sections by uniform random distribution. A starting time is associated to each seed point, so to seed new streaklines during the cardiac cycle. Although, in order to simulate uniform concentration of particles, it would be necessary to account for temporal input waveform and cross-sectional velocity distribution [59], we chose to seed particles uniformly in space and time, regardless input boundary conditions, as in [109]. Pulsatility is accounted for in that velocity field is treated as a periodic function during integration. Time is stored for each generated streakline point. Streaklines are generated as long as their points remain inside the domain or the speed goes to zero.

Once streaklines are computed, and eventually stored, Lagrangian particle animation can be generated frame by frame by extracting those points along streaklines whose time corresponds to the frame time. This way, by seeding Lagrangian particles for a given time, e.g. three cardiac cycles, and following them until they exit the domain, it is possible to effectively visualize time-dependent flow patterns, particle wash-out and high particle residence regions.

An extension of Lagrangian particle visualization was proposed by Steinman [109] and also implemented in this work. This technique simulates the presence of a shutter with finite speed for the acquisition of frames, so that each frame contains particle traces formed by the portion of streaklines whose time fall between frame time and frame time plus the shutter speed. As a result, length and orientation of particle traces provide effective visual feedback on particle speed and trajectory orientation, thus effectively evidencing flow features. Also this visualization technique was implemented.

## 7.7 Applications

We will now present two cases for which local hemodynamics characterization has been performed, a model of carotid bifurcation acquired by contrast-enhanced CT angiography, and a model of abdominal aorta acquired by contrast-enhanced MR angiography.

### 7.7.1 Carotid bifurcation acquired by CT angiography

A carotid bifurcation model was reconstructed from a contrast-enhanced CT acquisition performed on a 74 years old male patient in the Radiology Unit of Ospedale

S.Paolo, Milano, with a Lightspeed Multidetector CT (General Electrics, Fairfield, CT, USA). Imaging parameters were set as in the clinical routine (0.3085 mm in-plane resolution, 1.25 mm slice thickness). Reconstruction, editing and meshing were performed as described in the previous chapters, and the resulting computational mesh contained 55687 elements. Blood was modeled as Newtonian (viscosity  $\mu = 3.5$  cpoise, density  $\rho = 1.045 \text{ g/cm}^3$ ). Vessel wall was assumed rigid. Boundary conditions were imposed as time-varying parabolic velocity profiles on the common inlet and on the external outlet, while zero traction was imposed on the internal carotid outlet. Flow waveforms were taken from [87], and mean flow rates were set to 7 ml/s for the common inlet (0.717 cm diameter) and 2.1 ml/s for the external outlet (0.355 cm diameter), leading to 70%-30% flow division between the internal and the external branches. Flow rate waveforms are represented in Figure 7.1. Heartbeat frequency was set to 65.5, resulting in 0.916 s inter-beat time. Implicit Euler time integration was employed, with fixed time increments of

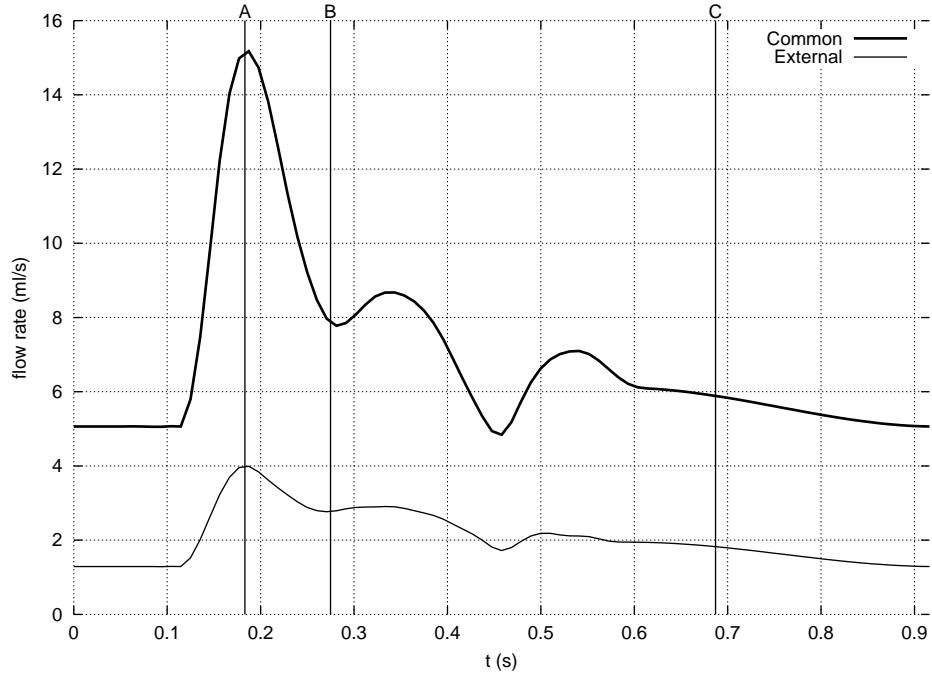


Figure 7.1: Flow rate waveform employed as boundary conditions for the CFD analysis of the carotid bifurcation. A, B and C represent peak systole, late systole and mid diastole respectively.

0.02 s. Computation for two initial cardiac cycles was performed in order to damp transient effects, and the third was retained as the solution. The velocity vector field and the wall viscous stress vector distribution were exported from Fidap and imported into our code for further processing.

Figure 7.2 shows velocity profiles computed at several planes for the carotid

bifurcation model at three representative time steps. In agreement with previous results [87, 112], the velocity profiles are skewed toward the bifurcation apex, while low-velocity regions are identified near the lateral walls past the bifurcation, in particular inside the carotid bulb. This effect is more pronounced at late systole.

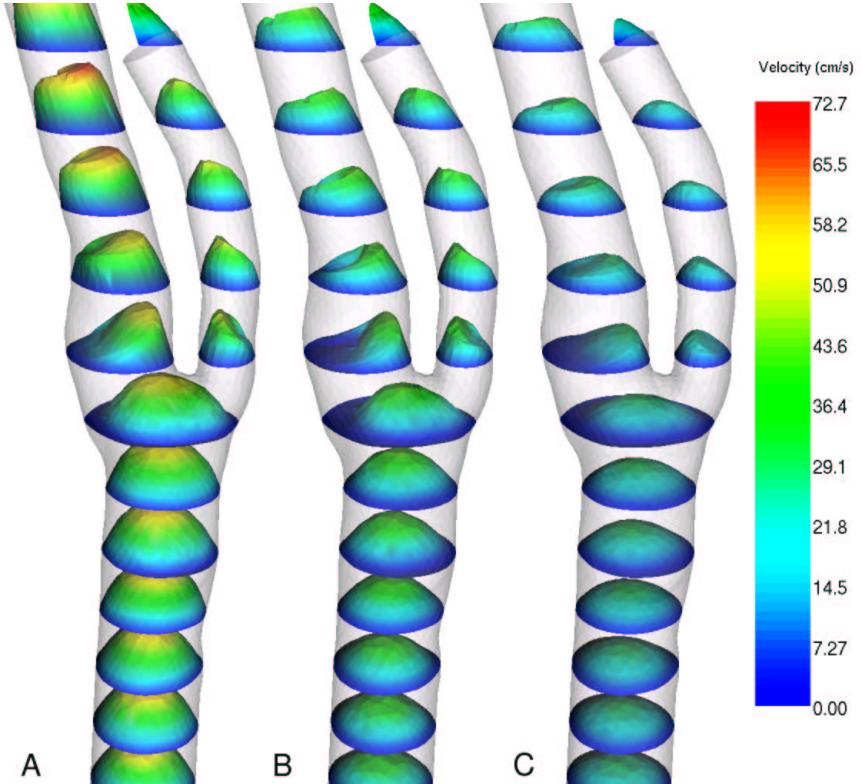


Figure 7.2: Velocity profiles at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.1) for the carotid bifurcation model.

Three frames at the time steps A, B, C of Lagrangian particle traces are shown in Figure 7.3. The view from above the internal outlet evidences the presence of secondary helical flow patterns, and the frontal view confirms the presence of a region of low-velocity disturbed flow at the carotid bulb.

Figure 7.4 shows the WSS distribution at time steps A, B, C. The scalar range has been chosen to evidence that the carotid bulb remains a low WSS region at all the three time steps, so as the bulging on the common carotid wall below the bifurcation.

Mean WSS (over the cardiac cycle) and OSI distributions were computed, as shown in Figure 7.5. Again, the scalar range for mean WSS has been chosen to evidence features of interest, such as the low WSS region from the bifurcation bulb around the bifurcation region, and at the common carotid bulging. Accordingly to the literature [58, 112], OSI is significantly non-zero at the carotid bulb. The non-

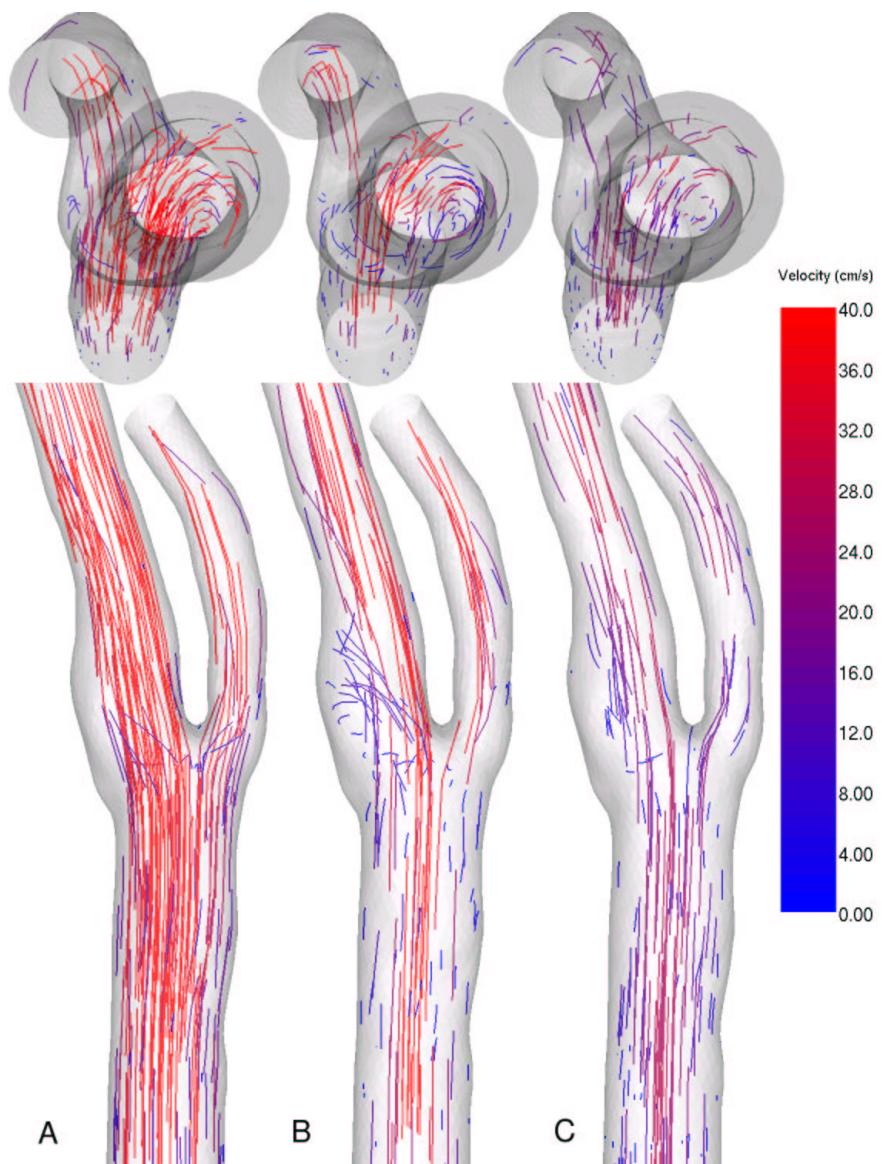


Figure 7.3: Lagrangian particle traces at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.1) for the carotid bifurcation model (above: view from above the internal carotid outlet).

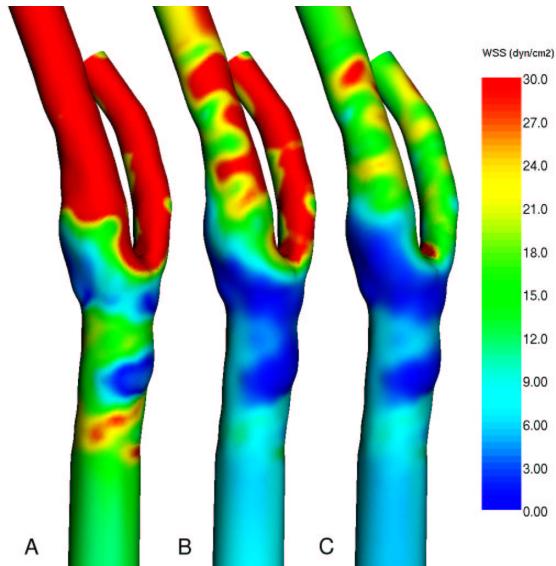


Figure 7.4: Wall shear stress distribution at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.1) for the carotid bifurcation model.

zero region extends from the bulb toward the external side, and non-zero values are also computed at the bulging on the common carotid wall. Note how the low mean WSS regions (in blue in Figure 7.5) co-localize with non-zero OSI regions, thus identifying *low and oscillating wall shear stress* (likely) athero-prone sites [67].

We finally identified three lines on the carotid bifurcation wall along which mean WSS and OSI were plotted, as shown in Figure 7.6. The plots confirm the presence of a very low mean WSS and high OSI region at the bifurcation bulb (line 2).

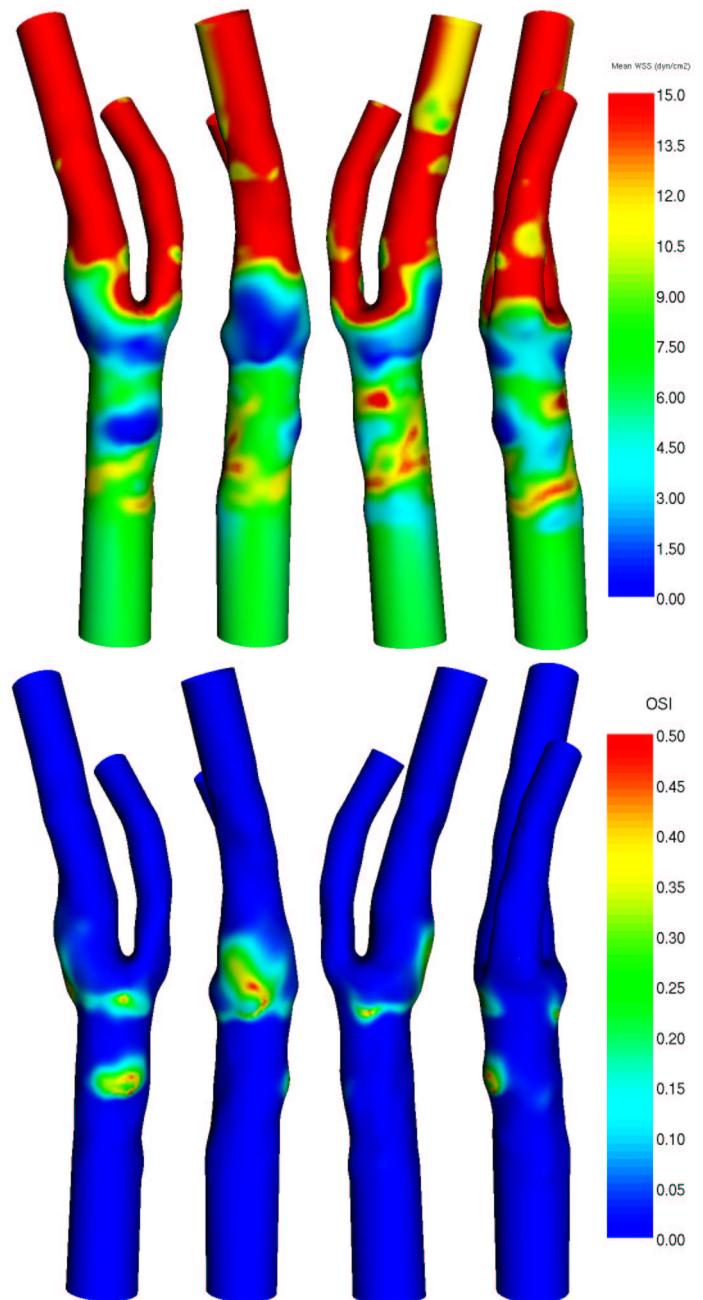


Figure 7.5: Mean wall shear stress (above) and OSI (below) distributions for the carotid bifurcation model, depicted at four orthogonal views. Note that the scalar range of mean WSS is not the same as that of Figure 7.4

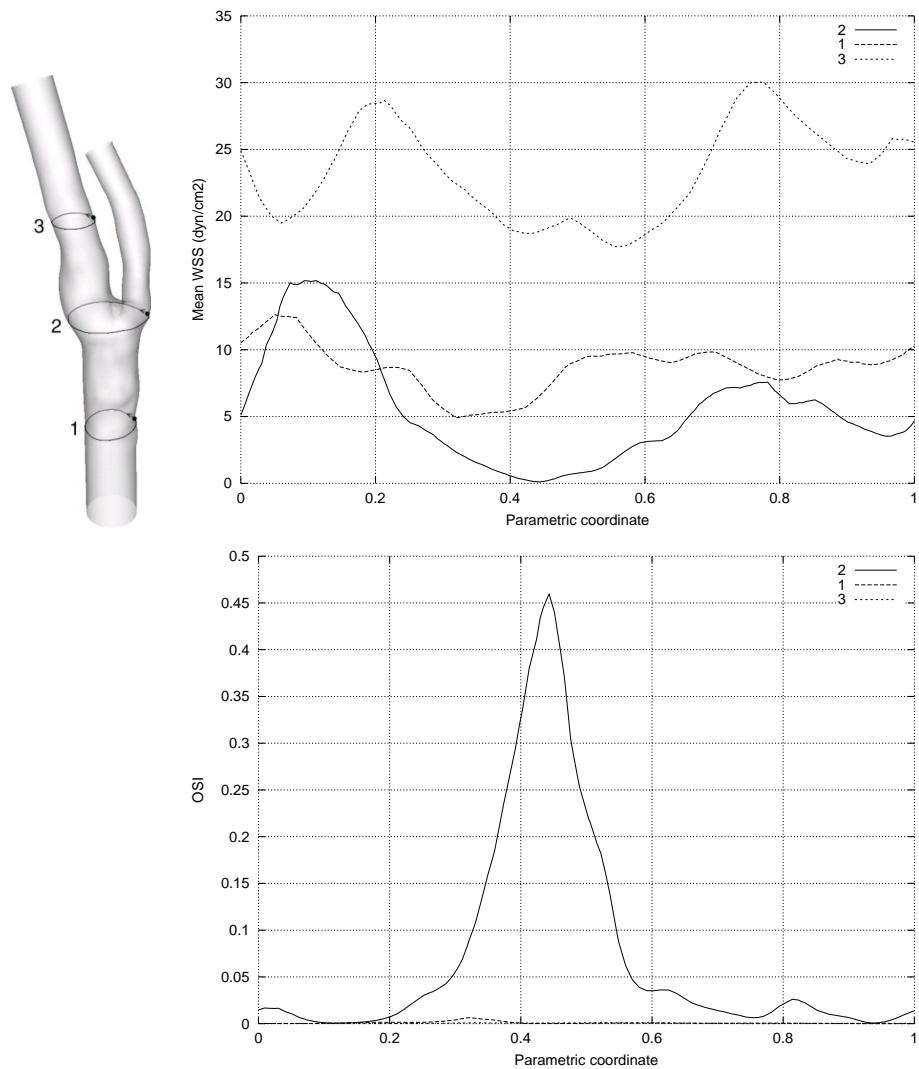


Figure 7.6: Right: definition of three lines of interest on the carotid bifurcation surface (1: common; 2: bifurcation; 3: internal). Right: plot of mean WSS (above) and OSI (below) along lines shown on the left.

### 7.7.2 Abdominal aorta acquired by MR angiography

An abdominal aorta model was reconstructed from a Gd-DTPA enhanced MR acquisition performed on a 63 years old male patient in the Radiology Unit of Ospedale S.Paolo, Milano, with a Gyroscan NT 1.5T (Philips). Imaging parameters were set as in the clinical routine (coronal orientation, 0.878 mm in-plane resolution, 3.0 mm slice thickness). Reconstruction, editing and meshing were performed as described in the previous chapters, and the resulting computational mesh contained 155214 quadratic elements (304507 nodes). Blood was modeled as Newtonian (viscosity  $\mu = 3.5$  cpoise, density  $\rho = 1.045 \text{ g/cm}^3$ ). Vessel wall was assumed rigid. Boundary conditions were imposed as time-varying parabolic velocity profiles on the aorta inlet and on the celiac, superior mesenteric, right and left renal outlets, on the basis of the mean blood flow values given in [78] (reported in Table 7.1), while zero traction was imposed on the aorta outlet. Flow waveform for the aorta inlet was taken from [119] (shown in Figure 7.7). The flow waveforms for celiac, superior mesenteric, right and left renal were obtained by appropriately scaling the aorta inlet flow waveform (this was justified by taking into account the flow waveforms shown in [119]). Heartbeat frequency was set to 68, resulting in 0.88 s inter-beat time. Computation was performed with the same modalities

Artery	Diameter (cm)	Blood flow rate (ml/s)
Aorta inlet	1.86	36.50
Celiac	0.73	9.83
Superior mesenteric	0.60	6.67
Right renal	0.55	6.67
Left renal	0.42	4.44

Table 7.1: Mean flow rate boundary conditions imposed as velocity profiles on the abdominal aorta model.

employed for the carotid bifurcation presented above.

Figure 7.8 shows the velocity profiles computed at several planes for the abdominal aorta model at three representative time steps. Due to the lateral bending of the abdominal aorta between the celiac artery and the renal bifurcations, velocity profiles are skewed on the left. Together with the elliptical shape presented by the aorta in the same region, this produces a recirculation on the left side between the celiac and the superior mesenteric artery. This effect is more pronounced at late systole (B).

Three frames at the time steps A, B, C of Lagrangian particle traces are shown in Figure 7.9. The images evidence how the lateral regions at the level of the celiac artery and below host low-velocity complex flow regions, so as the infrarenal region on the internal side of the bending.

Figure 7.10 shows the WSS distribution on the abdominal aorta surface at time steps A, B, C. The distribution evidences the complex flow patterns developing

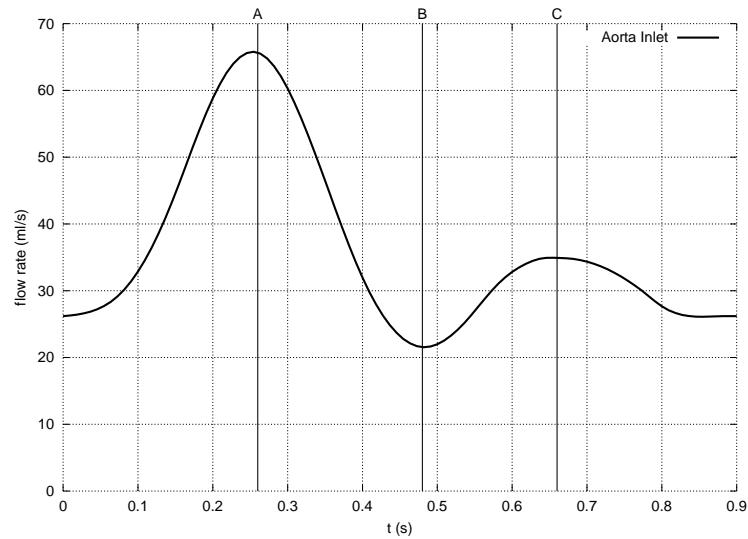


Figure 7.7: Flow rate waveform employed as boundary condition on the aorta inlet.  
A, B and C represent peak systole, late systole and mid diastole respectively.

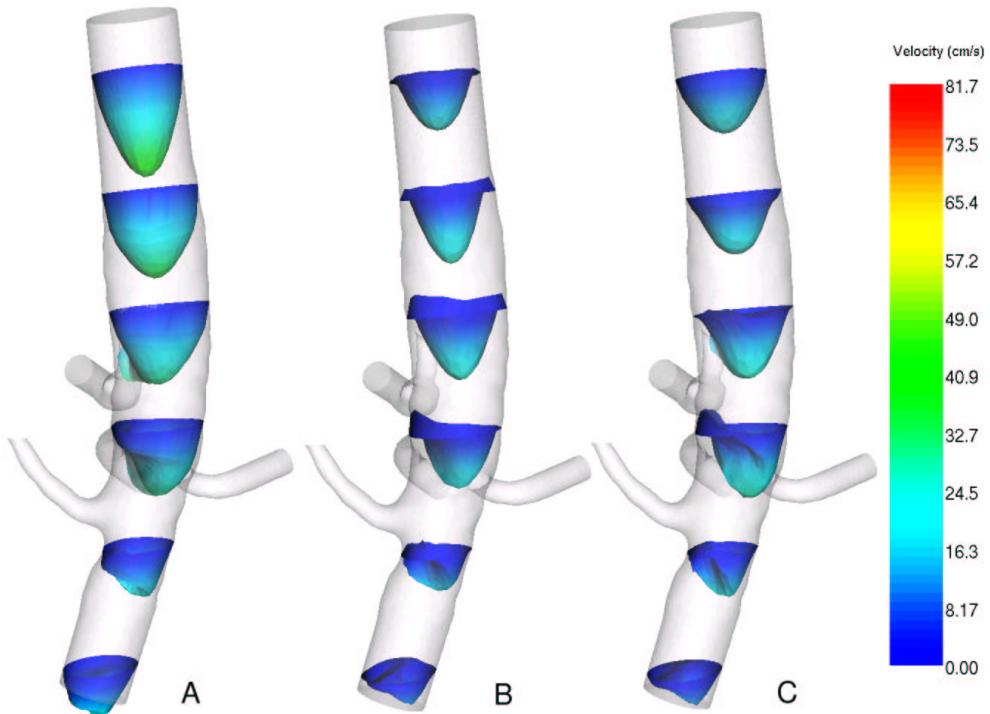


Figure 7.8: Velocity profiles at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.7) for the abdominal aorta model.

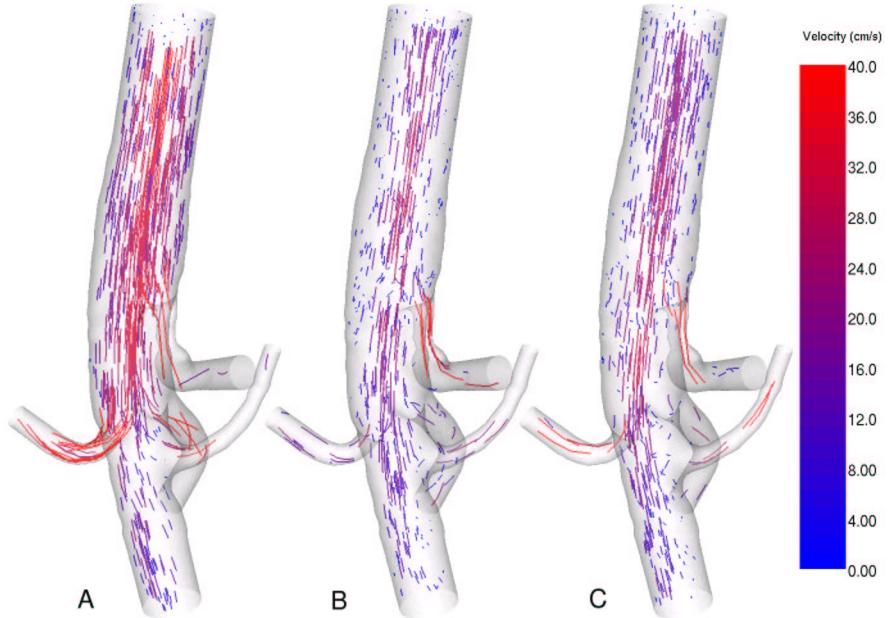


Figure 7.9: Lagrangian particle traces at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.7) for the abdominal aorta model.

around the bifurcations. In particular, high WSS spatial gradients are present on the distal side of the bifurcations, where flow divides.

Figure 7.11 shows mean WSS (over the cardiac cycle) and OSI distributions on the abdominal aorta surface. Low mean WSS regions are observed in the suprarenal aorta on the dorsal side, opposite to the celiac and superior mesenteric origins, and in the infrarenal ventral side. Particular low values are observed at the left side between the celiac and the superior mesenteric artery, where the aorta bends to the left and presents a pronounced elliptical section. OSI distribution presents a high-values in the same region, indicating complex flow and recirculation. Even if model geometry substantially differs from that presented in [119], there is an overall agreement in OSI distribution. In that work, in that in the anterior wall low OSI values were observed above the celiac origin, and higher values in the infrarenal tract, while on the posterior wall high OSI values were observed in the suprarenal and infrarenal tracts, and low values at the level of the renal bifurcation.

We finally identified five lines on the abdominal aorta wall along which mean WSS and OSI were plotted, as shown in Figure 7.12. Of particular interest are line 2, which presents very low mean WSS values between 0.5 and 0.7 (parametric coordinate), and line 3, which presents very low mean WSS values between 0.4 and 0.6. In both cases, OSI is above 0.4, thus confirming the presence of a region of low and oscillating WSS.

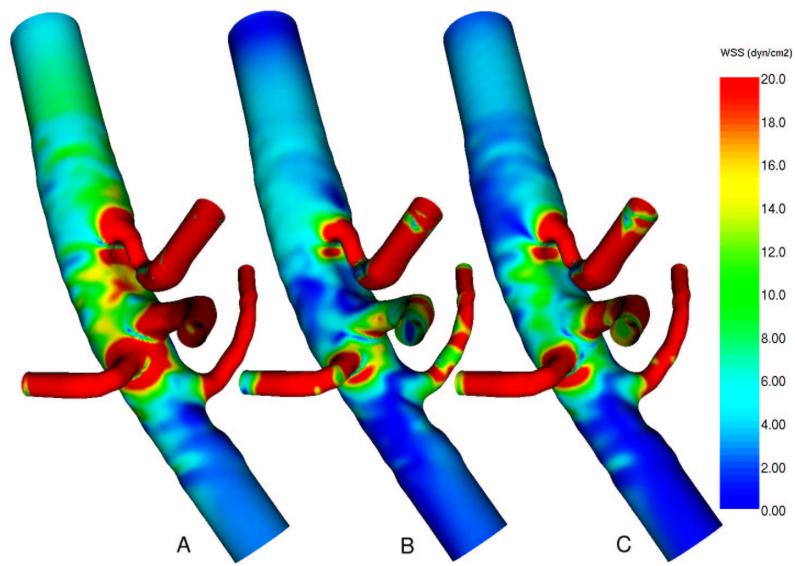


Figure 7.10: Wall shear stress distribution at (A) peak systole, (B) late systole and (C) mid diastole (refer to Figure 7.7) for the abdominal model.

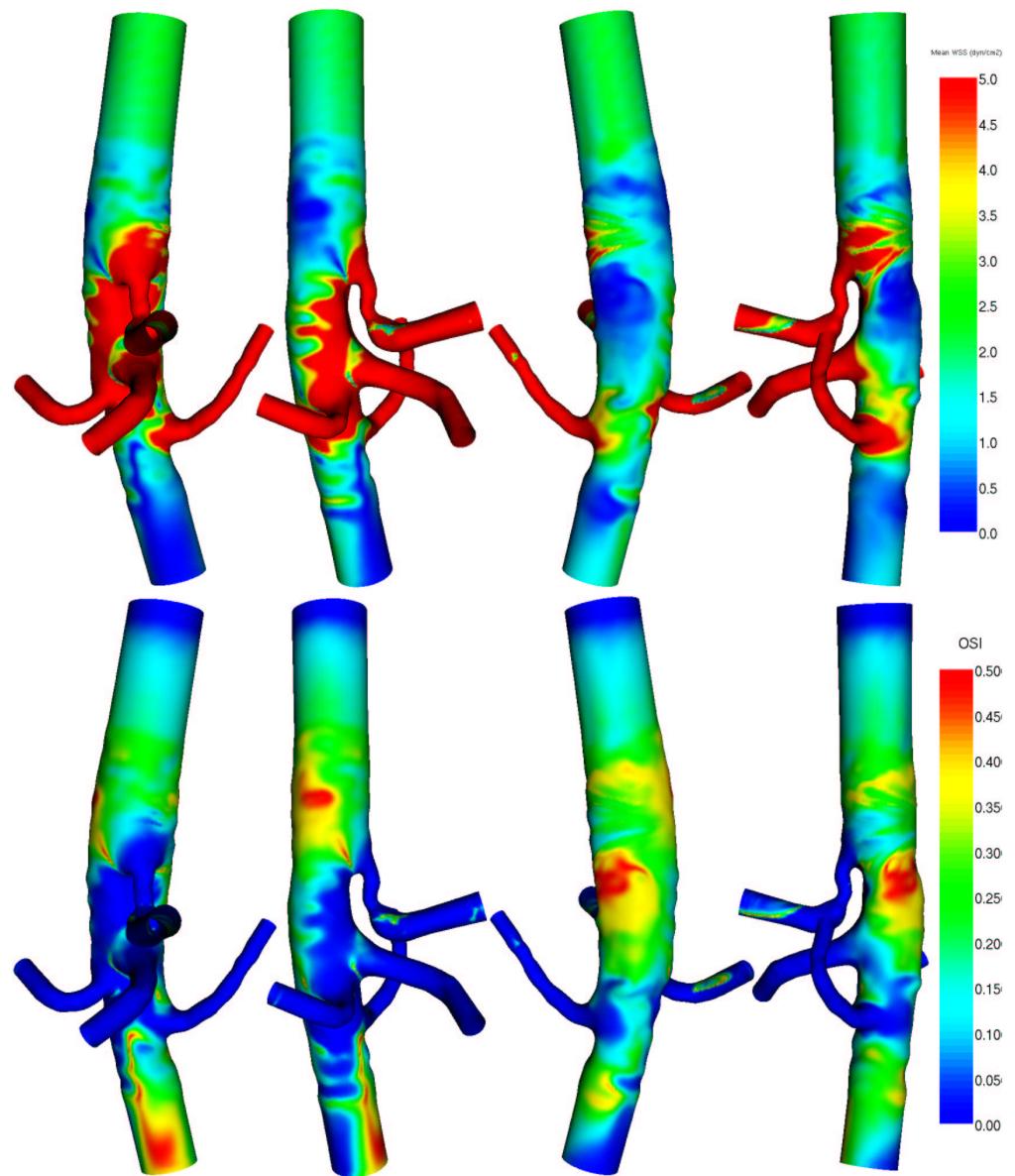


Figure 7.11: Mean wall shear stress (above) and OSI (below) distributions for the abdominal aorta model, depicted at four orthogonal views. Note that the scalar range of mean WSS is not the same as that of Figure 7.10

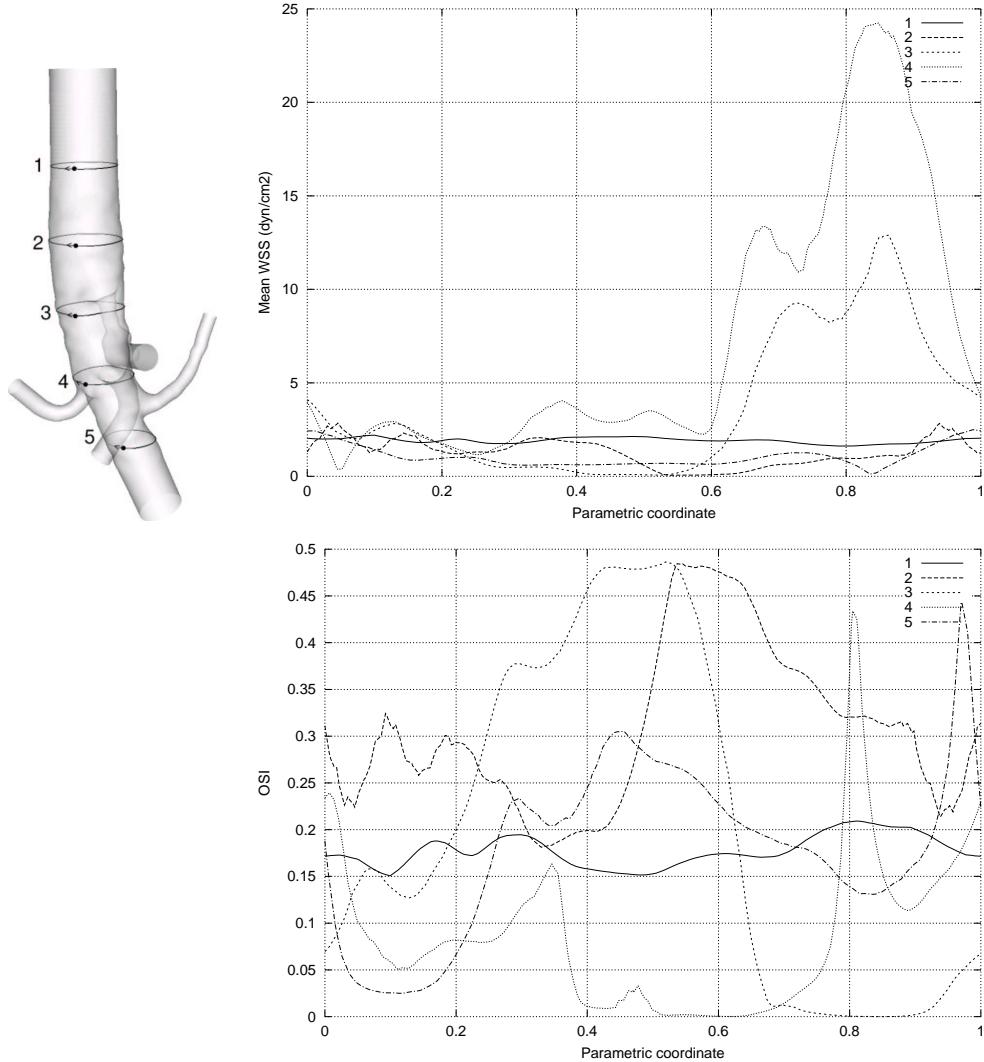


Figure 7.12: Left: definition of five lines of interest on the abdominal aorta surface (1: inlet; 2: supraceliac; 3: infraceliac; 4: suprarenal; 5: infrarenal). Right: plot of mean WSS (above) and OSI (below) along lines shown on the left.



# Chapter 8

## Conclusions

### 8.1 Summary of contributions

The aim of this work was to set up a framework for patient-specific reconstruction, geometric analysis and local hemodynamics characterization of large arteries. This has been accomplished with the development of a set of techniques which allow to reconstruct, edit, analyze and discretize 3D models of vasculature from 3D CT and MR angiography images requiring only high-level interaction, and successively to post-process the results of CFD analysis. Manual, time-consuming and operator-dependent tasks are completely avoided. Particular care was taken in the definition of variational criteria as the basis on which the presented techniques were developed, thus ensuring accuracy and reproducibility.

In Chapter 2, level sets has been used as segmentation and 3D modeling technique. Level sets had already been employed in the field of blood vessel modeling [35, 124, 123, 127], proving their effectiveness. The main advantages are the natural handling of topological changes, great deformations, the flexibility in the implementation of image-based terms, and the existence of numerical schemes (taken from hyperbolic partial differential equation solution techniques) which guarantee accuracy and stability. We implemented level sets with the approach given by Whitaker [130], which allows efficiency and sub-voxel accuracy. Our original contribution in this chapter resides mainly in single-vessel initialization and reconstruction approach, which allows to tailor level sets evolution parameters to the scale and features of each branch. Even though apparently more interaction is needed, reconstruction is made easier and faster, also for a numerical point of view. In fact, the time steps chosen by the CFL condition are somehow optimized for image features of each vessel branch. The application of the reconstruction techniques presented in this chapter have been presented in two our works [4, 6].

In Chapter 3, a simple, fast topological analysis and centerline approximation algorithm for interconnecting tubular surfaces has been presented. The algorithm, originally developed by us and employed for the analysis of complex 3D microvascular networks [7], has been employed for fast interactive initialization of level sets

in Chapter 2, and for automatic model decomposition for mesh generation in Chapter 6. The algorithm is completely new.

In Chapter 4, two techniques for editing the surface of reconstructed models were presented. Both approaches were introduced in a variational context. Both were presented in two our works [4, 6]. The first, based on a modification of Laplacian mesh filtering, is fast and simple, and yields good results although no guarantee of high-order continuity on the border of the edited region are given. The second, inspired by the work of Meyer et al. [74] and Schneider et al. [104], is based on the resolution of bi-Laplace-Beltrami fourth-order partial differential equation on the surface mesh, and provides second-order continuity on the boundary of the edited region. Moreover, in this latter approach the desired surface corresponds to the minimum of the energy functional which drives surface evolution. Besides introducing, tailoring and implementing such techniques in vascular modeling, our contributions reside in the development of weighting functions in weighted Laplacian filtering, the development of an original term for surface position update in bi-Laplace-Beltrami problem factorization (Equation 4.26), and in the choice of the solution scheme and relaxation strategy for the factorized problem. Surface editing techniques were employed for local surface fairing and for the creation of cylindrical extensions at the outer sections of the models (necessary for CFD analysis).

In Chapter 5, an original approach for centerline computation and geometric characterization of model surfaces was given. Centerlines have been defined with a variational principle, as in [36]. First an approximation of the medial axis of the models, the embedded Voronoi diagram, was computed. Then the Eikonal equation was solved on the Voronoi diagram by means of the fast marching method [106]. Since the Voronoi diagram is a non-manifold surface made up of convex polygons, the fast marching method for triangulated 2-manifolds was extended to such domain retaining its efficiency. Backtracing of centerlines by steepest descent of the Eikonal equation solution was also tailored to polygonal non-manifolds. An original way to relate surface mesh points to centerlines has then been given in the same framework, performed by solving a second Eikonal equation on the Voronoi diagram on the basis of the computed centerlines, and providing the distance of surface points their corresponding centerline point. This approach takes advantage of Voronoi diagram topology, and is provably more robust than minimum distance computation in  $\mathbb{R}^3$ . Based on this approach, a new surface-based quantity, the normalized tangency deviation, was then proposed, which characterizes surface features in a scale-independent way. In order to better evidence this property, analytical expression of normalized tangency deviation for a cylinder with elliptical base was given as a function of ellipse eccentricity only. The techniques developed in this chapter have been presented by us in [6, 5].

In Chapter 7, a brief survey on local hemodynamics computation problem was given. Moreover, a few post-processing techniques implemented in this work have been presented. In particular, shear-stress derived quantities and Lagrangian flow visualization techniques in pulsatile environments were described and imple-

mented. In a final section, we showed two examples of the application of local hemodynamics characterization, performed on a carotid bifurcation acquired by CT angiography and on an abdominal aorta acquired by MR angiography. The results of velocity field, WSS and OSI distributions are presented. Good agreement with the literature concerning flow patterns and WSS related quantities (such as OSI) was found.

## 8.2 Conclusions and future directions

The techniques presented in this work constitute a step toward patient-specific blood vessel modeling, and follow the rationale of making geometry constitute a variable of the problem, rather than a pre-processing matter. From this point of view, criteria based on solid variational principles, as those employed in this work, for the resolution of the several steps involved are necessary to ensure robustness and accuracy. Moreover, the trade-off between flexibility and automation must be taken into account. In general, full automation the complex problem of blood vessel modeling is achieved at expense of generalization to different imaging modalities, image quality and vascular segments. Providing high-level interaction which does not affect reproducibility, processing time or ease of use, may overcome this problem, giving more control on the solutions. Besides finalizing 3D modeling to CFD computation, reconstruction and geometric characterization techniques are gaining attentions *per se*, since first patient-specific CFD studies have evidenced the major role played by geometry on fluid dynamics. Our stress on geometric analysis is therefore further justified by these consideration.

The presented techniques are currently employed for patient-specific modeling of large arteries acquired from different modalities, both from geometric and fluid-dynamics points of view. A study (yet to be published) on 100 carotid bifurcation models reconstructed at the Imaging Research Labs and the John Robarts Research Institute, London, Ontario, by the group of Prof. David Steinman, has further demonstrated the validity of the centerline computation method proposed. The centerlines for all 100 models were computed in less than one hour with full automation and accurate results.

Several improvements and further possibilities are offered on the basis on the present work. For 3D modeling, level sets offer the possibility to incorporate evolution terms which account for image statistics or prior knowledge [115], improving automation and sensitivity to noise. Topological analysis may take advantage from the use of more sophisticated probing functions than the simple sphere (e.g. ellipsoids or geodesic distance measures). Model editing may take advantage from implicit solvers for linear systems [34], relieving the constraints on the update step employed, thus improving convergence speed and robustness. Moreover, different functionals could be employed which preserve cylindrical structures, so to allow to find minimal energy configurations for entire branches (e.g. grafts). As to geometric analysis, further work has to be done on surface characterization, and the

normalized tangency deviation concept can be extended to generic shape description. Furthermore, a better characterization of the role of noise on Voronoi diagram structure or other medial axis approximation methods is due.

From a broader point of view, patient-specific modeling of large arteries is now at a turning point. The advances in computing power and numerical methods will allow to perform computations of complete cardiac cycles in minutes rather than in hours, and the improvement in medical imaging techniques and equipments will go toward fast and non-invasive acquisition of anatomy and velocity at high resolutions. The application of the currently available techniques to populations of patients and normal subjects, in order to better understand and evidence the relationships between geometry, local hemodynamics and pathological alterations, is possible, and is progressively being achieved. From this point, several paths must be followed.

One direction is the refinement of CFD modeling techniques, with the inclusion of distensible walls, non-Newtonian rheologic models, transport phenomena, coupling to 1D lumped parameter model of the whole circulation, in order to achieve better prediction of the *in vivo* flow patterns. Particular attention must be paid not to increase model complexity beyond the benefit. The introduction of modeling features must be weighted with the error deriving from the eventual missing knowledge about the new model parameters, also considering the additional computational cost introduced.

Another important direction is the switch from analysis to simulation. Performing a CFD computation for given boundary conditions does not account for the great variability observed under different psychophysical conditions. The lack of clear results about the localization of vascular injury with respect to local hemodynamics may be related to this problem. Stochastic simulation techniques, such as developed by Marczyk [70] in the context of structural mechanics, take into account variability in model input parameters and boundary conditions and, at expense of performing several simulations, provide precious information on the dynamics of the physical system.

Finally, the expertise in CFD analysis must be coupled with increasingly accurate velocimetric techniques, such as phase-contrast magnetic resonance. The use of such techniques in the clinical context will probably increase the interest in patient-specific local hemodynamics. Even if it will not be possible to acquire velocity components with adequate resolution or in regions of complex flow, the classic fluid dynamics framework, where a domain in which nothing about the solution is known has a boundary in which the solution is known in detail, will likely not represent the need. The problem will then be to reconstruct velocity and pressure fields from the coarser and incomplete acquired data ensuring compatibility with the Navier-Stokes equations. In such problem formulation, similar to *data assimilation* employed in environmental modeling and weather forecasting, the solution has to be generated by simultaneously performing two minimization processes, that are the minimization of the residuals of the solution to Navier-Stokes equations (as in current CFD modeling), and the minimization of the difference

between the computed velocity field and the acquired data (through the simulation of the acquisition procedure).

The above only represent a personal vision of the directions which may be followed in the near future. However, whatever scenarios will open, three dimensional geometry and local hemodynamics will play increasingly important roles in the investigation of the cardiovascular system.



## Appendix A

# The Euler-Lagrange equation

Here we give a sketch of a fundamental result of calculus of variations which has been used throughout this work.

Consider the following functional

$$J = \int_{\Omega} F \left( x_1, \dots, x_n, y, \frac{\partial y}{\partial x_1}, \dots, \frac{\partial y}{\partial x_n} \right) dx_1, \dots, dx_n \quad (\text{A.1})$$

where  $\Omega$  is a n-dimensional domain. Suppose that  $J$  has to be extremized over a set of admissible functions  $y$  with fixed boundary conditions  $y|_{\partial\Omega} = \bar{y}$ . Assume that  $y$  is piecewise  $C^1$  on  $\Omega$ . Then  $J$  has an extremum in  $y$  only if the Euler-Lagrange equation is satisfied

$$\frac{\partial F}{\partial y} - \sum_{h=1}^n \frac{\partial}{\partial x_h} \left( \frac{\partial F}{\partial y_h} \right) = 0 \quad (\text{A.2})$$

where  $y_h = \frac{\partial y}{\partial x_h}$ .

The proof is given in the following. Consider a piecewise  $C^1$  function  $\eta(x_1, \dots, x_n)$  over  $\Omega$ , with  $\eta|_{\partial\Omega} = 0$ . Function  $y + \alpha\eta$  is then an admissible function  $\forall \alpha \in \mathbb{R}$ . If  $y$  is an extremum of  $J$ , it means that

$$\begin{aligned} 0 &= \frac{d}{d\alpha} J(y + \alpha\eta) \Big|_{\alpha=0} \\ &= \int_{\Omega} \left( \eta \frac{\partial F}{\partial y} + \sum_{h=1}^n \frac{\partial F}{\partial y_h} \frac{\partial \eta}{\partial x_h} \right) d\Omega \end{aligned} \quad (\text{A.3})$$

Using Gauss theorem, is it possible to write

$$\begin{aligned} \int_{\Omega} \nabla \cdot \left( \eta \left[ \frac{\partial F}{\partial y_h} \right] \right) d\Omega &= \int_{\Omega} \sum_{h=1}^n \left( \frac{\partial F}{\partial y_h} \frac{\partial \eta}{\partial x_h} + \eta \frac{\partial}{\partial x_h} \frac{\partial F}{\partial y_h} \right) d\Omega \\ &= \int_{\partial\Omega} \eta \left[ \frac{\partial F}{\partial y_h} \right] \cdot \mathbf{n} dS \end{aligned} \quad (\text{A.4})$$

where  $\left[ \frac{\partial F}{\partial y_h} \right]$  denotes the n-dimensional vector whose h-th component is  $\frac{\partial F}{\partial y_h}$ , and  $\mathbf{n}$  is the outward normal to  $\partial\Omega$ . Therefore, since we chose  $\eta|_{\partial\Omega} = 0$ , the boundary term is zero, therefore Equation A.3 becomes

$$\int_{\Omega} \left[ \frac{\partial F}{\partial y} - \sum_{h=1}^n \frac{\partial}{\partial x_h} \left( \frac{\partial F}{\partial y_h} \right) \right] \eta d\Omega = 0 \quad (\text{A.5})$$

which must hold for every  $\eta$ . Therefore  $y$  must satisfy Equation A.2 in order to be an extremal function for  $J$ .

The generalization of the Euler-Lagrange equation to the cases in which function  $F$  depends on higher-order derivatives (up to the  $m$ -th order) takes the following form

$$\frac{\partial F}{\partial y} + \sum_{i=1}^m \sum_{h=1}^n \left[ (-1)^i \frac{\partial^{(i)}}{\partial x_h^{(i)}} \left( \frac{\partial F}{\partial y_h^{(i)}} \right) \right] = 0 \quad (\text{A.6})$$

where  $y_h^{(i)} = \frac{\partial^{(i)} y}{\partial x_h^{(i)}}$ .

## Appendix B

# Element types

In the following we will give a brief overview on the type of finite elements used for the discretization of the domains [52]. For each element, the expression of the shape functions used for interpolation over the element in the parametric space are indicated as  $\phi$ . The transformation from parametric to physical space is given by

$$\mathbf{x} = \mathbf{N}^T \mathbf{X} \quad (\text{B.1})$$

where  $\mathbf{N}$  is the matrix whose columns are the interpolation functions, and  $\mathbf{X}$  is the matrix whose rows are the nodal physical space coordinates.

In the following will not present an exhaustive list of element types, but we will rather focus on the elements used in the following sections. Since most meshing schemes proceed by first discretizing the boundary surface of the models and then the interior volumes, we will start presenting basic 2D elements for surface discretization, and then we will switch to 3D elements.

### B.1 2D elements

The most common 2D elements used for surface discretization are triangular and quadrilateral elements, with both linear and quadratic interpolation (refer to Figure B.1)

#### 3-node triangle

Linear triangles are the same as normally used for the representation of piecewise linear surfaces in 3D modeling. Shape functions are expressed by

$$\phi = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \end{pmatrix} \quad (\text{B.2})$$

where  $L_1, L_2, L_3$  are barycentric coordinates, also known as area coordinates, for which  $L_1 + L_2 + L_3 = 1$  holds.

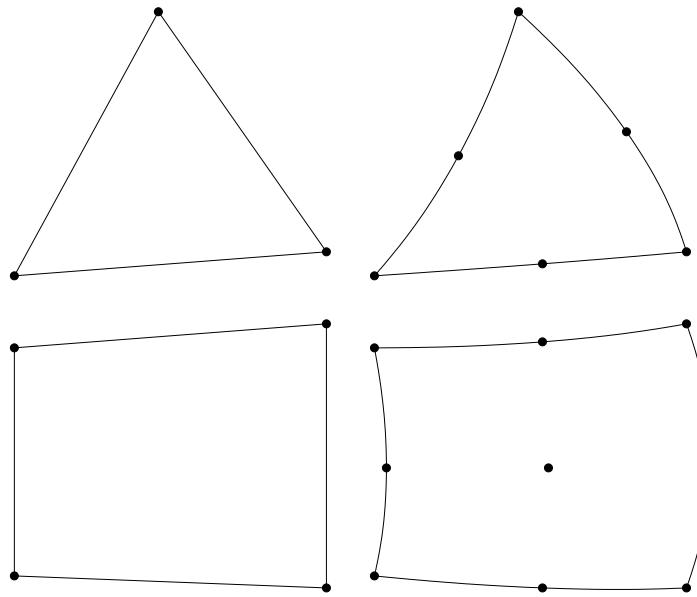


Figure B.1: 2D elements used in this work for mesh generation. From left to right and from top to bottom: 3-node triangle, 6-node triangle, 4-node quadrilateral, 9-node quadrilateral.

### 6-node triangle

Quadratic triangles are elements whose interpolation is given by quadratic functions. Shape functions are expressed by

$$\varphi = \begin{pmatrix} L_1(2L_1 - 1) \\ 4L_1L_2 \\ L_2(2L_2 - 1) \\ 4L_2L_3 \\ L_3(2L_3 - 1) \\ 4L_3L_1 \end{pmatrix} \quad (\text{B.3})$$

### 4-node quadrilateral

Shape functions for linear quadrilateral are given in terms of normalized or natural coordinates  $r, s$  in the parameter space, ranging from -1 to 1, taking the form

$$\varphi = \begin{pmatrix} \frac{1}{4}(1-r)(1-s) \\ \frac{1}{4}(1+r)(1-s) \\ \frac{1}{4}(1+r)(1+s) \\ \frac{1}{4}(1-r)(1+s) \end{pmatrix} \quad (\text{B.4})$$

### 9-node quadrilateral

Quadratic quadrilaterals use biquadratic interpolation functions of the following form

$$\varphi = \begin{pmatrix} \frac{1}{4}rs(1-r)(1-s) \\ -\frac{1}{2}s(1-s)(1-r^2) \\ -\frac{1}{4}rs(1+r)(1-s) \\ \frac{1}{2}r(1+r)(1-s^2) \\ \frac{1}{4}rs(1+r)(1+s) \\ \frac{1}{2}s(1+s)(1-r^2) \\ -\frac{1}{4}rs(1-r)(1+s) \\ -\frac{1}{2}r(1-r)(1-s^2) \\ (1-r^2)(1-s^2) \end{pmatrix} \quad (\text{B.5})$$

## B.2 3D elements

There is a wide variety of 3D elements available for finite element approximation. Their choice depends both on domain geometry and on the expected features of the solution. In the following sections some aspects of the criteria used for the choice will be made clear. A complete list of the element types along with the advantages and disadvantages of each goes beyond the scope of this chapter. Therefore, only the elements effectively employed in this work will be presented (see Figure B.2).

### 4-node tetrahedron

Linear tetrahedra are simplices in 3D (i.e. the most simple volumes which can be constructed in 3D). For this reason they are very flexible in terms of the type of geometry which can be discretized, in that there are no particular constraints on shape or surface mesh topology. The shape functions of linear tetrahedra are expressed by

$$\varphi = \begin{pmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{pmatrix} \quad (\text{B.6})$$

where  $L_1, L_2, L_3, L_4$  are again natural barycentric (or volume) coordinates, which satisfy  $L_1 + L_2 + L_3 + L_4 = 1$ .

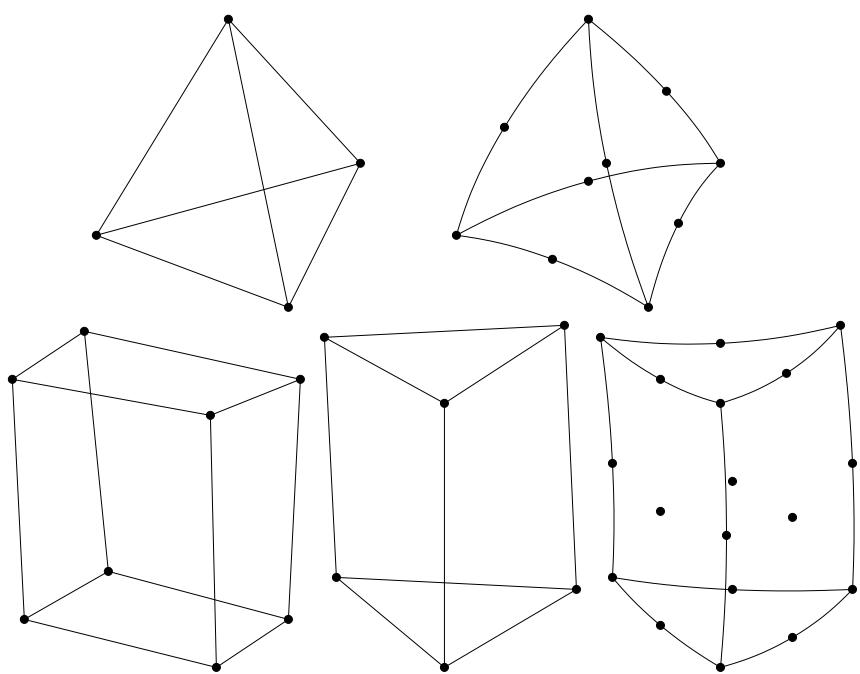


Figure B.2: 3D elements used in this work for mesh generation. From left to right and from top to bottom: 4-node tetrahedron, 10-node tetrahedron, 8-node hexahedron, 6-node wedge, 18-node wedge.

## 10-node tetrahedron

On quadratic tetrahedra, interpolation functions are given in terms of the following triquadratic functions

$$\varphi = \begin{pmatrix} L_1(2L_1 - 1) \\ 4L_1L_2 \\ L_2(2L_2 - 1) \\ 4L_1L_3 \\ 4L_2L_3 \\ L_3(2L_3 - 1) \\ 4L_1L_4 \\ 4L_2L_4 \\ 4L_3L_4 \\ L_4(2L_4 - 1) \end{pmatrix} \quad (\text{B.7})$$

## 8-node hexahedron

Linear hexahedral elements, also known as bricks, are geometrically less flexible than tetrahedra, since not every volume whose surface is meshed with quadrilaterals can be discretized with hexahedral elements. On the other hand, when properly used, these elements present some advantages over tetrahedra in terms of stability and accuracy of the CFD solution. The trilinear shape functions have the form

$$\varphi = \begin{pmatrix} \frac{1}{8}(1-r)(1-s)(1-t) \\ \frac{1}{8}(1+r)(1-s)(1-t) \\ \frac{1}{8}(1-r)(1+s)(1-t) \\ \frac{1}{8}(1+r)(1+s)(1-t) \\ \frac{1}{8}(1-r)(1-s)(1+t) \\ \frac{1}{8}(1+r)(1-s)(1+t) \\ \frac{1}{8}(1-r)(1+s)(1+t) \\ \frac{1}{8}(1+r)(1+s)(1+t) \end{pmatrix} \quad (\text{B.8})$$

where  $r,s,t$  are the normalized coordinates of the reference element, varying between -1 and 1.

## 6-node wedge

Wedges are triangular-based prisms. Their use in finite elements CFD codes is relatively recent. As shown in the following, they can be employed in hybrid meshes together with tetrahedra to locally improve mesh quality. Shape functions for linear

wedges are

$$\varphi = \begin{pmatrix} \frac{1}{2}L_1(1-t) \\ \frac{1}{2}L_2(1-t) \\ \frac{1}{2}L_3(1-t) \\ \frac{1}{2}L_1(1+t) \\ \frac{1}{2}L_2(1+t) \\ \frac{1}{2}L_3(1+t) \end{pmatrix} \quad (\text{B.9})$$

where  $L_1, L_2, L_3$  are natural barycentric coordinates for the reference element triangular base, and  $t$  is the normalized coordinate for reference element height, varying between -1 and 1.

### 18-node wedge

Quadratic wedges are normally used in conjunction with quadratic tetrahedra. Their shape function take the following form

$$\varphi = \begin{pmatrix} \frac{1}{2}L_1(2L_1-1)t(t-1) \\ 2L_1L_2t(t-1) \\ \frac{1}{2}L_2(2L_2-1)t(t-1) \\ 2L_3L_1t(t-1) \\ 2L_2L_3t(t-1) \\ \frac{1}{2}L_3(2L_3-1)t(t-1) \\ L_1(2L_1-1)(1-t^2) \\ 4L_1L_2(1-t^2) \\ L_2(2L_2-1)(1-t^2) \\ 4L_3L_1(1-t^2) \\ 4L_2L_3(1-t^2) \\ L_3(2L_3-1)(1-t^2) \\ \frac{1}{2}L_1(2L_1-1)t(t+1) \\ 2L_1L_2t(t+1) \\ \frac{1}{2}L_2(2L_2-1)t(t+1) \\ 2L_3L_1t(t+1) \\ 2L_2L_3t(t+1) \\ \frac{1}{2}L_3(2L_3-1)t(t+1) \end{pmatrix} \quad (\text{B.10})$$

## B.3 Differentiation over elements

At the beginning of this chapter the transformation for mapping reference element points into physical world point was given. When differentiation and integration operations are performed on the elements, it is necessary to explicit the relation between differential quantities in the parameter space and in the physical space.

The relation is expressed by

$$\begin{pmatrix} \frac{\partial \varphi}{\partial r} \\ \frac{\partial \varphi}{\partial s} \\ \frac{\partial \varphi}{\partial t} \end{pmatrix} = \mathbf{J} \begin{pmatrix} \frac{\partial \varphi}{\partial x} \\ \frac{\partial \varphi}{\partial y} \\ \frac{\partial \varphi}{\partial z} \end{pmatrix} \quad (\text{B.11})$$

where  $\mathbf{J}$  is the Jacobian matrix, expressed by

$$\mathbf{J} = \begin{pmatrix} \frac{\partial}{\partial r} \\ \frac{\partial}{\partial s} \\ \frac{\partial}{\partial t} \end{pmatrix} (\mathbf{N}^T \mathbf{X})^T \quad (\text{B.12})$$

where quantities are the same as introduced in Equation (B.1). The inverse relationship is retrieved inverting matrix  $\mathbf{J}$ .

The Jacobian is also involved in the computation of integral quantities in parametric space through the following relation

$$dxdydz = |\mathbf{J}| drdsdt \quad (\text{B.13})$$



# Bibliography

- [1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118:269–277, 1995.
- [2] N. Amenta, S. Choi, and R. Kolluri. The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19(2-3):127–153, 2001.
- [3] N. Amenta, S. Choi, and R. K. Kolluri. The power crust. In *Proceedings of Solid Modeling '01*, pages 249–260, 2001.
- [4] L. Antiga, B. Ene-Iordache, L. Caverni, G. P. Cornalba, and A. Remuzzi. Geometric reconstruction for computational mesh generation of arterial bifurcations from CT angiography. *Computerized Medical Imaging and Graphics*, 26:227–235, 2002.
- [5] L. Antiga, B. Ene-Iordache, and A. Remuzzi. Centerline computation and geometric analysis of branching tubular surfaces with application to blood vessel modeling. In *Journal of WSCG*, Plzen, Czech Republic, February 2003. Submitted.
- [6] L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. *IEEE Transactions on Medical Imaging*, 2003. In press.
- [7] L. Antiga, B. Ene-Iordache, G. Remuzzi, and A. Remuzzi. Automatic generation of glomerular capillary topological organization. *Microvascular Research*, 62(3):346–354, Nov 2001.
- [8] C. Arcelli and G. Sanniti di Baja. Euclidean skeleton via center-of-maximal-disc extraction. *Image Vision Comput*, 11(3):163–173, 1993.
- [9] J. P. Archie and R. W. Feldman. Critical stenosis of the internal carotid artery. *Surgery*, 89(1), Jan 1981.
- [10] L. H. Arroyo and R. T. Lee. Mechanisms of plaque rupture: mechanical and biologic interactions. *Cardiovascular Research*, 41:369–375, 1999.

- [11] D. Attali and J.-O. Lachaud. Delaunay conforming iso-surface, skeleton extraction and noise removal. *Computational Geometry: Theory and Applications*, 19:175–189, 2001.
- [12] D. Attali and A. Montanvert. Computing and simplifying 2D and 3D continuous skeletons. *Computer vision and image understanding*, 67(261-273), 1997.
- [13] C. Baillard and C. Barillot. Robust 3D segmentation of anatomical structures with level sets. In S. L. Delp, A. M. DiGioia, and B. Jaramaz, editors, *MICCAI*, volume 1935, pages 236–245. Springer, 2000.
- [14] G. K. Batchelor. *An introduction to fluid dynamics*. Cambridge University Press, 1967.
- [15] A. Belyaev and S. Yoshizawa. On evolute cusps and skeleton bifurcations. In *International Conference on Shape Modeling and Applications, Genova, Italy*, pages 134–141, May 2001.
- [16] M. Bern and P. Plassmann. Mesh generation. In *Handbook of Computational Geometry*, chapter 6. Elsevier Science, 1999.
- [17] T. D. Blacker. The Cooper tool. In *Proceedings of the Fifth International Meshing Roundtable*, pages 13–29, 1996.
- [18] H. Blum. A transformation for extracting new descriptors of shape. In *Models for the perception of speech and visual form*, pages 362–380. W. Wathen-Dunn Ed., MIT Press, Cambridge, MA, 1967.
- [19] M. Bongrazio, C. Baumann, A. Zakrzewicz, A. R. Pries, and P. Gaehtgens. Evidence for modulation of genes involved in vascular adaption by prolonged exposure of endothelial cells to shear stress. *Cardiovascular Research*, 47:384–393, 2000.
- [20] G. Borgefors. Distance transformations in digital images. *Computer Vision, Graphics and Image Processing*, 34:344–371, 1986.
- [21] G. Borgefors, I. Nyström, and G. Sanniti Di Baja. Computing skeletons in three dimensions. *Pattern Recognition*, 32:1225–1236, 1999.
- [22] R. Botnar, G. Rappitsch, M. B. Scheidegger, D. Liepsch, K. Perktold, and P. Boesiger. Hemodynamics in the carotid artery bifurcation: a comparison between numerical simulations and in vitro MRI measurements. *Journal of Biomechanics*, 33:137–144, 2000.
- [23] D. E. Brooks, J. W. Goodwin, and G. V. F. Seaman. Interactions among erythrocytes under shear. *Journal of Applied Physiology*, 28:172–177, 1970.

- [24] A. J. Bulpitt and N. D. Efford. An efficient 3D deformable model with a self-optimizing mesh. *Image and Vision Computing*, 14:573–580, 1996.
- [25] F. Cassot, M. Zagzoule, and J. P. Marc-Vergnes. Hemodynamic role of the circle of willis in stenoses of internal carotid arteries. an analytical solution of a linear model. *Journal of Biomechanics*, 33:395–405, 2000.
- [26] J. R. Cebral and R. Löhner. From medical images to anatomically accurate finite element grids. *International Journal for Numerical Methods in Engineering*, 51(985-1008), 2001.
- [27] J. J. Chiu, D. L. Wang, S. Chien, R. Skalak, and S. Usami. Effects of disturbed flow on endothelial cells. *Transactions of the ASME*, 120:2–8, Feb 1998.
- [28] H. I. Choi, S. W. Choi, and H. P. Moon. Mathematical theory of medial axis transform. *Pacific Journal of Mathematics*, 181(1):57–88, 1997.
- [29] L. D. Cohen and I. Cohen. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1131–1147, 1993.
- [30] L. D. Cohen and R. Kimmel. Global minimum of active contour models: a minimal path approach. *International Journal of Computer Vision*, 24(1):57–78, Aug 1997.
- [31] P. F. Davies. Flow-mediated endothelial mechanotransduction. *Physiological Reviews*, 75:519–560, 1995.
- [32] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):1–36, 1999.
- [33] N. DePaola, M. A. Gimbrone, P. F. Davies, and C. F. Dewey. Vascular endothelium responds to fluid shear stress gradients. *Arteriosclerosis and Thrombosis*, 12(11):1254–1257, 1992.
- [34] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In ACM SIGGRAPH, editor, *Proceedings of SIGGRAPH 1999*, pages 317–324, 1999.
- [35] T. Deschamps. *Curve and shape extraction with minimal path and level sets techniques. Applications to 3D medical imaging*. PhD thesis, Université Paris-IX Dauphine, France, Dec 2001.
- [36] T. Deschamps and L. D. Cohen. Fast extraction of minimal paths in 3D images and application to virtual endoscopy. *Medical Image Analysis*, 5(4):281–299, Dec 2001.

- [37] T. K. Dey and W. Zhao. Approximate medial axis as a Voronoi subcomplex. In *Proceedings of the 7th ACM Symposium of Solid Modeling Applications*, pages 356–366, 2002.
- [38] B. Ene-Iordache, L. Mosconi, G. Remuzzi, and A. Remuzzi. Computational fluid dynamics of a vascular access case for hemodialysis. *Journal of Biomechanical Engineering*, 123(3):284–292, June 2001.
- [39] D. Eppstein. Linear complexity hexahedral mesh generation. *Computational Geometry*, 12:3–16, 1999.
- [40] C. R. Ethier, D. A. Steinman, X. Zhang, S. R. Karpik, and M. Ojha. Flow waveform effects on end-to-side anastomotic flow patterns. *Journal of Biomechanics*, 31:609–617, 1998.
- [41] L. Formaggia, F. Nobile, J.-F. Gerbeau, and A. Quarteroni. Numerical treatment of defective boundary conditions for the Navier Stokes equations. *SIAM Journal of Numerical Analysis*, 40(1):376–401, 2002.
- [42] S.J. Fortune. Voronoi diagrams and Delaunay triangulations. In J. E. Goodman, Joseph O'Rourke, J. Josephn, and J. C'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 377–388. CRC Press, 1997.
- [43] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. van Walsum, and M. A. Viergever. Model-based quantitation of 3-D magnetic resonance angiographic images. *IEEE Transactions on Medical Imaging*, 18(10):946–956, Oct 1999.
- [44] N. Gagvani and D. Silver. Parameter controlled skeletonization of three dimensional objects. Technical Report CAIP-TR-216, Rutgers State University of New Jersey, Jun 1997.
- [45] G. Garcia-Cardena, J. Comander, K. R. Anderson, B. R. Blackman, and M. A. Gimbrone. Biomechanical activation of vascular endothelium as a determinant of its functional phenotype. *Proceedings of the National Academy of Science*, 98(8):4478–4485, Apr 2001.
- [46] K. J. Gooch and C. J. Tennant. *Mechanical forces: their effects on cells and tissues*. Springer-Verlag, Berlin, 1997.
- [47] G. Greiner. Variational design and fairing of spline surfaces. *Computer Graphics Forum*, 13(3):143–154, 1994.
- [48] D. W. Holdsworth, C. J. D. Norley, R. Frayne, D. A. Steinman, and B. K. Rutt. Characterization of common carotid artery blood-flow waveforms in normal human subjects. *Physiological Measurement*, 20:219–240, 1999.



realistic boundary conditions for the study of local and global hemodynamics. In G. W. Schmid-Schönbein, editor, *Proceedings of the 2001 Summer Bioengineering Conference - BED*, volume 50, pages 533–534, New York, NY, 2001.

- [62] T. Lee and R. L. Kashyap. Building skeleton models via 3D medial surface/axis thinning algorithms. *CVGIP: Graphical Models and Image Processing*, 56(6):462–478, 1994.
- [63] M. Lei, C. Kleinstreuer, and G. A. Truskey. Numerical investigation and prediction of atherogenic sites in branching arteries. *Transactions of the ASME*, 117:350–357, 1995.
- [64] R. Löhner. Regridding surface triangulations. *Journal of Computational Physics*, 126:1–10, 1996.
- [65] Q. Long, X. Y. Xu, M. W. Collins, M. Bourne, and T. M. Griffith. Magnetic resonance image processing and structured grid generation of a human abdominal bifurcation. *Computer Methods and Programs in Biomedicine*, 56:249–259, 1998.
- [66] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3D surface construction algorithm. *Computer Graphics*, 21:163–169, 1987.
- [67] A. M. Malek, S. L. Alper, and S. Izumo. Hemodynamic shear stress and its role in atherosclerosis. *JAMA*, 282:2035–2042, 1999.
- [68] A. M. Malek and S. Izumo. Control of endothelial cell gene expression by flow. *Journal of Biomechanics*, 28(12):1515–1528, 1995.
- [69] R. Malladi and J. A. Sethian. A real-time algorithm for medical shape recovery. *Proceedings International Conference on Computer Vision*, pages 304–310, Jan 1998.
- [70] J. Marczyk. *Principles of simulation based computer-aided engineering*. FIM Publications, Barcelona, Spain, 1999.
- [71] T. McInerney and D. Terzopoulos. A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4D image analysis. *Computerized Medical Imaging and Graphics*, 19(1):69–83, 1995.
- [72] T. McInerney and D. Terzopoulos. Topologically adaptable snakes. In *Proceedings of the Fifth International Conference on Computer Vision (ICCV'95)*, pages 840–845, Cambridge, MA, June 1995.
- [73] T. McInerney and D. Terzopoulos. Deformable models in medical image analysis: a survey. *Medical Image Analysis*, 1(2):91–108, 1996.

- [74] M. Meyer, M. Desbrun, P. Schröder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *International Workshop on Visualization and Mathematics, Berlin-Dahlem (Germany)*, pages 22–25, May 2002.
- [75] J. S. Milner, J. A. Moore, B. K. Rutt, and D. A. Steinman. Hemodynamics of human carotid artery bifurcations: computational studies with models reconstructed from magnetic resonance imaging of normal subjects. *Journal of Vascular Surgery*, 28:143–156, 1998.
- [76] J. A. Moore, D. A. Steinman, D. W. Holdsworth, and C. R. Ethier. Accuracy of computational hemodynamics in complex arterial geometries reconstructed from magnetic resonance imaging. *Annals of Biomedical Engineering*, 27:32–41, 1999.
- [77] J. E. Moore and D. N. Ku. Pulsatile velocity measurements in a model of the human abdominal aorta under resting conditions. *Journal of Biomechanical Engineering*, 116:337–346, 1994.
- [78] J. E. Moore and D. N. Ku. Pulsatile velocity measurements in a model of the human abdominal aorta under simulated exercise and postprandial conditions. *Journal of Biomechanical Engineering*, 116:107–111, 1994.
- [79] J. E. Moore, C. Xu, S. Glagov, C. K. Zarins, and D. N. Ku. Fluid wall shear stress measurements in a model of the human abdominal aorta: oscillatory behavior and relationship to atherosclerosis. *Atherosclerosis*, 110:225–240, 1994.
- [80] R. M. Nerem. Vascular fluid mechanics, the arterial wall, and atherosclerosis. *Transactions of the ASME*, 114:274–282, 1992.
- [81] T. O’Donnel, X. S. Fang, A. Gupta, and T. Boult. The extruded generalized cylinder: a deformable model for object recovery. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, pages 104–109. IEEE, Jun 1994.
- [82] R.L. Ogniewicz. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the Euclidean geometric heat flow onto Voronoi skeletons. Technical Report 95-4, Harvard Robotics Laboratory, Harvard University, 1995.
- [83] R.L. Ogniewicz and M. Ilg. Voronoi skeletons: theory and applications. In *Proceedings of Conference on Computer Vision and Pattern Recognition*, pages 63–69, Jun 1992.
- [84] S. V. Patankar. *Numerical heat transfer and fluid flow*. McGraw-Hill, 1980.

- [85] N. J. Pelc, F. G. Sommer, K. C. P. Li, T. J. Brosnan, R. J. Herfkens, and D. R. Enzmann. Quantitative magnetic resonance flow imaging. *Magnetic Resonance Quarterly*, 10(3):125–147, 1994.
- [86] K. Perktold, M. Hofer, G. Rappitsch, M. Loew, B. D. Kuban, and M. H. Friedman. Validated computation of physiologic flow in a realistic coronary artery branch. *Journal of Biomechanics*, 31:217–228, 1998.
- [87] K. Perktold and G. Rappitsch. Computed simulation of local blood flow and vessel mechanics in a compliant carotid artery bifurcation model. *Journal of Biomechanics*, 28(7):845–856, 1995.
- [88] K. Perktold and M. Resch. Numerical flow studies in human carotid artery bifurcations: basic discussion on the geometric factor in atherogenesis. *Journal of Biomedical Engineering*, 12:111–123, 1990.
- [89] K. Perktold, M. Resch, and H. Florian. Pulsatile non-newtonian flow characteristics in a three-dimensional human carotid bifurcation model. *Transactions of the ASME*, 113:464–475, 1991.
- [90] S. Prakash and C. R. Ethier. Requirements for mesh resolution in 3D computational hemodynamics. *Journal of Biomechanical Engineering*, 123(2):134–144, 2001.
- [91] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical recipes in C*. Cambridge University Press, 2nd edition, 1997.
- [92] C. Pudney. Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Computer Vision and Image Understanding*, 72(3):404–413, Dec 1998.
- [93] A. Puig. Discrete medial axis transform for discrete objects. Technical report, Polytechnical University of Catalunya, Spain, 1998.
- [94] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*. Springer, Heidelberg, Germany, 2000.
- [95] D. Quemada. A rheological model for studying the hematocrit dependence of red cell-red cell and red cell-protein interactions in blood. *Biorheology*, 18:501–506, 1981.
- [96] A. Redaelli, G. Rizzo, S. Arrigoni, E. Di Martino, D. Origgi, F. Fazio, and F. Montevercchi. An assisted automated procedure for vessel geometry reconstruction and hemodynamic simulations from clinical imaging. *Computerized Medical Imaging and Graphics*, 26:143–152, 2002.
- [97] A. Remuzzi, B. Ene-Iordache, L. Mosconi, S. Bruno, A. Anghileri, L. Antiga, and G. Remuzzi. Radial artery wall shear stress evaluation in

- patients with arteriovenous fistula for hemodialysis access. *Biorheology*, 1-3:423–430, 2003.
- [98] R. Ross. The pathogenesis of atherosclerosis – an update. *The New England Journal of Medicine*, 314(8):488–500, 1986.
  - [99] R. Ross and J. A. Glomset. The pathogenesis of atherosclerosis. *The New England Journal of Medicine*, 295:369–420, 1976.
  - [100] D. Rypl and Z. Bittnar. Discretization of 3D surfaces reconstructed by interpolating subdivision. In *Numerical Grid Generation in Computational Field Simulations*, pages 679–688, Mississippi, 2000. The International Society of Grid Generation.
  - [101] T. Saito and J. Toriwaki. A sequential thinning algorithm for three-dimensional digital pictures using the Euclidean distance transformation. In *Proceedings of the Ninth Scandinavian Conference of Image Analysis*, 1994.
  - [102] H. Schirmacher, M. Zöckler, D. Stalling, and H. C. Hege. Boundary surface shrinking - a continuous approach to 3D center line extraction. In B. Girod, H. Niemann, , and H. P. Seidel, editors, *Proceedings of Image and Multidimensional Signal Processing '98*, pages 25–28, 1998.
  - [103] R. Schneider and L. Kobbelt. Generating fair meshes with  $G^1$  boundary conditions. In *Geometric Modeling and Processing 2000 Proceedings*, pages 251–261, 2000.
  - [104] R. Schneider and L. Kobbelt. Geometric fairing of irregular meshes for free-form surface design. *Computer Aided Geometric Design*, 18(4):359–379, 2001.
  - [105] W. Schroeder, K. Martin, and W. E. Lorensen. *The Visualization Toolkit*. Prentice Hall, Upper Saddle River, New Jersey, 2nd edition, 1998.
  - [106] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.
  - [107] J. A. Sethian and A. Vladimirsky. Fast methods for the Eikonal and related Hamilton-Jacobi equations on unstructured meshes. *Proceedings of the National Academy of Sciences*, 97(11):5699–5703, May 2000.
  - [108] E. Sifakis, C. Garcia, and G. Tziritas. Bayesian level sets for image segmentation. *Journal of Visual Communication and Image Representation*, 13:44–64, 2002.
  - [109] D. A. Steinman. Simulated pathline visualization of computed periodic blood flow patterns. *Journal of Biomechanics*, 33(5):623–628, May 2000.

- [110] D. A. Steinman. Image-based CFD modeling in realistic arterial geometries. *Annals of Biomedical Engineering*, 30(4):483–497, Apr 2002.
- [111] D. A. Steinman, C. R. Ethier, and B. K. Rutt. Combined analysis of spatial and velocity displacement artifacts in phase contrast measurements of complex flows. *Journal of Magnetic Resonance Imaging*, 7:339–346, 1997.
- [112] D. A. Steinman, J. B. Thomas, H. M. Ladak, J. S. Milner, B. K. Rutt, and J. D. Spence. Reconstruction of carotid bifurcation hemodynamics and wall thickness using computational fluid dynamics and MRI. *Magnetic Resonance in Medicine*, 47(1):149–159, 2002.
- [113] A. G. Straatman, R. E. Khayat, E. Haj-Qasem, and D. A. Steinman. On the hydrodynamic stability of pulsatile flow in a plane channel. *Physics of Fluids*, 14(6):1938–1944, Jun 2002.
- [114] M. Styner and G. Gerig. Three-dimensional medial shape representation incorporating object variability. In *Proceedings of Computer Vision and Pattern Recognition*, 2001.
- [115] J. S. Suri, L. Kecheng, S. Singh, S. N. Laxminarayan, Z. Xiaolan, and L. Reden. Shape recovery algorithms using level sets in 2-D/3-D medical imagery: a state-of-the-art review. *IEEE Transactions on Information Technology in Biomedicine*, 6(1):8–28, 2002.
- [116] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95 Conference Proceedings*, pages 351–358, August 1995.
- [117] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins. Computational investigation in vascular disease. *Computers in Physics*, 10(3):224–232, May/June 1996.
- [118] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins. Finite element modeling of blood flow in arteries. *Computer Methods in Applied Mechanics and Engineering*, 158(1-2):155–196, 1998.
- [119] C. A. Taylor, T. J. R. Hughes, and C. K. Zarins. Finite element modeling of three-dimensional pulsatile flow in the abdominal aorta: relevance to atherosclerosis. *Annals of Biomedical Engineering*, 26:975–987, 1998.
- [120] J. B. Thomas, J. S. Milner, and D. A. Steinman. On the influence of vessel planarity on local hemodynamics at the human carotid bifurcation. *Biorheology*, 39(3-4):443–448, 2002.
- [121] G. B. Thurston. Rheological parameters for the viscosity, viscoelasticity and thixotropy of blood. *Biorheology*, 16:149–162, 1979.
- [122] G. A. Truskey, K. M. Barber, T. C. Robey, L. A. Olivier, and M. P. Combs. Characterization of a sudden expansion flow chamber to study the response

- of endothelium to flow recirculation. *Journal of Biomechanical Engineering*, 117:203–210, 1995.
- [123] C. M. van Bemmel, L. Spreeuwers, M. A. Viergever, and W. J. Niessen. Level-set based carotid artery segmentation for stenosis grading. In *MICCAI*, volume 2489, pages 36–43. Springer, 2002.
  - [124] C. M. van Bemmel, L. J. Spreeuwers, B. Verdonck, M. A. Viergever, and W. J. Niessen. Blood pool agent contrast-enhanced MRA: level-set-based artery-vein separation. In M. Sonka and J. M. Fitzpatrick Editors, editors, *Proceedings of SPIE*, volume 4684, pages 1464–1475, 2002.
  - [125] G. van Langenhove, J. J. Wentzel, R. Kramps, C. J. Slager, J. N. Hamburger, and P. W. Serruys. Helical velocity patterns in a human coronary artery. a three-dimensional computational fluid dynamic reconstruction showing the relation with local wall thickness. *Circulation*, 102:e22–e24, 2000.
  - [126] A. Verroust and F. Lazarus. Extracting skeletal curves from 3D scattered data. Technical Report RR 3250, INRIA, Rocquencourt, 1997.
  - [127] K. C. Wang, R. W. Dutton, and C. A. Taylor. Improving geometric model reconstruction for blood flow modeling. *IEEE Engineering in Medicine and Biology Magazine*, 18(6):33–39, Nov/Dec 1999.
  - [128] D. R. Wells, J. P. Archie, and C. Kleinstreuer. Effect of carotid artery geometry on the magnitude and distribution of wall shear stress gradients. *Journal of Vascular Surgery*, 23(4):667–678, 1996.
  - [129] R. Whitaker. Algorithms for implicit deformable models. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 822–827, 1995.
  - [130] R. Whitaker. A level set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–231, 1998.
  - [131] J. R. Womersley. Method for the calculation of velocity, rate of flow, and viscous drag in arteries when the pressure gradient is known. *Journal of Physiology (London)*, 127:553–563, 1955.
  - [132] Z. J. Wood. Semi-regular mesh extraction from volumes. Master’s thesis, Caltech, Pasadena, CA, 2000.
  - [133] Q.J Wu and J. D. Bourland. Three-dimensional skeletonization for computer-assisted treatment planning in radiosurgery. *Computerized Medical Imaging and Graphics*, 24:243–251, 2000.
  - [134] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Transactions on Image Processing*, pages 359–369, March 1998.

- [135] P. J. Yim, J. R. Cebral, R. Mullick, and P. J. Choyke. Vessel surface reconstruction with a tubular deformable model. *IEEE Transactions on Medical Imaging*, 20(12):1411–1421, Dec 2001.
- [136] P. J. Yim, R. Mullick, R. M. Summers, H. Marcos, J. R. Cebral, R. Lohner, and P. L. Choyke. Measurement of stenosis from magnetic resonance angiography using vessel skeletons. In *Proceedings of SPIE*, volume 3978, pages 245–255, 2000.
- [137] C. K. Zarins, D. P. Giddens, B. K. Bharadvai, V. S. Sotiurai, R. F. Mabon, and S. Glagov. Carotid bifurcation atherosclerosis – quantitative correlation of plaque localization with flow velocity profiles and wall shear stress. *Circulation Research*, 1983.
- [138] S. Z. Zhao, X. Y. Xu, M. W. Collins, A. V. Stanton, A. D. Hughes, and S. A. Thom. Flow in carotid bifurcations: effect of the superior thyroid artery. *Medical Engineering and Physics*, 21:207–214, 1999.
- [139] S. Z. Zhao, X. Y. Xu, A. D. Hughes, S. A. Thom, A. V. Stanton, B. Ariff, and Q. Long. Blood flow and vessel mechanics in a physiologically realistic model of a human carotid arterial bifurcation. *Journal of Biomechanics*, 33:975–984, 2000.
- [140] D. Zorin, P. Schröder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. In *Proceedings of SIGGRAPH 1996*, pages 189–192. ACM SIGGRAPH, 1996.