POLITECNICO DI TORINO

Department of Mechanical and Aerospace Engineering

Master's Degree Course

in Biomedical Engineering

PATIENT-SPECIFIC MODELS OF DISEASED CAROTID BIFURCATIONS TREATED WITH PATCH

Supervisors: **Candidate:**

Prof. Umberto Morbiducci   Concetto Micciulla

Prof. Diego Gallo   242457

A.Y. 2017-2018

A nonno Tore e nonno Concetto

Abstract

Oxygen-rich blood is principally carried by carotid arteries from the heart to the brain. Each of those arteries is subdivided in common carotid (CCA) that ascends from the aortic arch and internal carotid (ICA) and external carotid (ECA) that bring blood, mainly, to the brain and to the face respectively. Those arteries can be affected by many pathologies, the most common is carotid stenosis, a progressive reduction of the lumen. This pathology develops with higher frequency between elderly. The lumen narrowing is the consequence of the formation of a lipidic plaque between intima and media.

It is possible to treat this pathology with different techniques, one of the most common is endarterectomy via patch graft. It consists in the insertion of a biocompatible material after surgical removal of the plaque and is intended to restore the physiological lumen width. After-surgery restenosis risk is important to evaluate through hemodynamic since "disturbed flow" promotes atherogenic environment.

The data set is composed by two patient-specific cases treated with carotid endarterectomy, reconstructed from CT scans.

The aim of this study is to characterize patient-specific models treated with patch and linking a geometrical analysis and computational fluid dynamics (CFD). Finally, the results obtained were compared with the findings of another study of patients treated with stent.

The geometrical analysis was carried out with a Python automatic script able to extract from a carotid model the geometrical parameters that characterize the models in terms of curvature, torsion and expansion of the carotid bifurcation from CCA, as described in Bijari *et al. 2012*.

The hemodynamic characterization was made performing some CFD simulations. In order to have the better compromise between solution accuracy and computational cost a sensitivity analysis was carried out to choose the grid cardinality.

For the unsteady simulation, at the inflow and outflow sections were set a time dependent velocity profile taken from literature. The CFD analysis was intended to evaluate wall-shear-stress-based descriptors like: Time Average Wall Shear Stress (TAWSS), Oscillatory Shear Index (OSI) and Relative Residence Time (RRT) and helicity-based descriptors like: Localized Normalized Helicity (LNH), Mean Helicity (h_1), Helicity Intensity (h_2) and helicity topology descriptors (h_3 and h_4). Referring to the 10th and 20th percentile for TAWSS and to the 80th and 90th for OSI and RRT was determined the percentage of the vessel area affected by critic values of TAWSS, OSI and RRT.

From the results obtained with the geometrical analysis is clear how endarterectomy via patch mostly influences the flare of the vessel. The CFD shows that the percentage area exposed to risk values of TAWSS, OSI and RRT is below 20%. Helicity-based descriptors h_1 and h_2 show the presence of counter-rotating flows, while from h_4 is possible to observe that this flow structures are approximately balanced.

From the comparison of the results of this study with those of other 2 models treated with stent, it was noticed how the influence of the two techniques on the geometry is different and that both expose the carotid to similar risk of restenosis.

INDEX

List of Figures

List of Tables

List of Charts

Chapter 1

The Cardiovascular System and Carotid Bifurcation Pathologies

The aim of this chapter is to give a general overview about the cardiovascular system and in the second part to focus on the carotid bifurcations.

The first part it is divided in three different parts: the heart, vessel's properties and the blood. In particular, it describes: the structure of the heart and how it works, properties and composition of the vessels, the composition of the blood and specifically its rheology, density, viscosity and different rheological model used to approximate blood behaviour.

Moving to the second section, first of all, it gives a general overview about the carotid bifurcation and atherosclerosis. After that the first paragraph defines the anatomy of carotid bifurcation. The second section is about what is atherosclerosis and how it develops in the carotid bifurcation. The third part describes the atherosclerosis diagnosis and surgery.

1.1 General information
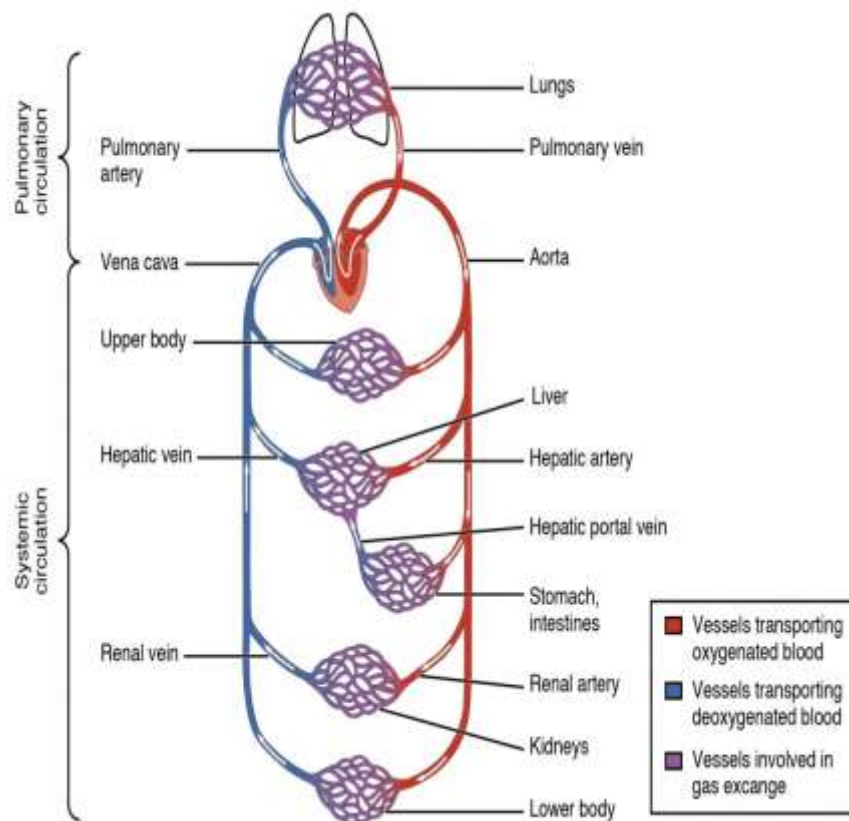
Figure 1.1: Cardiovascular system.

The cardiovascular system includes the heart and the vessels, the function of the heart is to pump blood through arteries, veins and capillaries.

The blood carries oxygen, nutrients, wastes and immune and other functional cells needed by human cells and organs to provide for homeostasis and basic living functions. The pumping action of the heart keeps the balance between cardiac output and venous return. Cardiac output (CO) is the quantity of blood pumped in to the vessels by each ventricle in one minute.

The normal adult blood volume is 5 litres and it usually goes through the heart once every minute. As may be expected the cardiac output varies with the demand of the body.

The cardiac cycle can be split in two phases, ventricular systole (contraction/ejection phase) and diastole (relaxation/filling phase) and refers to events that occur during one heartbeat. A normal heart rate is around 72 beats/minute, and the cardiac cycle lasts 0.8 seconds.

Blood vessels can be distinguished in arteries and veins; arteries transport blood away from the heart, while veins transport the blood back to the heart. Capillaries carry blood to tissue cells and are the exchange sites of nutrients, gases, wastes, etc. [1]

1.2 The heart
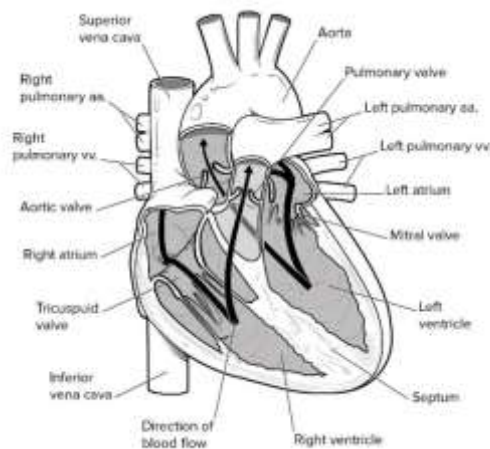
Figure 1.2: Heart structure. [1]

The heart is situated obliquely in the mediastinum, it is a muscular organ weighing 250-300 grams. It supplies blood to the all body and accepts it in return for gas exchange in the pulmonary circuit.

The heart is divided in 4 chambers, subdivided in two different types: atrium and ventricle. The right-side chambers supply the pulmonary circulation and the left side is necessary for supplying the systemic circulation. The chambers are divided by an atrioventricular valve (A-V valve). In the specific, the right-side chambers are divided by the tricuspid valve and left-sided chambers by the mitral valve.

Blood can go in only one direction thanks to the valvular system that, basing on pressure gradients, adapts opening and closure of valves. Cardiac muscle cells are branching striated, uninucleate cells which include also myofibrils. Desmosomes and gap junctions are contained in intercalated discs that are the link between adjacent cardiac cells. Due to electrical coupling the myocardium works as a functional syncytium.

Mitochondria are inside cardiac muscle, they are necessary for aerobic respiration because they produce adenosine tri-phosphate (ATP), which is an important energy contribute to cellular function. [1]

1.3 Vessel's properties

The blood vessels are organised in a closed system of tubes that transport blood to all the body and back to the heart. Arteries are blood vessel that transfer blood away from the heart, where they branch into ever-smaller vessels. Arterioles are the smallest arteries that further become tinier and are called capillaries where wastes and nutrients are exchanged with cells. Veins are larger vessels that return blood to the heart, inside them flows blood coming from smaller vessels.

There are two different circuits: the pulmonary circuit and the systemic circuit, where veins and arteries transport blood. Blood rich in oxygen is brought by systemic arteries to the tissues of the body. Systemic veins bring blood poor in oxygen back to the heart, cause thanks to the arteries a lot of oxygen has been distributed to the cells. Blood poor in oxygen comes back to the lungs trough arteries for gas exchange. After, freshly oxygenated blood returns from pulmonary veins, and is pumped back out in the systemic circulation from the lungs to the heart.

Since arterioles and arteries are closer to the heart, they have thicker walls than veins and venules, this is necessary to accept blood flowing at a far greater pressure. The lumen size is a characteristic of each type of vessel. Veins have bigger lumens than arteries that have to maintain a steady pressure of blood through the system. Arterial lumens have thicker walls and small diameters, so they appear more rounded in cross section than the lumen of veins. [2]
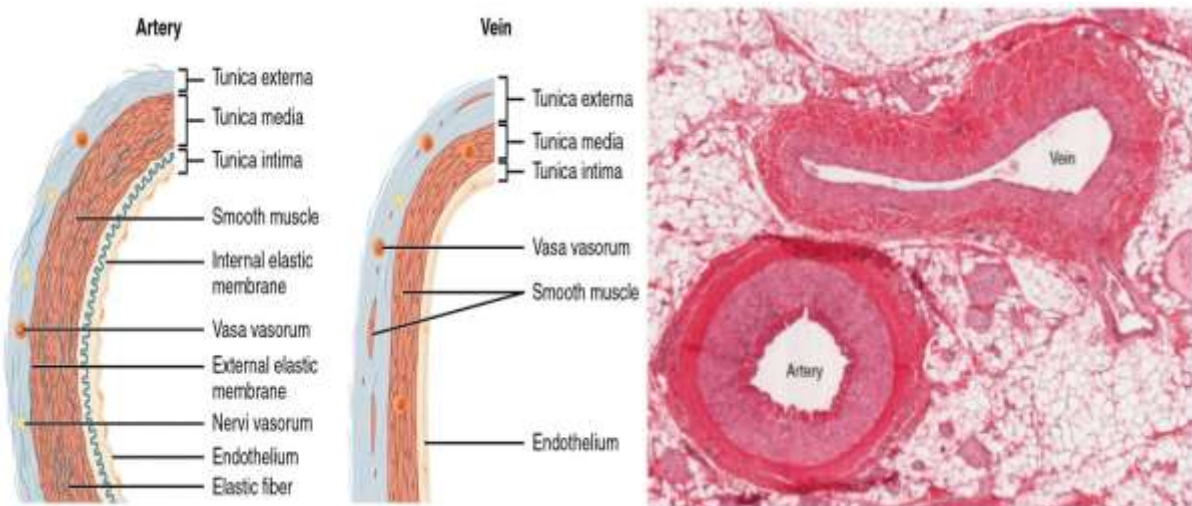
Figure 1.3: Arteries and veins structure

The **tunica intima** is a monolayer of endothelial cells and it is the only barrier between blood and their components and all the underlying thrombogenic tissues. The integrity of the endothelium is essential for maintaining the normal structure of the all vessel. Furthermore, the endothelial damage is responsible, at least at the beginning, of atherogenesis. The endothelium releases local chemicals called endothelins that can choke the smooth muscle within the walls of the vessel to increase blood pressure. Furthermore, uncompensated overproduction of endothelins may contribute to hypertension and cardiovascular disease. [2]

Basal lamina is the basement membrane next to the endothelium, composed by connective sub-endothelium tissue, proteoglycans, elastin and other glycoproteins of matrix. It has a structural and filtering function. Internal elastic membrane is a thick layer composed by elastic fibers disposed longitudinally. Such layer is not continuous, but porous. Through these spaces, especially in the arteries of medium calibre, the smooth muscle cells can penetrate and reach the intima. [2]
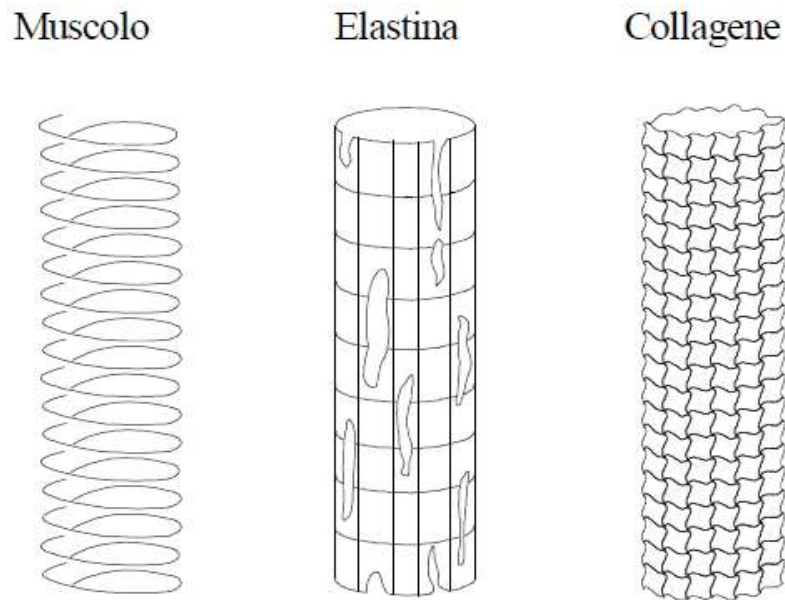
Figure 1.4: Arterial walls constituents. [3]

The **tunica media**, or muscular layer, is composed of different layers of smooth muscular cells supported by connective tissue that is primarily composed of elastic fibers, most of which are arranged in circular sheets. Concerning the outer portion of the tunic, there are also layers of longitudinal muscle cells. The diameter of the vessel lumen changes thanks to the contraction and relaxation of the circular muscles. Blood flow is diminished with vasoconstriction, making the lumen tinier and augmenting blood pressure, as the smooth muscles in the walls of the tunica media contracts. Correspondingly, blood flow is increased by vasodilation, allowing the lumen to dilate and blood pressure to fall, as the smooth muscle relaxes. *Nervi vasorum* are small vascular nerves that administrate vasodilatation and vasoconstriction. Since they are usually sympathetic fibers, according to the nature and the receptor location on the target cell and the neurotransmitter, some trigger vasodilation and other are related to vasoconstriction. Erection, during sexual arousal in the external genitalia of both sexes, is due to vasodilatation activated by parasympathetic stimulation. With collagenous fibers there is a large number of elastic fibers that present as wavy lines in prepared slides. Dividing the tunica media from the outer tunica externa in larger arteries there is the external elastic membrane, which also looks wavy in slides. This structure is not usually seen in smaller arteries nor in veins. [2]

The **tunica externa** or outer tunic, also called the tunica adventitia, is a scabbard of connective tissue constituted primarily of collagenous fibers. Furthermore, it has also some band of elastic fibers. Smooth muscle fibers are also in the tunica externa in veins, which is the thickest tunic in veins and may be thicker than the tunica media in some larger arteries. [2]

Blood is guided away from the heart thanks to arteries. All arteries have thick walls for bearing the high pressure of blood expelled from the heart. They are containing a high percentage of elastic fibers in all their layers. As blood pumped from the ventricles passes through them, elastic fibers permit them to expand, and after the surge has passed to recoil. In order to preserve the heart from pumping harder to augment the volume of blood casted out by each drain and carry on adequate pressure and flow, artery walls are able to expand and shrink, if not so their resistance to blood flow would abundantly augment and blood pressure would increment to higher levels. The elastic shrinking of the vascular walls helps to keep a steady pressure gradient in order to drive the blood through the arterial system. Since the large diameter of the lumen allows it to accept a large volume of blood from the heart and drive it to smaller branches, elastic arteries are called conducting arteries. The percentage of elastic fibers in tunica intima of the artery diminishes farther from the heart while the amount of smooth muscles in its tunica media augments; here the artery is defined as muscular artery. The diameter of muscular arteries extents from 0.1 mm to 10 mm. [2]

1.4 The blood

Blood is a tissue composed of a suspension of different elements in an aqueous solution, called plasma, which has as main activity carrying the different substances inside the body.

The blood has many essential functions:

- Supplies the tissues with oxygen through haemoglobin enclosed in erythrocytes and brings carbon dioxide away from them;

- Brings nutrients (amino acids, sugars, mineral salts) and collects the wastes from the various apparatus;

- Contributes to body temperature regulation;

- Brings hormones, enzymes and vitamins;

- Controls the defence of the organism through the work of white blood cells;

- By the coagulation, it achieves the immediate healing of wounds, awaiting the interference of long-term tissue reintegration mechanisms.

The generation of the corpuscular component (Haematopoiesis) takes place during the first years of life in the bone marrow of the short and flat bones; all the organs and tissues where this process takes place are called blood dyscrasias. Every corpuscular element of blood origins from a unique undifferentiated parent cell, called hemocytoblast. With the centrifugation the particulate matter can be divided by the liquid part (plasma) and the volume percentage taken from the matter elements is called haematocrit (Hct). The haematocrit of a healthy adult is between 38% and 52% for man and between 36% and 46% for woman.

Erythrocytes are little cells without nucleus with an average diameter of 8 μm. Their shape is a biconcave disk, surrounded by a membrane called stroma. The haemoglobin is able to link reversibly with the oxygen and appreciably augments the quantity of transported oxygen. In fact, the amount of oxygen in the blood could be calculated as the sum of dissolved oxygen concentration into the plasma ($[O_2]$) and the concentration of oxyhaemoglobin ($[HbO_2]$), expressed in molar units (mmol/l):

$$ctO_2 = [O_2] + [HbO_2]$$

The dissolved oxygen concentration depends on oxygen partial pressure (pO_2) according to Henry's law:

$$O_2 = [\alpha O_2] + [pO_2]$$

where α is the coefficient of oxygen solubility in the blood. In arterial normal conditions about 98% of haemoglobin exists in form of oxyhaemoglobin. Inside a vessel red cells tend to be uniformly allocated and, due to their membrane polarization, they remain separate but for low values of shear rate erythrocytes tend to accumulate and form structures called rouleaux. [3]

Figure 1.5: Rouleaux. [4]

White blood cells (**leukocytes**) are a mass of different spheroidal shape elements, they have a nucleus and are marginally less deformable than red blood cells since their cytoskeleton is developed throughout the cell volume. They defend the body and they are developed by the spleen, bone marrow and lymph nodes. There are three major groups of white blood cells: granulocytes, lymphocytes and monocytes, which, in normal conditions, are around 70, 25 and 5% respectively. The most frequent disease of white blood cells is the transformation into tumour cells of the blood (leukaemia). [3]

Platelets are spherical corpuscles with an average diameter of 3 µm. The main purpose of platelets is to keep haemostasis with the formation of platelet mass to repair vessels damages. The role of the platelets in coagulation process is divided in three steps: adhesion, secretion and aggregation. When a vessel is injured platelets are called to the breaking point and they adhere to the vessel wall (adhesion). After, platelets deliver (secretion) chemicals elements that they enclose, they have the aim to call other platelets that deposit upon the previous ones (aggregation) and release serotonin. This is necessary for reducing vessels diameter due to injure and slow down blood flow. [3]

Plasma is a slightly alkaline fluid composed of 90% of water, in which are dissolved organic matter and mineral particulates. The organic matters are composed of carbohydrate, lipids, glycoproteins, hormones, amino acid and vitamins. The mineral particulates are dissolved in positive and negative ionic elements. [3]

1.4.1 Blood Reology

Rheology is a discipline that analyses sliding properties of materials. Solid materials have elastic deformation according to Hooke's law; on the other side fluid and gases have Newtonian flow. A homogeneous fluid is described by two quantities: density and viscosity both correlated to the temperature.

Viscosity can be associated to the motion state. The motion state is expressed as speed vectors set of the fluid elements. Viscous stress between the fluid mass is a function of velocity

components changes between contiguous elements. The shear rate is the radial variation of axial velocity, indicated with γ̇.

A fluid is Newtonian if viscosity is shear rate independent and the tangential stress is directly proportional to γ̇ through dynamic viscosity (μ).

The blood is not a homogeneous fluid and has a complex nature therefor, it does not have a Newtonian behaviour but a complex one. However, in vessels with a diameter larger than 0.3mm, thinking about blood as a homogeneous and Newtonian fluid is a valid simplification. Since the diameter of an erythrocyte is at least two orders of grandeur smaller than the characteristic size of the duct, blood can be considered as a homogeneous fluid and at first approximation Newtonian until it reaches the arterioles.

There are cases where blood cannot be represented as Newtonian:

- Vessels with diameter greater than 0.3 mm and low shear rate;

- Vessels with diameter lower than 0.3 mm and high shear rate. [3]

1.4.2 Blood density

Blood density depends on the haematocrit and it can be considered both temperature and pressure independent. The formula that connects the haematocrit and the blood density is:

$$\rho_s = (1 - Ht)\,\rho_p + Ht\rho_{gr}$$

where:

- ρ_s is blood density (1060 Kg/m3 in normal conditions);

- ρ_p is plasma density (1035 Kg/m3 in normal conditions);

- ρ_{gr} is corpuscular elements density (1090 Kg/m3 in normal conditions);

- Ht is the haematocrit (45% in normal conditions). [3]

1.4.3 Blood viscosity

Viscosity can be associated to the internal stickiness of a fluid. It can be described in a one-dimensional flow as:

$$\tau = -\,\mu\frac{\partial v_x}{\partial y}$$

where $\frac{\partial v_x}{\partial y}$ is the shear rate γ̇, this expression is the differential form of Newton's formula and it is valid for a Newtonian fluid. Consequently, in this formula the shear rate and shear stress are directly proportional with viscosity as proportional coefficient.

The unit of measurement for viscosity in I.S. is [$\frac{N}{m^2}s$] or [Pa·s], but [cP] is more used where (1 cP = 10^{-3} Pa·s).

The comparison between the blood behaviour for different shear rate values and Newtonian fluid is showed by the figure below.
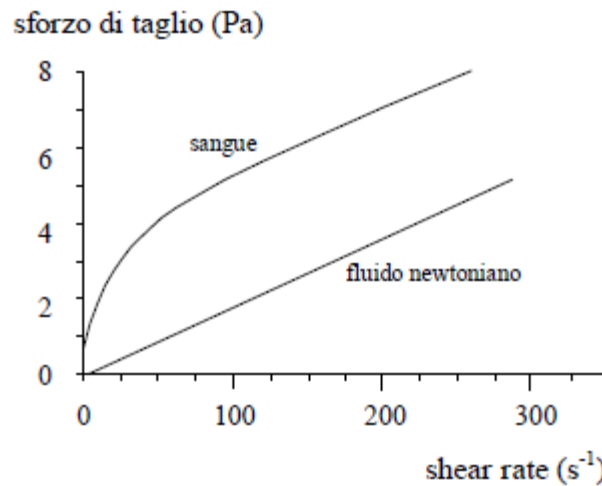
Chart 1.1: Blood behaviour. [3]

The blood has a non-Newtonian behaviour in case of shear rate lower than 100 s^{-1} but for shear rate higher than this value a Newtonian behaviour is a valid simplification. The rouleaux formation are responsible of the non-Newtonian behaviour and makes the blood sliding more difficult within the vessel.

The blood behaviour has been expressed with different non-Newtonian rheological models. A non-Newtonian fluid can be defined with the same formula of a Newtonian fluid, where the dynamic viscosity is replaced with apparent viscosity:

$$\tau = \mu_{app}\dot{\gamma}$$

where μ_{app} is not a constant and depends on the flow field. [3]

1.4.4 Rheology's models

Below a description of four different non-Newtonian models can be found:

- **Bingham:** is expressed by the following formula:

$$\tau = \tau_0 + \mu_B\dot{\gamma}$$

where τ_0 represents the yield stress, the minimum value of shear stress to permit fluid motion, and μ_B Bingham's viscosity.

This model is a good approximation for the asymptotic behaviour but is a simple approximation for the blood behaviour at low shear rate *(figure 1.7)*. [3]
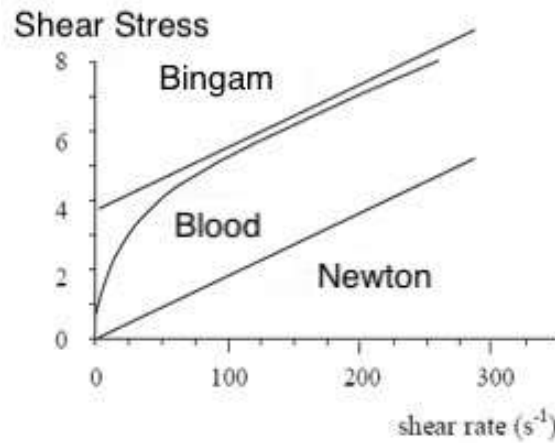
Chart 1.2: Bingham's model. [3]

- **Casson:** is expressed by the following formula:

$$\sqrt{\tau} = \sqrt{\tau_0} + \sqrt{\mu_c}\,\sqrt{\dot{\gamma}}$$

where τ_0 represents the yield stress and μ_c Casson's viscosity, which is a parameter of the fluid given by experimental data. [3]

- **Power-Law:** is expressed by the following formula:

$$\mu(\dot{\gamma}) = \text{k}(\dot{\gamma})^{n-1}$$

where *n* is the Power-Law's index which represents the existence of the non-Newtonian behaviour and k is the flow consistency factor. [3]

- **Carreau:** this model is very useful for expressing the shear thinning property of blood viscosity. Is expressed by the following formula:

$$\mu(\dot{\gamma}) = \mu_\infty + (\mu_0 - \mu_\infty)[1 + (\lambda\dot{\gamma})^2$$

where μ_0 and μ_∞ are the zero and infinite shear rate limit viscosities respectively and n the power-law index and λ is the relaxation time constant. [3]
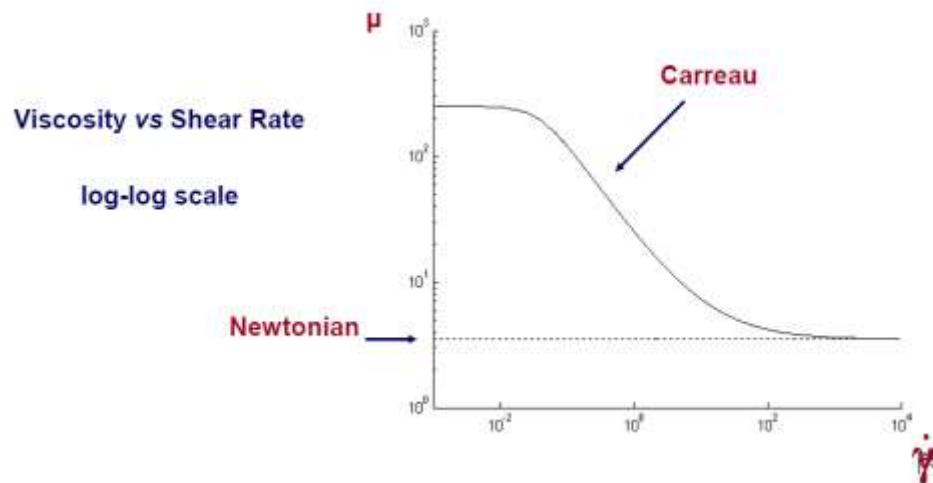
Chart 1.3: Carreau's model. [3]

11

1.5 The carotid bifurcation

The main quantity of blood to the head and neck region is brought by a big artery, called the **common carotid artery (CCA)**. On each side of the body a common carotid artery can be found diverging in their origin.

The **right CCA** ascends from the brachiocephalic artery posterior to the right sternoclavicular joint, while the **left CCA** starts from the aortic arch within the superior mediastinum. The CCAs rise lateral to the oesophagus and trachea within a profound cervical fascia, with the vagus nerve and the internal jugular vein.

In this paragraph are shown the anatomical variations and relations of the CCA as well as a resume of its structures and branches; after this a brief introduction to any clinical pathology that can affect these arteries. [5]
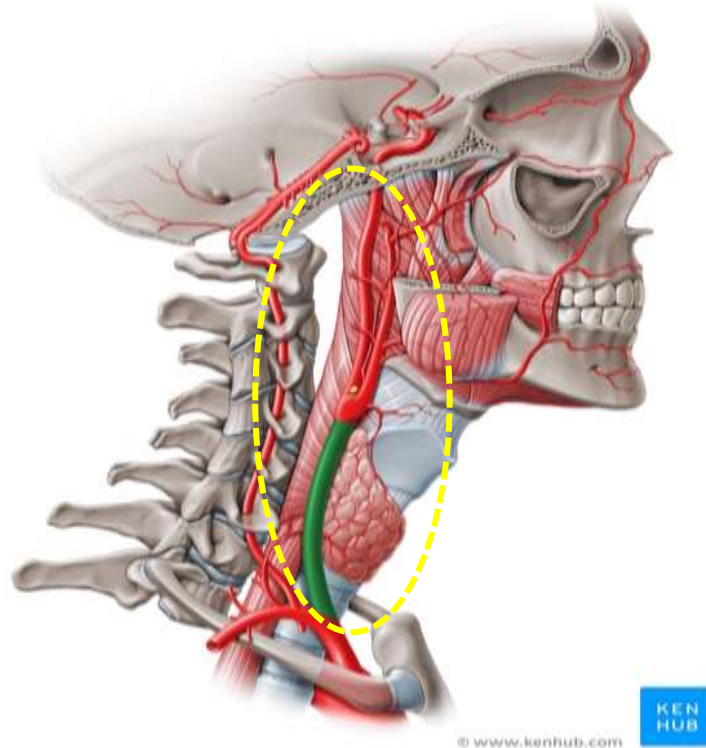
Figure 1.6: Common carotid artery. [5]

There are many structures in the neck, therefore it is important for clinical practice to evaluate the interactions of the CCA with other structures. Between the two CCAs, in the lower part of the neck, can be found the: larynx, thyroid gland, trachea and pharynx. Other structures related to the CCA are the **carotid sheath**, the CCA ascends within it, anterior to the vagus nerve and medial to the internal jugular vein. [5]

In its part under **the omohyoid muscle**, the CCA is under the: sternocleidomastoid muscles, sternothyroid, sternohyoid, platysma, deep cervical fascia, superficial fascia and, finally, skin. Close to the **cricoid cartilage**, the intermediate tendon of the omohyoid muscle

anterolaterally crosses the CCA. From this level the artery location is more superficial, on it there are only the sternocleidomastoid, the platysma, fascia layers and the skin.

Over **the omohyoid muscle** the CCA is intersected by the superior thyroid artery, in particular by its sternocleidomastoid branch. Inside the **carotid triangle**, the CCA is under the **anca cervicalis**, a loop of nerves coming from the cervical plexus. [5]
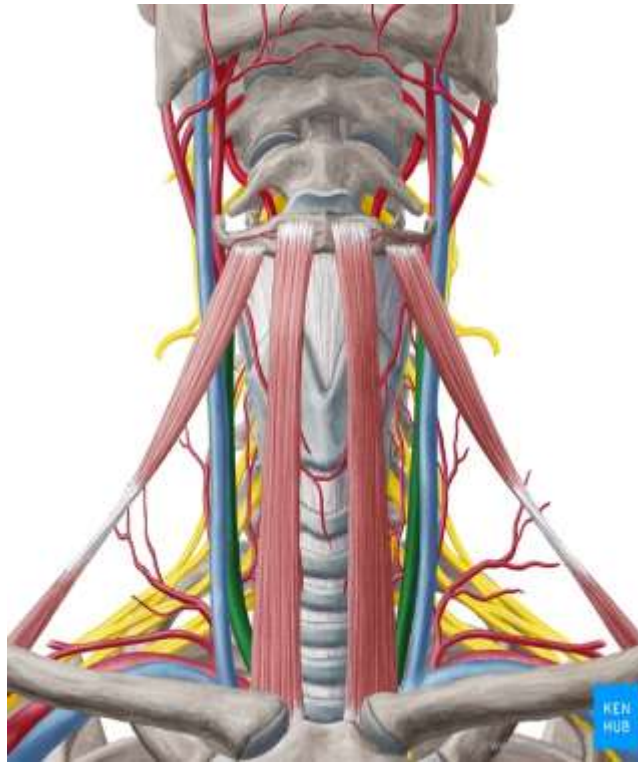
Figure 1.7: Common carotid artery - ventral view. [5]

The CCA is superficially crossed by three veins at different levels: the superior and middle thyroid veins respectively at the upper part of the thyroid cartilage and at the cricoid cartilage level, while the anterior jugular above the clavicle. Posterior to the CCA lie the ascending cervical branch of the inferior thyroid artery and the sympathetic trunk. Under C6's level, the longus colli and the scalenus anterior muscles include between them the artery that is anterior to the sympathetic trunk, vertebral vessels, inferior thyroid artery and subclavian artery.

Anatomical Variations: the origin of the left common carotid artery can change and ascend with the brachiocephalic artery. In 12 % of cases, the right common carotid artery ascends above the level of the sternoclavicular joint. Sporadically, at a higher level near the hyoid bone the common carotid artery splits. More infrequently, it splits lower than usual at the level of the larynx. In some cases, the external and internal carotid do not exist, because the common carotid artery does not split. Typically, the common carotid artery discharges no other branches, apart from these two terminal branches. Nevertheless, it can rarely give increase to superior thyroid, inferior thyroid, vertebral, occipital, rising pharyngeal or superior laryngeal arteries. At the level of the superior border of the laryngeal thyroid cartilage, the artery splits into two terminal branches, which are defined below. [5]
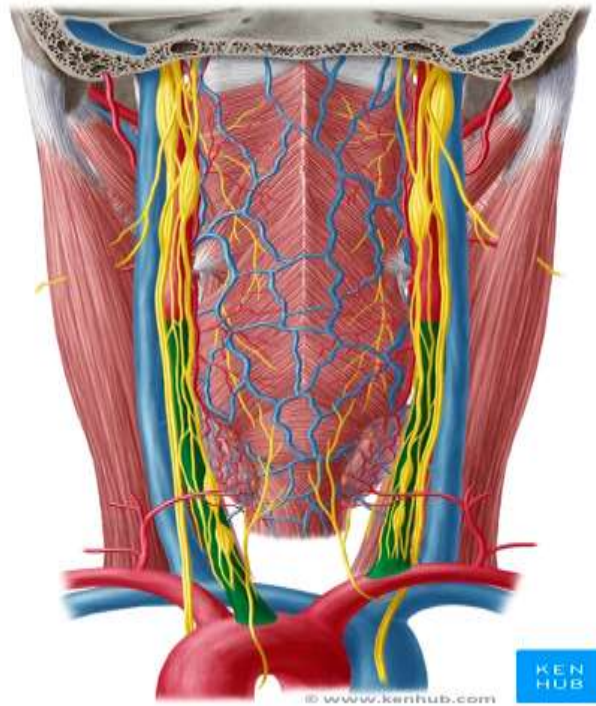
Figure 1.8: Common carotid artery - dorsal view. [5]

The **external carotid artery** ascends at the level of the intervertebral disc, between C3 and 4, and arises slightly anteriorly before inclining poster-laterally. In the **carotid triangle**, it is anteromedial to the internal carotid artery. Eight main branches divide the external carotid, which stock regions of the head and neck. [5] [6]
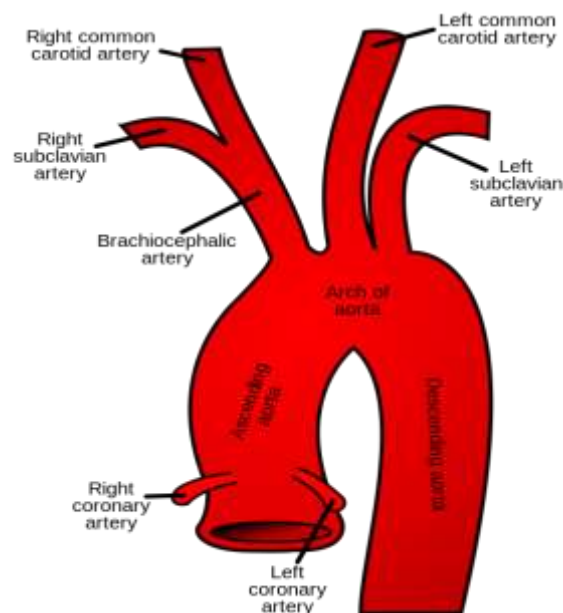
Figure 1.9: Aortic arch. [6]

The **internal carotid artery** goes from its origin at the **carotid bifurcation** to the anterior perforated substance. [5] Near its bifurcation, the common carotid artery forms two dedicated structures, which are defined below.

An expansion of the base of the internal carotid artery is the **carotid sinus,** which is correlated in giving information about the arterial blood pressure to the hypothalamus. Indeed, it is innervated by the carotid branch of the **glossopharyngeal nerve** since is a **baroreceptor.** [5]

The transmitting information about the arterial chemical composition to **respiratory centres** in the brainstem are committed by the **carotid body**, which is an oval structure situated posterior to the carotid bifurcation.

The **glossopharyngeal nerve** innervates the carotid branch, like the carotid sinus. The carotid body is enclosed by a fibrous capsule and consists of multiple lobules separated by septa. Within each lobule, there are two types of cells: glomus (type I) cells and sustentacular (type II) cells. Neurotensin, amines, adrenaline, noradrenaline and dopamine are storing peptides engaged by **glomus cells.** The **sustentacular cells** disconnect the glomus cells from a wide network of fenestrated sinusoids. Hypercapnia, hypoxia and amplified hydrogen ion concentration (low pH) stimulate the carotid body, which is a **chemoreceptor**. The rate and volume of respiration are change by the carotid body due to a reflex concerning the respiratory centres in the brainstem. [5]

1.6 The Carotid bifurcation atherosclerosis

The common carotid artery is a favourable site for the development of **atherosclerosis**, a degenerative arterial illness resulting in the creation of plaques. The lumen is constricted by a big plaque deposited inside the inner wall of the artery; it is composed of calcium, fat, cholesterol and other substances of the blood. Generally, normal healthy arteries have smooth inner walls and are flexible. The creation of a plaque is caused by hypertension and small injuries of the blood vessel. Usually, the process called "hardening of the arteries" takes origin from atherosclerosis, it implies that arteries become rigid. Due to this is possible a narrowing of common carotid artery, as known stenosis, which augments the risk of a stroke or a transient ischaemic attack (TIA). It is possible to treat it with pharmacological administration, such as warfarin or aspirin, or surgical management. Treatment can include pharmacological administration, such as aspirin or warfarin, or surgical management. It is suggested using surgical intervention, as a **carotid endarterectomy**, for people with a stenosis greater than 50%, which is a removal of atherosclerotic plaque material within the artery. [7] [8]
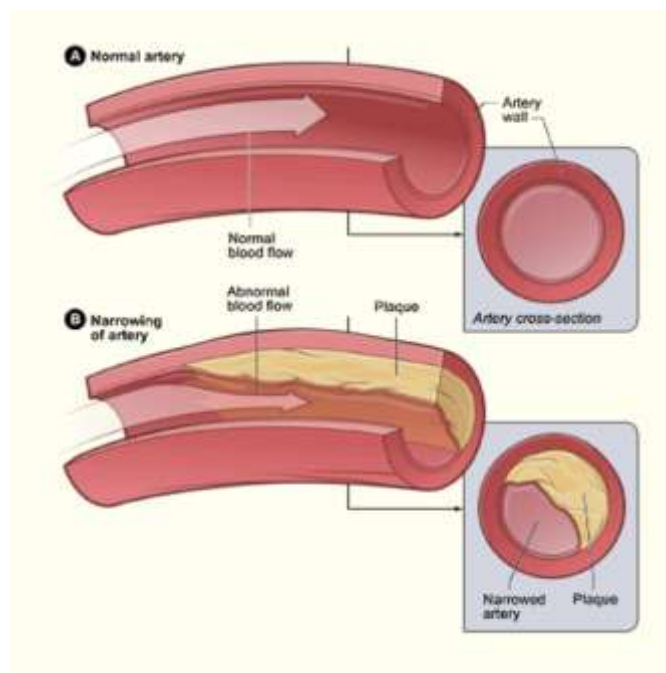
Figure 1.10: Cross section of (A) a normal artery with normal blood flow and (B) an artery with plaque. [7]

The risk of stroke augments depending on three possible ways of carotid stenosis:

- Blood flow to the brain is reduced by a progressive narrowing of the artery, since plaque amasses can raise larger. Plaque can occlude completely the artery (Fig. 1.11 A);

- Blood clots can be forming and blocking blood flow to the brain, because plaque amasses can roughen and deform the artery wall (Fig 1.11B);

- Plaque amasses can break and escape, traveling downstream to lodge in a smaller artery and brick blood flow to the brain. [8]
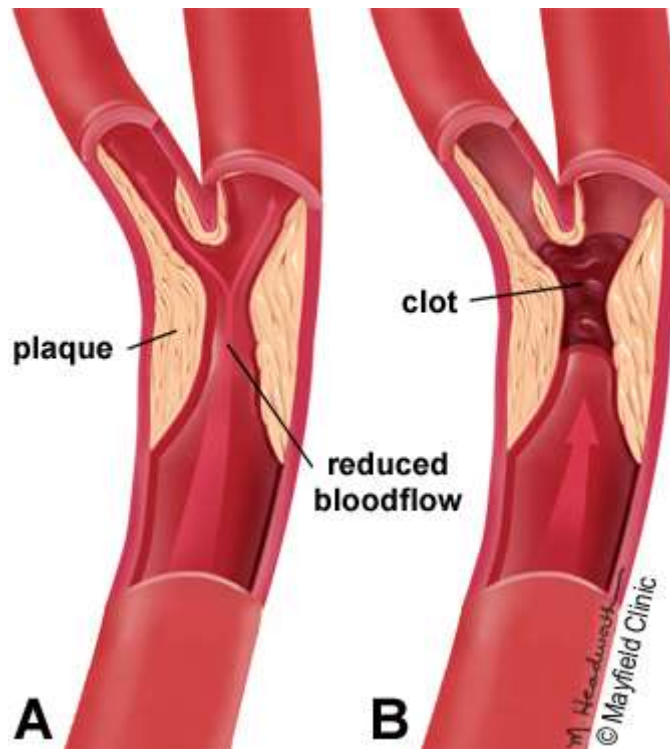
Figure 1.11: (A) Occlusion due to plaque and (B) clot formation. [8]

There are different possible risk factors for atherosclerosis, divided in reversible and irreversible factors.

Irreversible factors:

- Age;
- Sex;
- Previous diseases (diabetes or impaired glucose tolerance);
- Genetic abnormalities (hypertension, hypercholesterolemia, obesity). [3]

Reversible factors:

- Hypertension (unbalanced diet);
- Tobacco smoking;
- Stress;
- Obesity (unbalanced diet);
- Hypercholesterolemia (unbalanced diet);
- Using oral contraceptives. [3]

1.7 Atherosclerosis diagnosis and surgery

Doctor must learn symptoms, current medications, current and previous medical problems, and family history. The doctor uses a stethoscope for listening the carotid artery and should notice a whistling noise called "bruit", which may be caused by a turbulent blood flow produced by atherosclerosis. One or more diagnostic tests are completed to notice narrowing of the carotid arteries. Carotid stenosis is identified by either a magnetic resonance angiography (MRA), a CT angiogram (CTA) of the neck, a cerebral angiogram or a doppler ultrasound of the neck.

Multiple small strokes can be deducted with imaging. Doctors can make a diagnosis of carotid stenosis if tests show reduced blood flow in one or both carotid arteries. Below the description of the different techniques:

- **Doppler ultrasound** is a non-invasive test able to estimate blood flow through a vessel. The ultrasound probe is positioned over the carotid arteries on the neck. It reveals how much blood is flowing through the artery and the artery narrowing degree;

- **Computed Tomography Angiography** is a non-invasive X-ray used to have a complete image of anatomical structures within the brain. It is necessary using a contrast agent with the inoculation into the blood stream, this permits to show the arteries of the brain. It gives doctors the possibility to see the narrowed artery and to evaluate the degree of narrowing, offers the best pictures of blood vessel and soft tissues.

- **Magnetic Resonance Angiography** (MRA) of the neck is used with a contrast dye inoculated through an IV to brighten blood vessels in the neck, comparable to the CT angiogram.

- **Cerebral Angiogram** is a slightly invasive test using X-rays and a contrast agent inoculated into the arteries through a catheter in the groin, that permits doctors to visualize all arteries in the brain.

There are different possible treatments to reduce the risk of stroke, the choice depends on the gravity of the arterial narrowing and whether the subject is experiencing stroke-like symptoms or not (asymptomatic). People with low-grade carotid stenosis (less than 50%) or without symptoms are typically treated with medications. The same solution is also suitable for people whose medical condition would increase the risk for the patient.

There are different medications:

- It is possible, for allowing the blood to flow more simply, to use **antiplatelet** medications (clopidogrel, ticlopidine, aspirin), which thin the blood and avoid clotting in the narrowed arteries.

- Since high blood pressure is a major risk of stroke, using **antihypertensive** medications (angiotensin blockers, beta blockers, ACE inhibitors, calcium channel blockers, diuretics etc.) helps control and adjust blood pressure.

- The possibility of plaque creation in atherosclerosis can decrease using **cholesterol-lowering statins.** A low-fat and low-cholesterol diet mixed with statins can decrease LDL "bad" cholesterol by an average of 25-30%.

Patients who have suffered one or more TIAs or strokes and who have a modest to high-grade carotid stenosis are usually treated with surgical treatment. The aim of surgery is to avoid stroke by eliminating or reducing the plaque accumulation and expanding the artery lumen to permit blood to flow more naturally to the brain.

- **Carotid endarterectomy** is an open surgery intended to eliminate the plaque. In the neck a skin cut is made and the carotid artery is positioned. Above and below the area of stenosis, to stop blood flow, are positioned momentary clamps. Throughout, blood flow to the brain is brought by the carotid artery on the other side of the neck. Over the fixed area, the surgeon makes a cut in the artery and eliminates the plaque. Then, the artery is closed with small sutures and the clamps detached to permit blood flow to the brain. Patients who have had symptoms (stroke or TIA) and have an obstruction greater than 50% are classically designated for carotid endarterectomy. It is also suggested for patients who have no symptoms and have an obstruction greater than 60%. Surgery decreases the risk of stroke by 6.5% over a five-year period for patients with moderate obstruction of 50 to 69%. The risk of stroke is reduced by 80% for patients with high-grade blockage of more than 70%. Patients whose stenosis is 50% or less does not compensate the risks of the surgery.
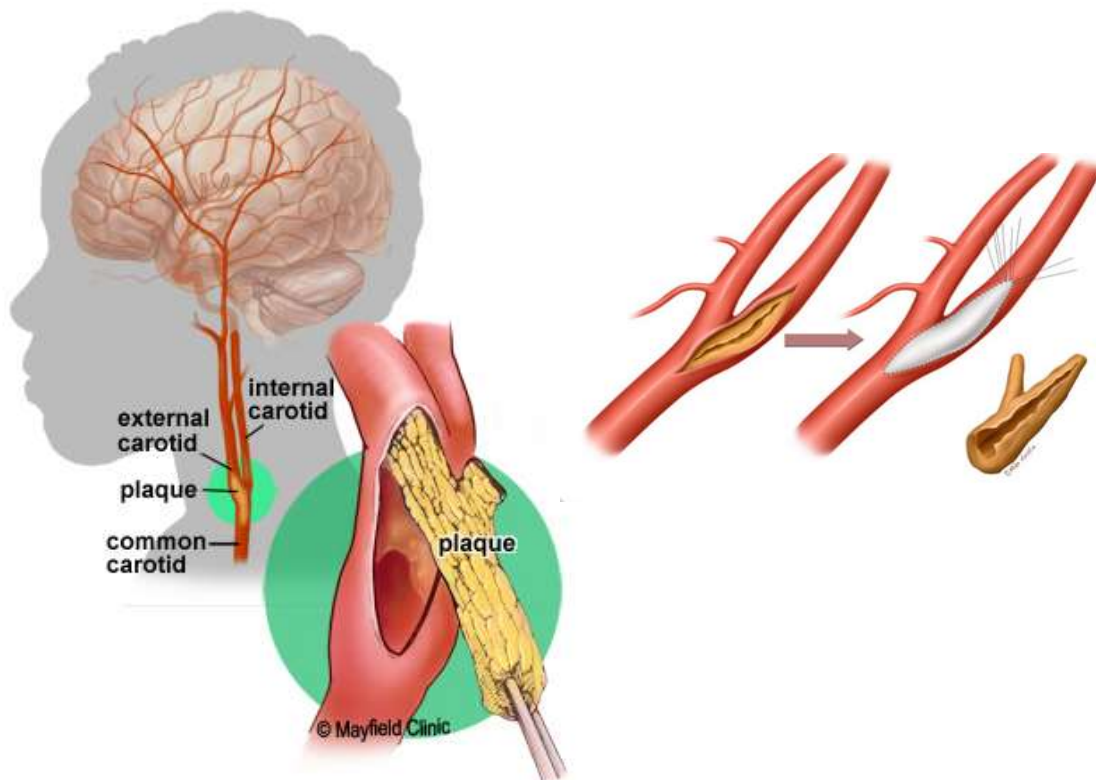
Figure 1.12: Carotid endarterectomy. [8]

- **Carotid angioplasty / stenting** is a low-invasive endovascular technique that enlarges the lumen of the artery and compresses the plaque. It is made during an angiogram in a radiology suite. A flexible catheter is guided to the position of the plaque within the carotid artery, it is introduced in the femoral artery in the groin. After, a small catheter, having an expandable balloon at the tip, is placed across the plaque and it is opened to expand the artery and push the plaque against the artery wall. At the end, the balloon is emptied and detached. In conclusion, a stent is positioned over the plaque, which is a self-expanding tube. This technique is typically designated for patients: who have other medical conditions that increase the risk of complications during surgery, with a moderate to high-grade carotid stenosis greater than 70%, with a recurrent stenosis or who have stenosis that was produced by prior radiation therapy.
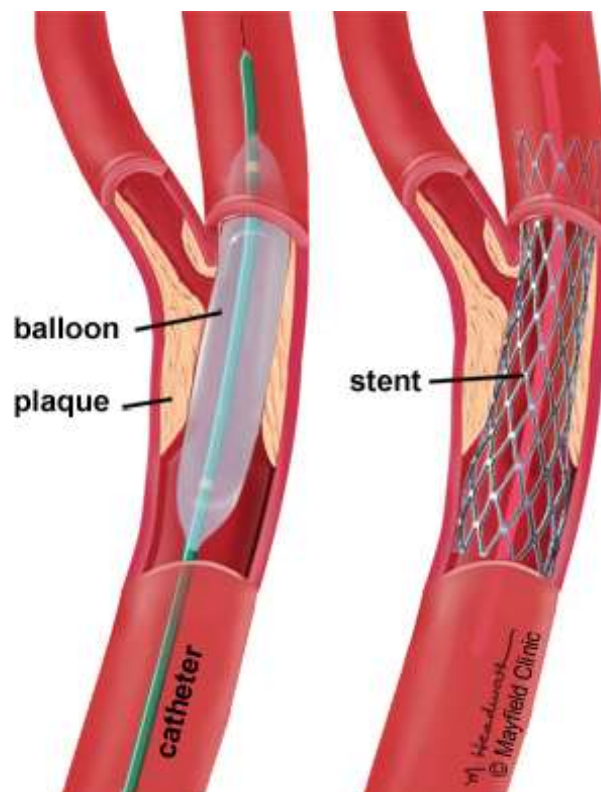
Figure 1.13: Carotid angioplasty / stenting. [8]

- **Carotid artery bypass** is used to redirect the blood stream around the plaque-blocked area, where a length of artery or vein is taken from somewhere else in the body, typically the ulnar or radial arteries in the arm or the saphenous vein in the leg. Blood flow is redirected (bypassed) through the graft, since the vessel graft is linked above and below the obstruction. Bypass is typically only used when the carotid is completely occluded.

This thesis work will focus on the comparison between two surgery techniques, it aims to compare patch graft (PG) and carotid artery stenting (CAS). Carotid endarterectomy (CEA) is the most used surgical treatment to re-establish the patency of extracranial district and

reduce the risk of stroke of an atherosclerotic plaque narrowing the carotid artery. PG is used to restore the original shape of carotid bifurcation and it is not used to expand the site of CEA. PG should diminish the hemodynamic consequences of restenosis. However, PG surgery uses a double suture line that augments carotid fixing time and increases the risk of cerebral ischemia and disturb to the patient in case of locoregional anaesthesia. Furthermore, the insertion of alloplastic material increases blood losses and the possibility of different complications, like infections and pseudoaneurysms, and the longer time for carotid cross-fixing could augment the number of neurocognitive deficits. In many textbooks of vascular surgery, like the American Society of Vascular Surgery, and in the standard use in the guidelines of the European is showed that PG decreased 30-day and long-term risk of stroke. A later update had less robust result, but in recent studies reported no higher complication rates neither wickedest long term clinical after using PG. Surgeons have individual preferences despites recommendations supporting use of PG. This due to doubts of some authors, especially when are interested smaller arteries to CEA or in case of related repair of a distal ICA with coiling or kinking. [9]
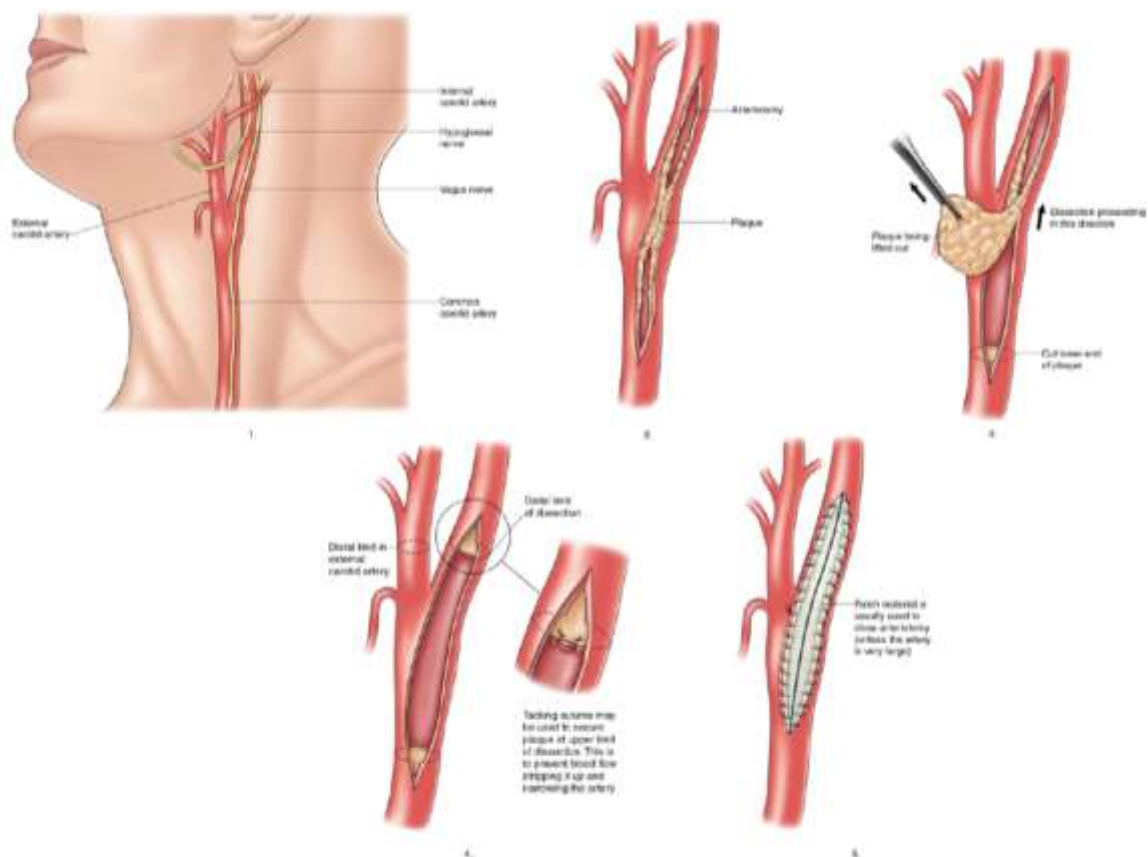
Figure 1.14: Patch graft surgery. [8]

CAS is an alternative to CEA in normal and high-risk surgical patients. It is usually recommended for the treatment of patients under 70 years. The occurrence of restenosis

arises in general in the first year after CAS. Restenosis was correlated with an augmented risk of ipsilateral stroke. The tissue damages, result of restenosis after stenting, are distinguished primarily by neointimal hyperplasia in one year of stenting. Computational fluid dynamic is useful in predicting haemodynamic differences before and after surgery and is necessary to qualitatively evaluate the risk of carotid stenosis or restenosis. [10]
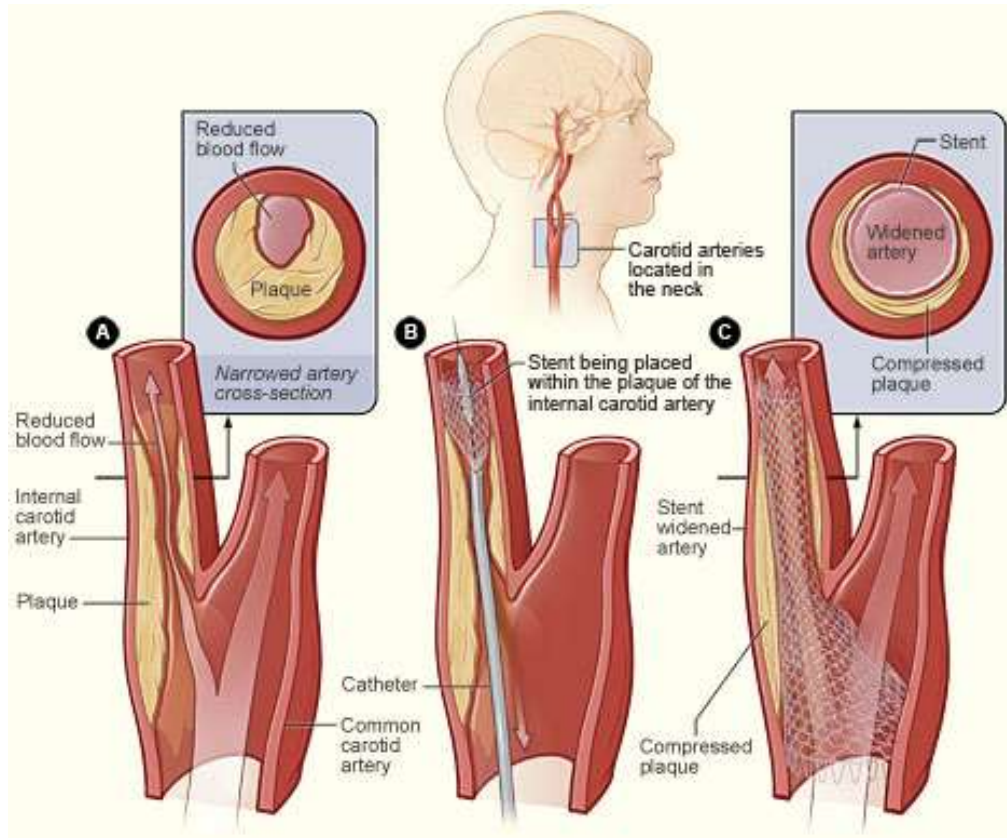
Figure 1.15: Carotid artery stenting surgery. [11]

Chapter 2
Morphometric characterization

The aim of this chapter is to explain the role of geometry in the cardiovascular system.

Its parts are: the role of geometry, the relationship between hemodynamic and geometry in carotid bifurcation, aim of the work, dataset and reconstruction of the models, Bijari's geometric parameters.

The paragraph concerning Bijari's Geometric parameters is divided in: description of the parameters and model processing. This chapter gives an explanation about the important role of geometry, focusing on the carotid bifurcation. It describes the aim of the work, how from the dataset every model was reconstructed and how every parameter proposed by Bijari et al. was processed.

2.1 The role of Geometry

There is an important theory concerning the role of geometry as a key influence factor for pathology. The geometry of certain kinds of vessel regions can cause the pathology of the wall because it alters the local hemodynamic and shapes of flow.

Every fluid mechanical event that is linked to the pathogenesis of atherosclerosis is itself prejudiced by different geometric features of the arteries in which they arise. The tortuosity or radius of curvature at turns, size of vessels, the angles between branches in a bifurcation and other comparable geometric parameters undoubtedly have important effects on hemodynamic and could even affect whether some phenomena, such as separation of flow or flow reversal at the wall. For example, if some part of an individual's vasculature has a geometry that aggravated the hemodynamic stress, that person might be exposed at a bigger risk for atherosclerosis in that zone. Certainly, this scenario creates the foundation for the notion of "geometric risk factor", that is a risk value of some geometric variable or an uncommon geometric feature of a vulnerable arterial section that improves an atherogenic hemodynamic stress. [12]

Those reflections led to the definition of the "Geometric Risk Hypothesis":

"Recognizing that the hemodynamic variables of flow are in turn mediated by the geometry of the arterial lumen, the study was undertaken to explore the hypothesis that some of the variability among individuals and populations in the location and rate of development of atherosclerosis is due to corresponding variations in the detailed geometry of the susceptible arterial segments". [13]

Numerous examples to emphasize the importance of vessel's geometry are explained in the paragraphs below.

Arterial bifurcation:

The presence of a bifurcation effects and modifies the velocity profile. Indeed, the flow layers cannot follow the curvature of the vessel and this generates a separation region in which low speed recirculation phenomena can be found. This happens both close to the inner and external walls of the vessel. The result is the decrease of vessel's section available for blood flow. The extension of these zones depends on the inlet flow rate (depending on the heart rate). If there is not enough space for flow's development this condition can generate a "disturbed flow" condition in the vessel.

This modification of the vessel's shape conducts also to "entrance effects", for example flow in the ascending aorta is probably not completely developed, because it is influenced by the profile as it leaves the ventricle, or, flow in a daughter artery just after a bifurcation is predisposed to the separation of flow in parent vessel. The entrance effects continue for a distance proportional to Reynolds number and the radius of the artery.
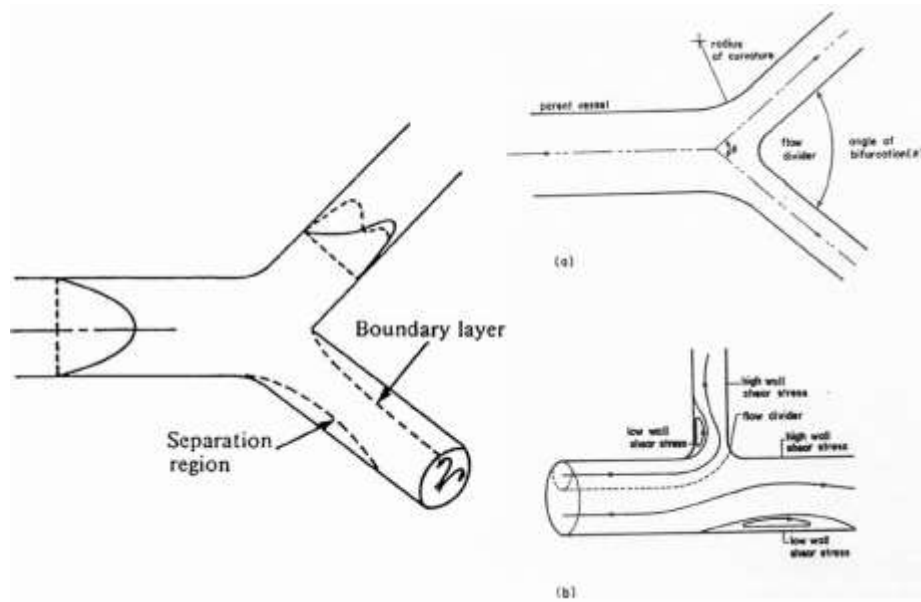
Figure 2.1: Flow past arterial bifurcation and branching sites. [3]

Arterial curvature:

Another important variable, that influences the geometry and consequently flow's behaviour is the curvature of the vessel. Indeed, most arteries are curved, and this has an important consequence on the flow structure. Due to this, the main motion on the centerline, the locus of centres of cross-sections of the pipe, is related to a motion on the transverse cross-section direction of the vessel.

This secondary flow is generated by the difference between centrifugal force and pressure force, and its intensity is under the influence of the curvature radius and the flow rate. Secondary flow are described by Dean's number:

$$Dn = Re \sqrt{\frac{D}{R}}$$

where Re is Reynolds' number, D is hydraulic diameter and R is the curvature radius.

At the beginning, a steady pressure gradient directs flow down the tube and it is mainly located in the longitudinal cross-section direction almost as a Poiseuille profile. On the other hand, the centrifugal force pushes the flow layers and moves the velocity profile towards the external side of the curve. This generates a radial pressure gradient that makes the flow move from the external toward the internal side of the curve. This is called secondary flow and looks like two counter-rotating vortices. The balance condition is obtained when there is equilibrium between centrifugal, pressure and viscous forces.
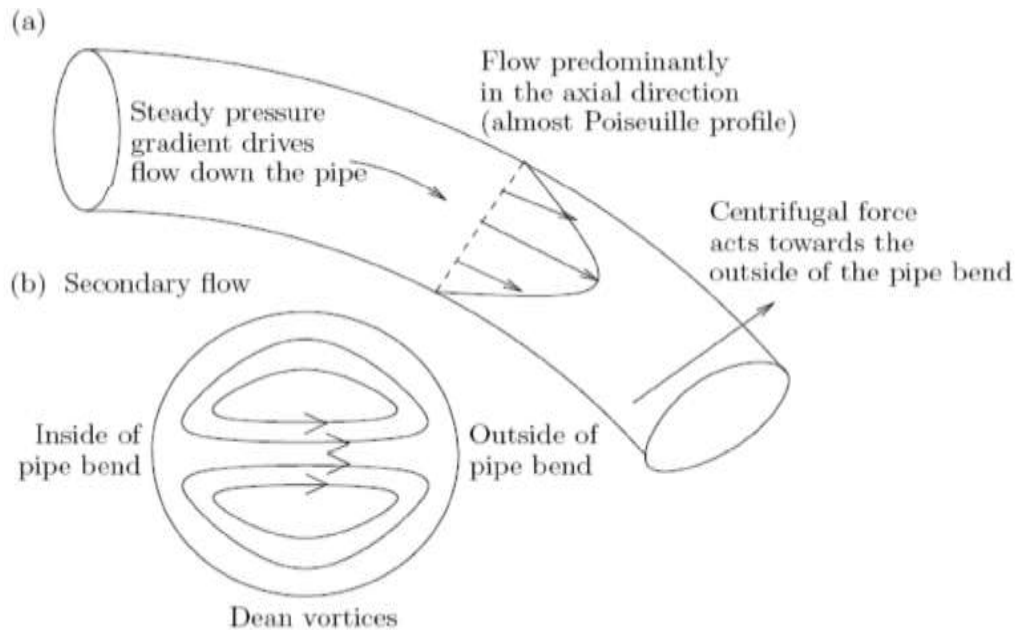
Figure 2.2: Dean flow. (A) Longitudinal cross-section, showing Poiseuille-like profile, (B) transverse cross-section showing secondary flow, which forms Dean vortices. [3]

Arterial taper and elasticity:

Due to the elasticity in the arterial walls, the cross-sectional area modifies in reply to a pressure variation. Consequentially the cross-section along the tube is not uniform. Arterial taper is common in most arteries, it changes the entrance effects of the flow and generates also reflection of the hemodynamic waveforms. The backward and forward velocity profiles overlap and can overload the heart, sometimes this represents a pathological event.

The connection between arterial fluid and solid mechanics can be summarized in the list below:

- Unsteady flow in straight arteries;

- Unsteady flow in curved arteries - secondary flow;

- Branches and bifurcation: flow disturbance;

- Mechanics of the viscoelastic arterial wall;

- Blood is a non-Newtonian fluid with complex rheological behaviour.
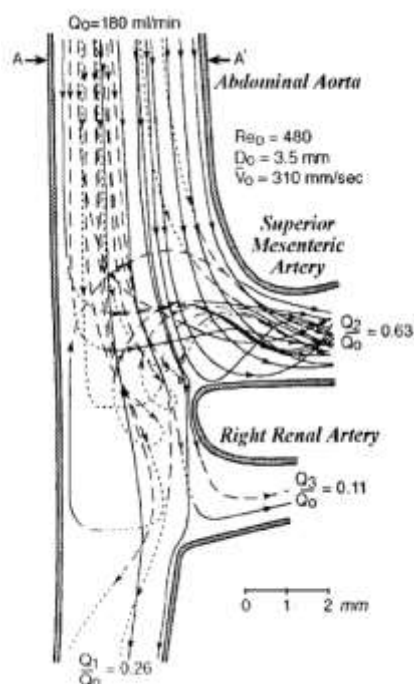
Figure 2.3: Arterial fluid and solid mechanics. [3]

2.2 The relationship between hemodynamic and geometry in carotid bifurcation

All those events described before, concerning different geometry variables have an important role in arterial flow disturbances and its influence on the treatment of vascular diseases. This is extremely evident in the carotid bifurcation since atherosclerosis develops primarily near curves and in the major branches of the arterial network (DeBakey *et al.*,1985). Hence, hemodynamic forces have a crucial role in the development of atherosclerosis. [14] Many in vitro and in vivo studies have verified that these disturbed-flow patterns induce oscillatory and low values of wall shear stress in the areas where they develop, and this can cause an endothelial-cell dysfunction (Slager *et al.*,2005).

In the carotid bifurcation arterial blood from the common carotid artery (CCA) is sent to the extracranial and intracranial circulations via the external (ECA) and internal (ICA) carotid artery branches respectively. The existence of this bifurcation and specially the expansion of the vessel produces complex intravascular flow structures and generates flow separation and recirculation. Consequently, this area presents low and oscillatory shear stress.

Early CFD studies using idealised geometries of the bifurcation confirmed the impact of bifurcation angle and bulb diameter on the intensity of disturbed shear stress and on the size of the affected zone at the carotid bulb. Later, a more systematic method, using a parametric geometry, showed a link between the bifurcation angle, the bulb diameter, and the branches diameters and the prediction of low shear stress. Researches using patient-specific geometries presented that bifurcation planarity only had a minor effect on disturbed shear stress related to artificially straightened counterparts. [15]

2.3 Aim of the work

There is an important correlation between geometry and hemodynamic as Lee *et al.* (2008) and Chiastra et al (2017) recently demonstrated. Since geometric variables are used to define the shape of an artery, they can also be suboptimal predictors of disturbed flow. Lee *et al.* have demonstrated that, at least in normal carotid bifurcations, the vessels sections and tortuosity can predict disturbed flow while branch angle or planarity cannot. Later studies showed that modest redefinitions of geometric variables, combined with a premeditated consideration of the real hemodynamic they are planning to predict, can bring improvement of predictions without a loss of reproducibility.

This thesis will use as reference the article: *"Improved prediction of disturbed flow via hemodynamically-inspired geometric variables" from* Bijari *et al.* 2012. It shows six geometric variables that have a correlation with the vessel hemodynamic and the opportunity to predict the entity of disturbed flow.

The variables will be calculated on a dataset to prove their outcomes in terms of relationship between hemodynamic and geometry. The main purpose is attempting to establish the efficiency of the surgery using Patch/Stent for re-establishing a not pathological hemodynamic state and defending against the atherosclerotic plaque development after surgery, this will be done analysing the geometric parameters and relating them to the CFD patient specific results.

2.4 Data set and reconstruction of models

The data set is made of four patients, their CT images were taken in Larissa University Hospital (Cases: 1, 3, 4) or in Euromedica Medical Diagnostic Centre of Larissa.

Basic patients' information is listed below.

CASE	SEX	AGE	TYPE OF TREATMENT	SCAN
1	M	76	Stent	Toshiba Aquillion
2	M	64	Stent	Siemens Somatom
3	F	81	Patch	Toshiba Aquillion
4	M	73	Patch	Toshiba Aquillion

Table 2.1: Patients' information.

Every model was reconstructed from the CT images using ITK-SNAP and then was smoothed with Materialise Mimics.

Below images show carotid bifurcations of the patients before and after surgery.
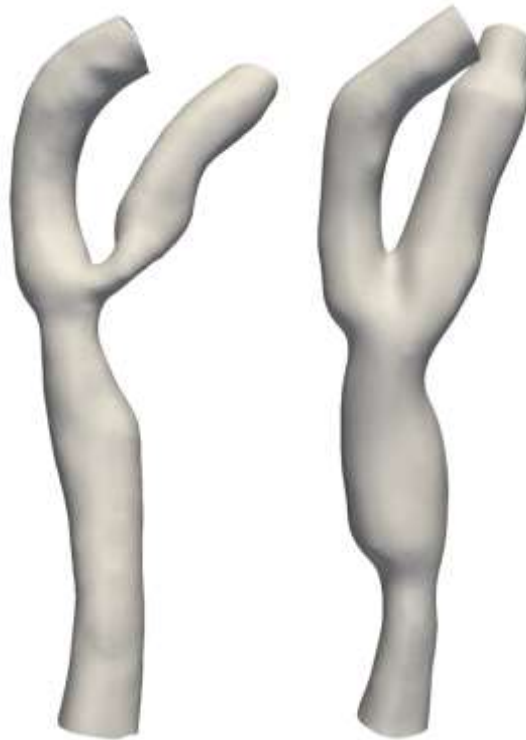
Figure 2.4: Carotid bifurcations for stent-treated patients. Before and after surgery respectively on the left and on the right. (Case1)
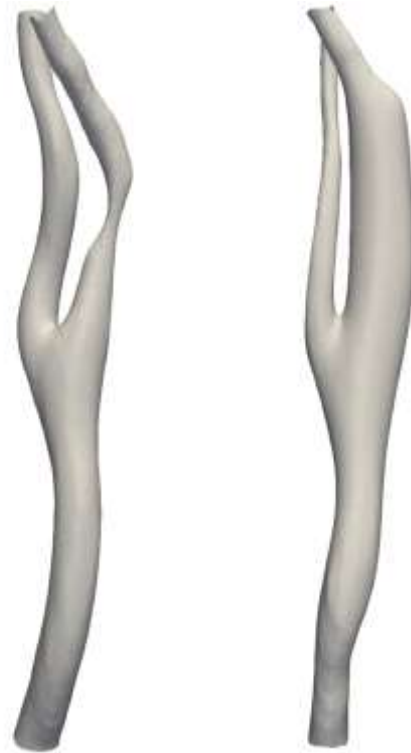
Figure 2.5: Carotid bifurcations for stent-treated patients. Before and after surgery respectively on the left and on the right. (Case2)
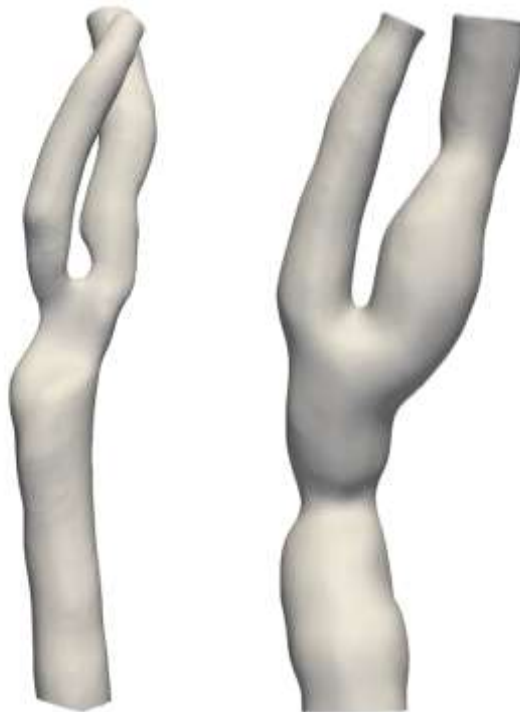
Figure 2.6: Carotid bifurcations for patch-treated patients. Before and after surgery respectively on the left and on the right. (Case3)
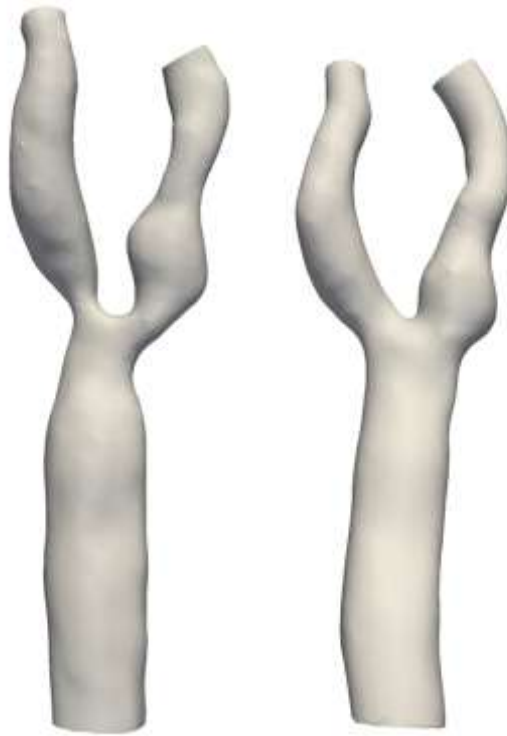
Figure 2.7: Carotid bifurcations for patch-treated patients. Before and after surgery respectively on the left and on the right. (Cas4)

2.5 Bijari's geometric parameters

After digitally reconstructed, each 3D lumen was analysed with a fully automated process that returns the geometric characterization. The centreline of a vessel is the succession of all the centres of the maximum radius inscribed spheres and is calculated, according to its definition, from the CCA to both the ICA and ECA branches. This centreline and their related sphere radii were then used to find the nominal plane of the bifurcation and the origin, this allows to divide the vessel into its three different branches; it is also possible to define a vessel-specific coordinate system. Naturally, a metric distance along the centreline is defined and is related to the maximum inscribed spheres, ensuring to describe objective geometric parameters for bifurcation having different sizes and shapes. [16]

2.5.1 Description of parameters

Traditionally considered as the ratio between the cross-sectional areas of the daughter branches and that of the parent artery, the expansion, or flare, of a bifurcation was confined from Bijari *et al.* proximal to the branching. [17]

Flare A: the ratio between the maximum CCA cross-sectional area proximal to the bifurcation flow divider (CCAmax) and the cross-sectional area calculated in a point three sphere radius distal (CCA$_3$). [17]

$$\frac{A_{CCAmax}}{A_{CCA3}}$$

Flare M: since the maximum flare usually occurs in the plane of the bifurcation, a second flare variable (FlareM) was defined as the ratio between the major axes of the CCAmax and CCA$_3$ areas. [17]

$$\frac{D_{CCAmax}}{D_{CCA3}}$$

Flare R: since the rate of flaring from locations CCA$_3$ to CCAmax might be a factor in promoting flow separation FlareR was defined as half the difference between the two major axes divided by the distance between the two planes:

$$\frac{(D_{max} - D_{min})}{2L}$$

the mean slope of the flare. [17]
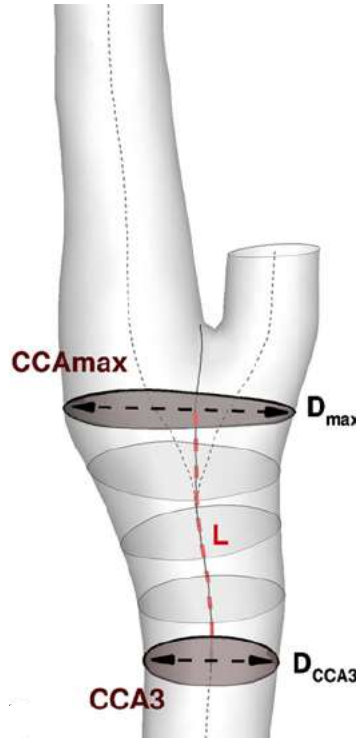
Figure 2.8: The thin solid line is the average of the (dotted) ICA and ECA centerlines; the thicker (red) dashed lines identify the major axes used to compute FlareM, and the distance between planes for compute FlareR. [17]

Tort 3D: it was noted that the CCA-ICA tract is typically shaped as a sigmoid with an inflection point close to the flow divider, in a region where flow separation typically occurs. Tort3D is defined using the 3D centerline from CCA_3 to this inflection point, which was identified as the closest-to-the-flow-divider location of the maximum centerline torsion. [17]

$$\frac{L}{D} - 1$$

Tort 2D: since secondary flows are driven primarily by planar curvature, a second tortuosity variable is defined (Tort2D) in which L and D were based on a projection of the 3D centerline onto a best fit plane between CCA_3 and the inflection point. [17]

$$\frac{L_{proj}}{D_{proj}} - 1$$

Curv 2D: since curvature itself, rather than tortuosity, is more directly linked to the production of secondary flow, a third variable (Curv2D) was defined as the curvature of a fitted circumference to the projected centreline, always between CCA_3 and the inflection point. [17]

$$\frac{1}{R}$$
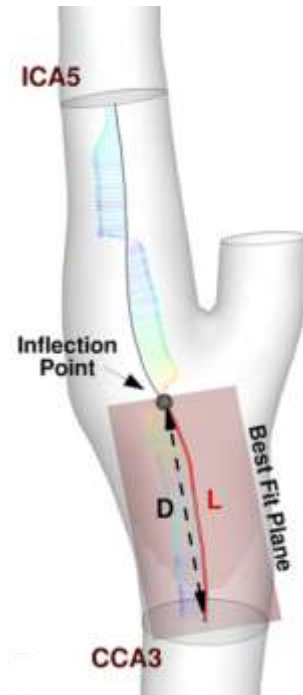
Figure 2.9: The pivot is identified as the point of maximum rate of change of the Frenet normals (vector direction), or equivalently the maximum torsion (vector color). The thick (red) line identifies the centreline segment used as the basis for the Tort parameters. [17]

Since the definition of Flare parameters given from Bijari *et al*. in [17] was meant to be used for healthy carotids and led to improper evaluation of the parameters for the preoperative models, a redefinition of the Flare parameters has been proposed.

To do so, we analysed the CCA sections areas along the mean-centerline from the CCA_3 section to the flow-divider. The 'Abscissa' parameter is the distance of each section from to CCA_3.
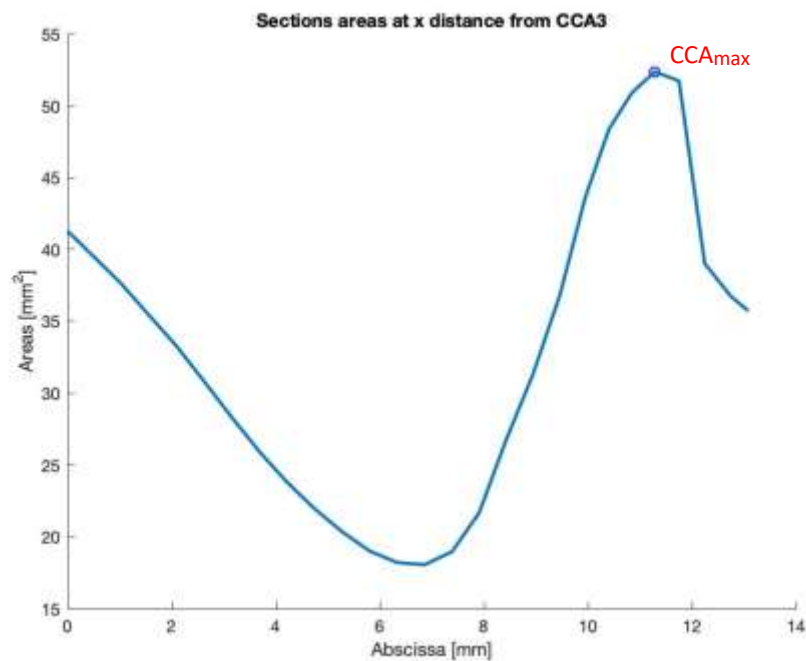
Chart 2.1: Areas along the mean centreline for different values of abscissa for Case 1 pre-operative.
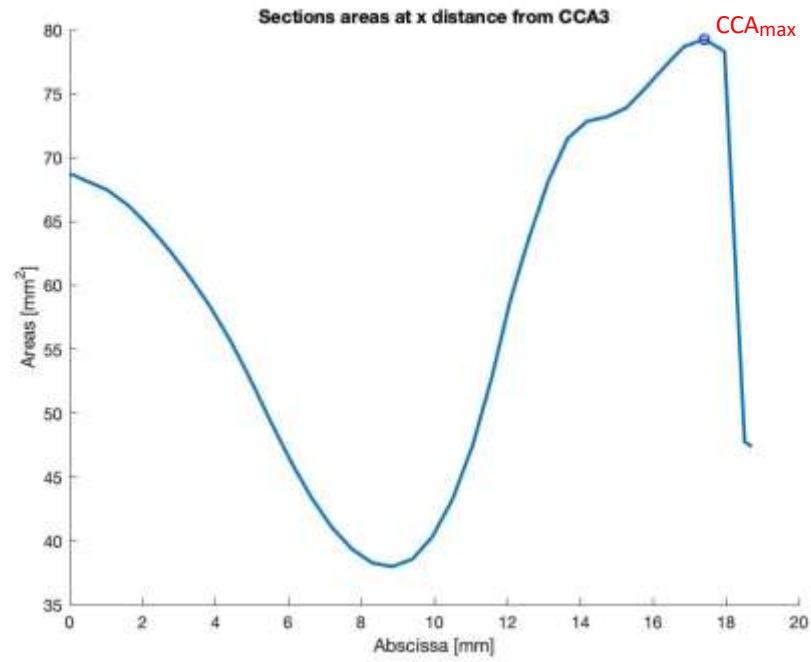
Chart 2.2: Areas along the mean centreline for different values of abscissa for Case 1 post-operative.
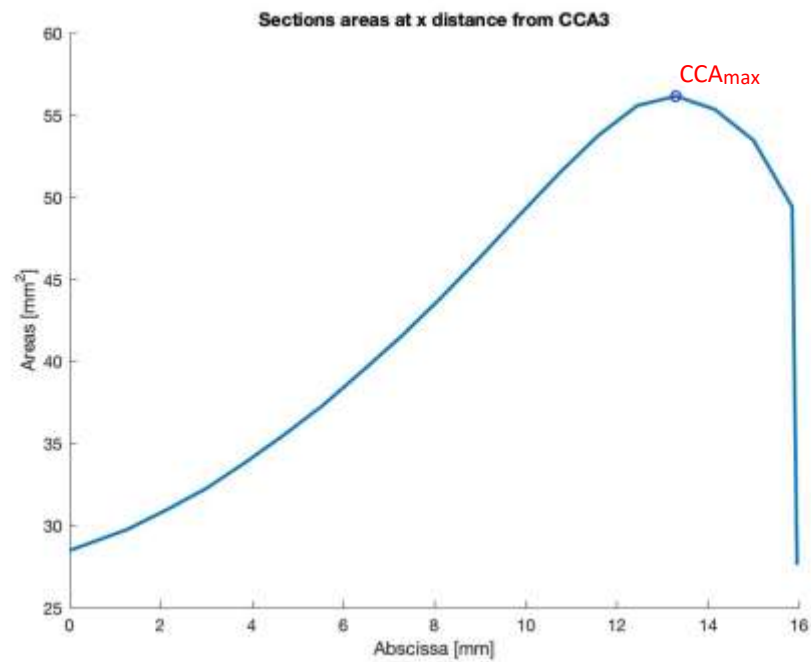
Chart 2.3: Areas along the mean centreline for different values of abscissa for Case 2 pre-operative.
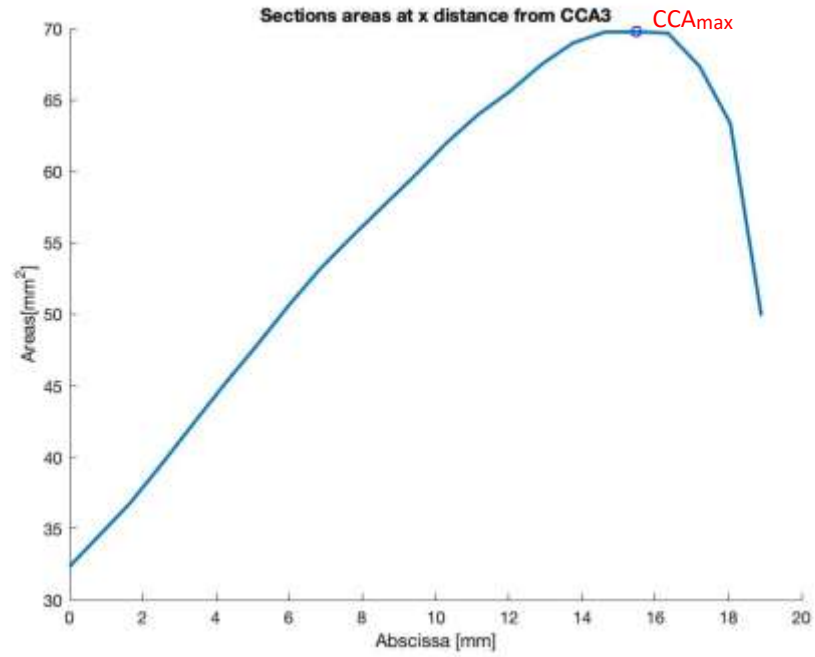
Chart 2.4: Areas along the mean centreline for different values of abscissa for Case 2 post-operative.
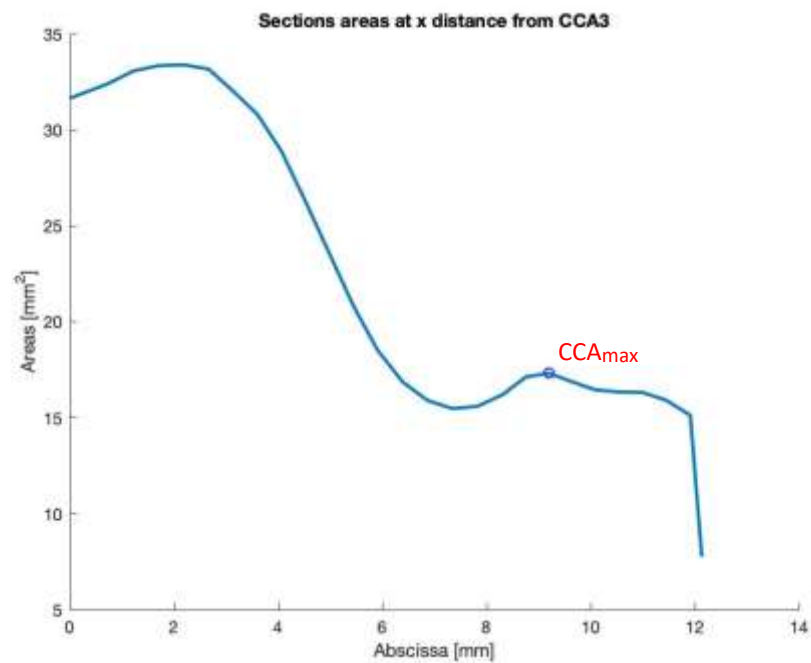
Chart 2.5: Areas along the mean centreline for different values of abscissa for Case 3 pre-operative.
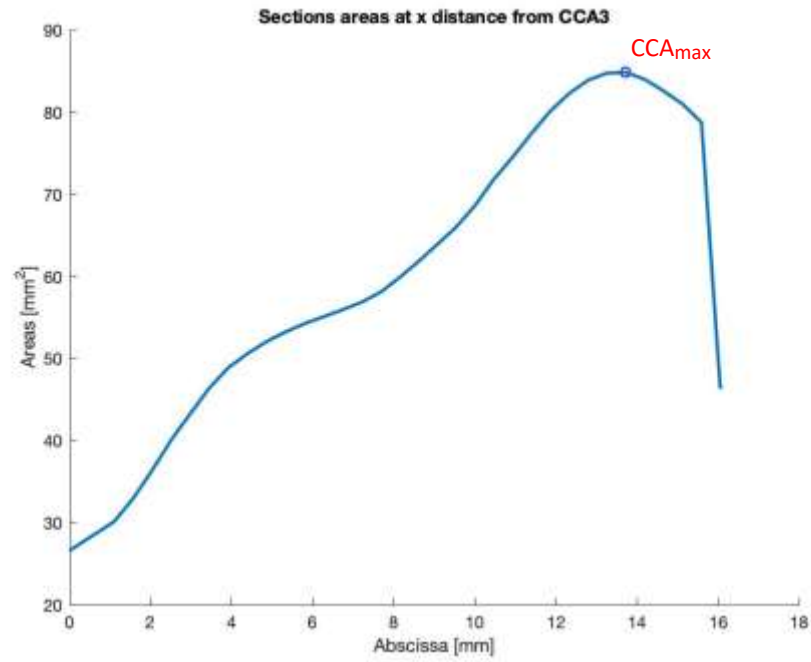
Chart 2.6: Areas along the mean centreline for different values of abscissa for Case 3 post-operative.
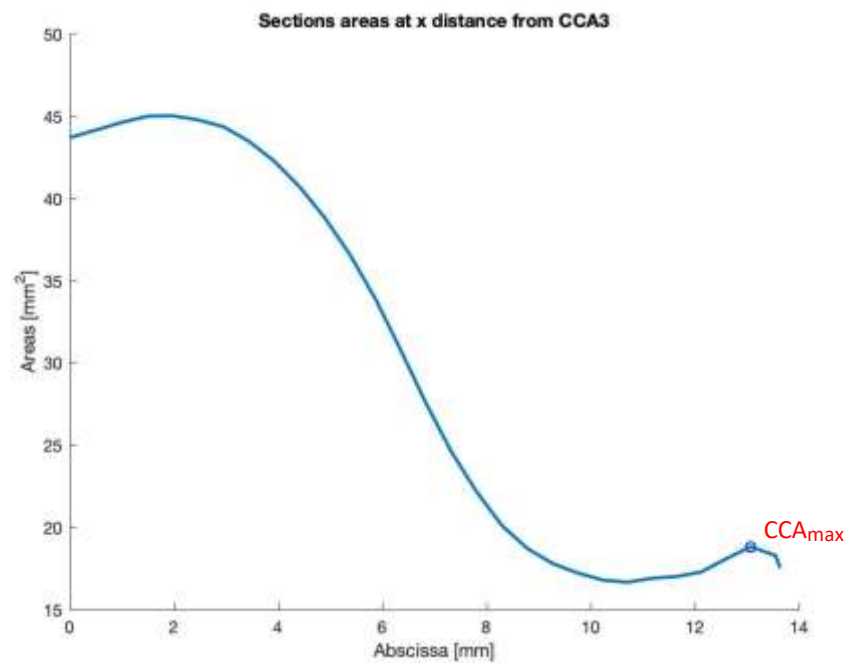
Chart 2.7: Areas along the mean centreline for different values of abscissa for Case 4 pre-operative.
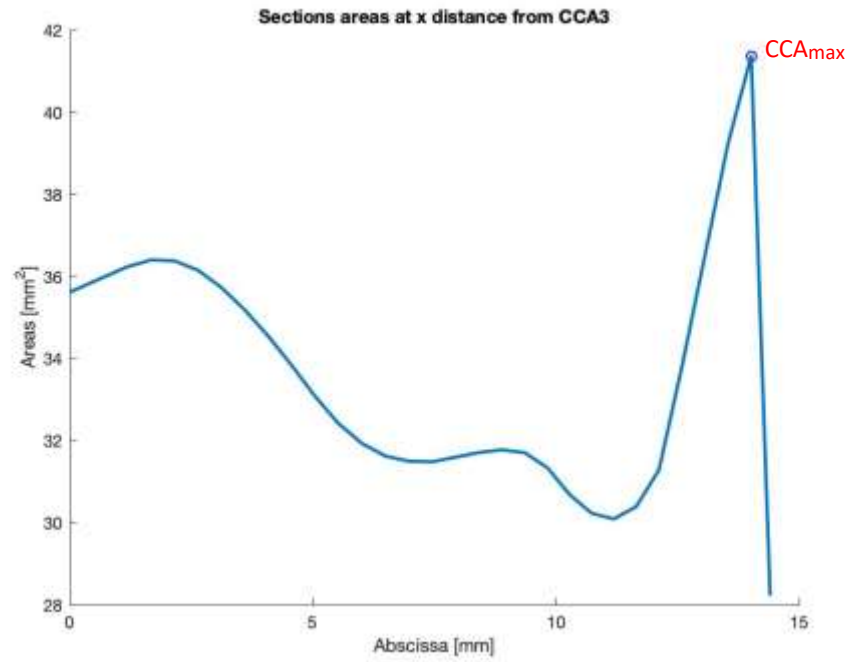
Chart 2.8: Areas along the mean centreline for different values of abscissa for Case 4 post-operative.

As the plots show it could lead to more appropriate results searching the CCA_{max} among the CCA sections included between CCA_0 and the flow divider.
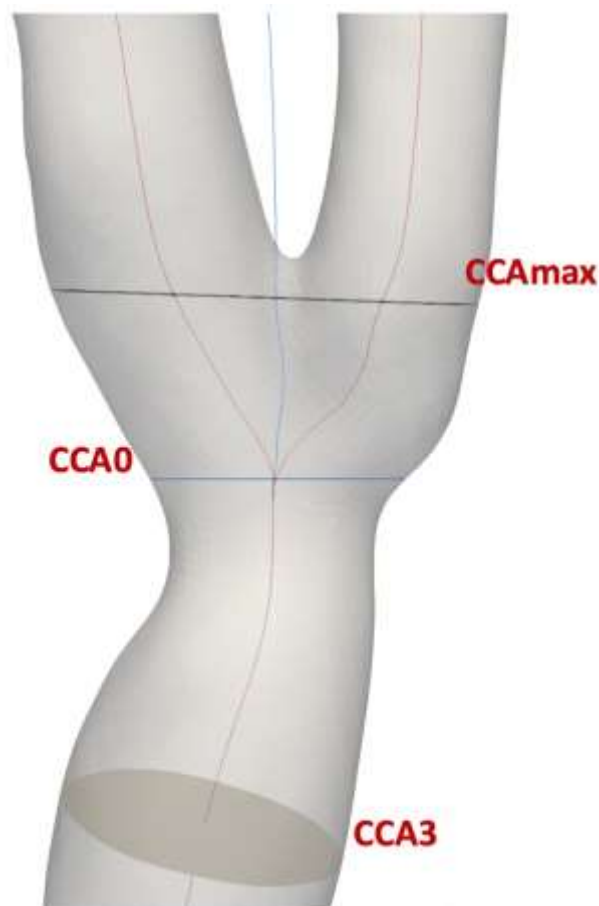
Figure 2.10: Redefinition of the searching zone for Flare.

2.5.2 Bijari's processing

Two different scripts have been written using Python, one for flare parameters and one for tortuosity parameters. The scripts include also libraries from VMTK (the Vascular Modeling Toolkit): an open source software with libraries and tools for 3D reconstruction, mesh generation, geometric analysis and surface data analysis for image-based modeling of blood vessels. [16] The scripts are completely automatic and can calculate all the parameters, it is only necessary to graphically initialize the inlet and outlet surfaces of the vessels.

The first script computes the parameters Flare A, Flare M, Flare R.

1. First step is to open .stl file. If it is necessary is possible to clip the model in order to open the section of inlet and two outlets. Furthermore, it can add flow extension with different values.

2. It computes the centerlines of the model using as input the inlet and outlets of the vessel. Two centrelines, one for ICA and one for ECA, are generated. The centerlines allow to compute automatically the CCA_3 and CCA_0 sections. The next step is to calculate the mean of the centrelines of the ICA and ECA, with a simple mathematical mean of coordinates. The mean centreline is computed in order to find the flow divider, since it is the intersection between the model and the mean centreline.

Figure 2.11: ICA centreline (blue), the ECA centreline (black), the mean centreline (red) and the CCA_3 section.

3. The mean centerline is cut between the CCA_0 and the flow divider and is used to extract the CCA sections. The CCA_{max} is found and the mean centerline is cut between CCA_3 and CCA_{max}. After that, the script uses a VMTK library to extract area and the

coordinates of the centre of the section found in order to have all the necessary data to compute flare parameters.
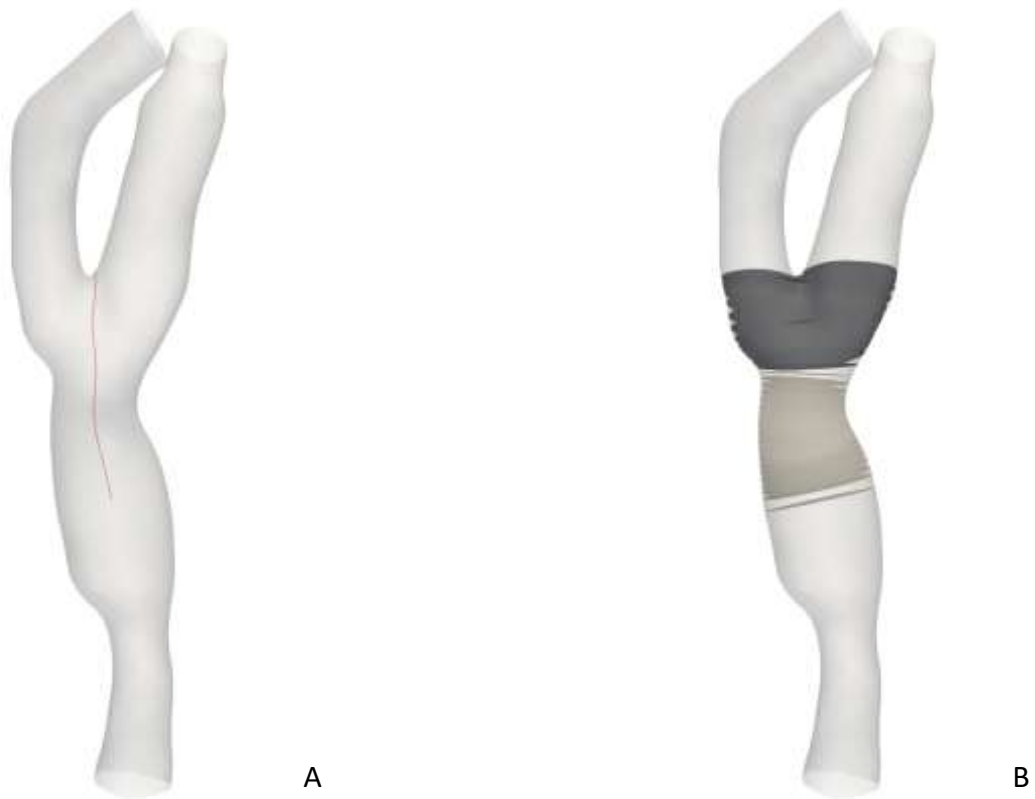
Figure 2.12: (A) Mean centerline cut between CCA$_3$ and the flow divider. (B) CCA sections along the cut mean centreline.

The second script computes the parameters Tort3D, Tort2D, Curv2D. The first is in common with the Flare script.

1. First step is to open .stl file. If it is necessary is possible to clip the model in order to open the section of inlet and two outlets. Furthermore, it can add flow extension with different values. (The example shown an extensive ratio of 1).

2. Just like the previous script, it computes the centerline of the ICA branch. The centerline is resampled and filtered in order to be more smoothed. The script then computes curvature and torsion along the ICA centerline, finds all the peaks from the torsion values and finds the closer peak to the flow divider. The closer peak to the flow divider is what is defined by Bijari *et al.* as "Inflection Point". The script then gets in input this centerline and it obtains an analytical description of the centerline created with a 5th grade spline and calculates curvature and torsion after successive derivatives. After this step, it cuts the ICA between the CCA$_3$ and the Inflection Point and compute the Tort 3D parameter.

Figure 2.13: The mean centreline (red) and the cut ICA between CCA$_3$ and the Inflection Point (black).

3. The script uses all the 3D points of the centerline from CCA$_3$ surface until the inflection point to calculate the plane that best fits the centerline by minimizing the sum of the quadratic distances, perpendicular to the plane, between the plane and the points. The fit is made by computing the eigen-values/vectors related to the distribution of points. The centerline's points are then projected onto the best fit plane and from those new points is calculated the Tort2D parameter, in which L and D are calculated from the projected points.

4. At the end, the script calculates the curvature of a circle fitted via least squares to the projected centreline segment between CCA$_3$ and the inflection point and it computes Curv2D.

Chapter 3
Hemodynamic and CFD Analysis

The aim of this chapter is to explain the role of hemodynamic in the cardiovascular system and show how our CFD simulations were run.

It is divided in its parts: the role of hemodynamic and CFD processing. The first paragraph describes the Eulerian and Lagrangian approaches and the hemodynamic wall descriptors useful for the post processing. The paragraph regarding the CFD processing is divided in: motion equations, computation of flow extensions, sensitivity analysis and unsteady simulations.

3.1 The role of hemodynamic

Hemodynamic factors are strictly linked to the local triggering and progression of atherosclerosis; they control multiple features of vascular physiology and biology, having a significant role in arterial disease development as well as vascular homeostasis. The complex interaction among biological factors, hemodynamic factors and systemic risk factors leading to the formation of atherosclerotic plaque and to vascular pathology may be a triangle in analogy to Virchow's triad of thrombosis.
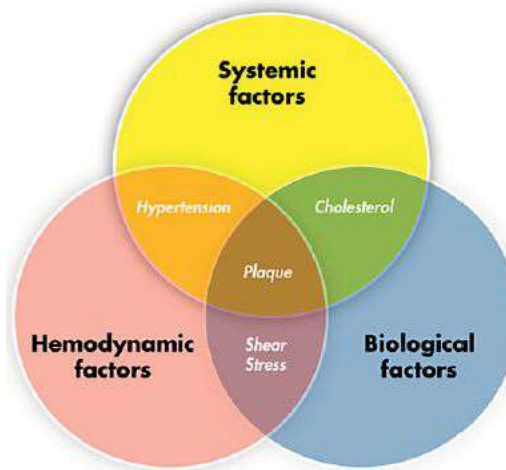
Figure 3.1: Factors related in atherosclerotic plaque formation. [15]

A hemodynamic factor that is generally documented for its contribution in atherosclerotic plaque formation is endothelial shear stress, the frictional force per area applied at the vessel wall by the flowing blood. In general, during the initial plaque development, oscillatory and low shear stress is considered atherogenic, while high shear stress is athero-protective. Wall shear stress (WSS) can be calculated by multiplying the gradient of local blood flow velocity close to the vessel wall and the blood viscosity. The balance of pressure, viscous and centrifugal forces determines that blood flow velocities are distributed; highly asymmetric and skewed velocity profiles can be found in arterial branching and bifurcating areas and in bending arterial segments or, more generally, where flow separation and reattachment can occur together with complicated different patterns of high and low shear stress at the vessel wall. Additionally, the direction and velocity of blood flow in arteries oscillate throughout the cardiac cycle. Currently, shear stress is most usually analysed using advanced computational fluid dynamics (CFD). Nevertheless, these approaches are classically limited to engineering experts and can be influenced by assumptions and uncertainties in input parameters like vessel geometry and blood flow rates. [15]

The internal side of blood vessels is lined by a monolayer of endothelial cells, which control leukocyte transmigration and adhesion, platelet aggregation, mass permeability/ transport and smooth muscle function (vessel dilation and constriction) through the deliverance vasoactive substances such as endothelin-1, prostacyclin and nitric oxide and by the expression of junctional molecules and specific adhesion.

Endothelial cells are straight in contact with flowing blood; shear stress effects vascular activity by activating endothelial mechanoreceptors which diffuse biochemical signals in reply to the shear stress. The endothelium in areas of disturbed shear stress shows augmented activation and expression of the pro-inflammatory Nuclear Factor Kappa B, an augmented oxidant/anti-oxidant balance, triggering of the unfolded protein reply and endoplasmic reticulum stress as well as low expression of endothelial nitric oxide synthase, thrombomodulin and Krüppel like transcription factors, all protective factors. These disturbed shear stress-induced variations in endothelial cell behaviour effect the vulnerability, inflammation and injury of the vessel, they limit the recovering capacity of endothelial cells and, in this way, explain the principal nature of atherosclerotic lesion creation. [15]

3.1.1 Eulerian And Lagrangian Approach

In a continuous, with a specific reference system, some quantities exist attributed to a reference position x, y, z at one time t. It is common use to call this general variable *c* and it can be tensor, vector or scalar:

$$c = c(x, y, z, t)$$

Partial derivate, called Eulerian, definition is:

$$\frac{\partial c}{\partial t}$$

and it is used to analyze the variation of the variable during time in a fix point.

The derivate of *c* over time is:

$$\frac{dc}{dt} = \frac{\partial c}{\partial t} + \frac{dx}{dt} \cdot \frac{\partial c}{\partial x} + \frac{dy}{dt} \cdot \frac{\partial c}{\partial y} + \frac{dz}{dt} \cdot \frac{\partial c}{\partial z}$$

and represents the variation of the function along all the system spatial coordinates (*x, y, z*) and the temporal coordinate (*t*). The attention is focused on a particular point in space during time.

Another approach, called Lagrangian, is to analyze the matter motion along *x, y* and *z* during time; in this case the derivative explains what happens to a moving particle and is written as:

$$\frac{Dc}{Dt} = \frac{\partial c}{\partial t} + v_x \cdot \frac{\partial c}{\partial x} + v_y \cdot \frac{\partial c}{\partial y} + v_y \cdot \frac{\partial c}{\partial z}$$

[18]

Current in vitro imaging procedures, such as particle image velocimetry, computational methods, used to quantify blood flow and in vivo imaging systems, like phase-contrast magnetic resonance imaging and PC MRI use, typically, a Eulerian approach. The velocity field itself has marginal clinical importance, classically the interest is in the transport of fluid and in the forces produced by it. Although is generally rational to deduce the bulk fluid motion from

the velocity data, an appropriate quantification of flow structures in unsteady and physiologic flows could be hard to obtain and could lead to misinterpretation. [19]

The Lagrangian specifications can be used to evaluate emodynamic quantities such as particle deposition, residence times of individual fluid particles and particle shear exposure since they are based on the dynamical path history of a fluid element, information that is identified through the Lagrangian characterization. Other hemodynamic quantities, such as WSS, can be studied with Eulerian approach that offers a picture of the spatial distribution of these quantities in a specific time. Particle stress history, particle deposition and the level of pitch and torsion of flow in space and time, obtained through helicity can be described by Lagrangian derived parameters. [19]

3.1.2 The hemodynamic descriptors

The importance of WSS is clear from the effects on vascular biology. Shear stress may alter plaque composition and its vulnerability, furthermore triggers the developing of atherosclerosis as said before. Indeed, shear stress distribution looked by itself critical for plaque phenotype in mice [20], i.e. plaques in a low shear stress area contained high numbers of a large lipid core and macrophages and low amounts of collagen and smooth muscle cells, conforming to a vulnerable plaque phenotype in human. In contrast, oscillatory shear stress provokes smaller plaques with a more stable phenotype.

The link between atherosclerosis and shear stress is correlative. Studies in vitro concerning hemodynamic in adult carotid bifurcations have shown that regions where atherosclerotic plaque formation occurs and maximum intimal thickening were related with low and oscillatory shear stress, while the marginally affected arterial lesions were exposed to high shear stress and high flow velocities. [21] [22] These remarks suggested that low shear stress may improve atherogenesis in the carotid bifurcation.

Especially, these irregularities happen in particular events:

- Oscillatory WSS: in vitro and in vivo studies of endothelial cells have revealed different effects between laminar shear flow with a certain direction and disturbed shear flow without a clear direction in terms of their structure and functions, gene expression and signal transduction. Indeed, WSS can modify the orientation and morphology of the endothelial cell layer; endothelial cells exposed to a laminar flow with high levels of WSS tend to elongate and align in the direction of flow, while in zones of disturbed flow endothelial cells endure low oscillatory WSS and they appear more polygonal without a defined orientation, without an organization of the intercellular junctional proteins and cytoskeleton;

- Residence times of macromolecules close the wall;

- Low stress value, causing an increase of the compliance of endothelial cells. Due to the low speed flow in the vessel, the cells can react with the corpusculate part of the blood. This can activate endothelial dysfunction;

- High stress value that causes laceration, conditions wall permeability and encourages albumin circulation, with the cholesterol crossing the vessel wall together with low-density lipoproteins and other macromolecules.

The role of hemodynamic, in cardiovascular flow studies, is very important for its link with cardiovascular diseases. In the past, studies efforts were primarily on wall shear stress as a quantitative indicator of disturbed flow used to: facilitate the understanding of the flow dynamics, calculate hemodynamic disorders as potential indicators of vascular wall dysfunction, decrease the inherent complexity of the four-dimensional blood flow. [3] [14]
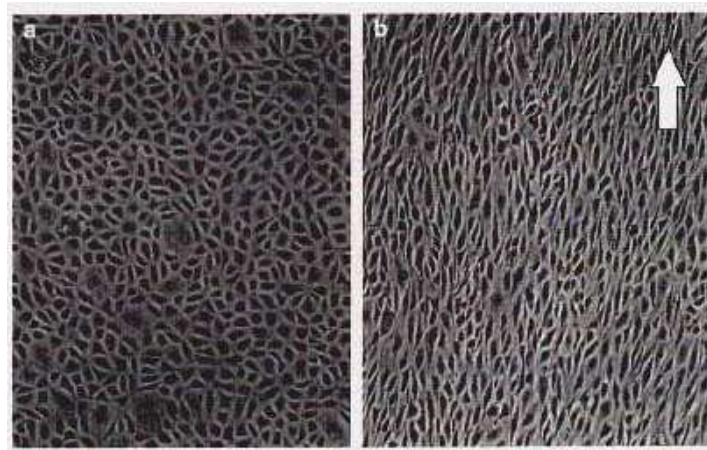
Figure 3.2: A) Endothelial cells subjected to disturbed shear flow without a defined direction; B) Endothelial cells subjected to laminar shear flow with a specific direction. [3]

Hemodynamic Wall Descriptors

- **WSS**: Wall Shear Stress is calculated as:

$$WSS = \mu \left(\frac{\partial v}{\partial y}\right)_{y=0}$$

where μ is the dynamic viscosity, v is the velocity parallel to the wall and y the distance from the wall. Techniques for determining realistic WSS employ anatomical and flow data obtained in vivo via clinical imaging (MRI, US, Angiography) and they are imported into a computational fluid dynamics (CFD) code to derive WSS and WSS-derived descriptors. [3]
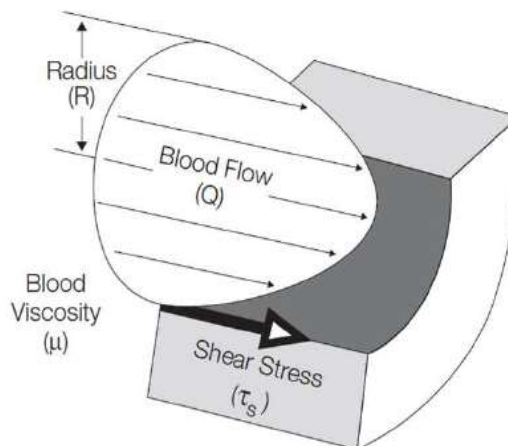
Figure 3.3: Example of WSS in a vessel. [3]

- **TAWSS:** Time-Averaged Wall Shear Stress (TAWSS) is computed by integrating each nodal WSS vector magnitude at the luminal surface over the cardiac cycle.

    It is defined as:

$$TAWSS = \frac{1}{T} \int_0^T |WSS(s,t)| \cdot dt$$

Low TAWSS values (lower than 0.4 Pa) can stimulate a proatherogenic endothelial phenotype.

Moderate TAWSS values (greater than 1.5 Pa) encourage quiescence and an athero-protective gene expression profile.

Higher TAWSS values (grater than 10-15 Pa, relevant from 25-45 Pa) can provoke endothelial trauma. [3] [23] [19]

- **OSI:** Oscillatory Shear Index (OSI) is used to classify regions on the vessel wall exposed to highly oscillating WSS directions during the cardiac cycle.

    It is defined as:

$$OSI = 0.5 \left[1 - \left(\frac{\left| \int_0^T WSS(s,t) \cdot dt \right|}{\int_0^T |WSS(s,t) \cdot dt|} \right) \right]$$

Low OSI values happen where flow disturbance is minimal.

Hight OSI values (with a maximum of 0.5) highlight locations where the instantaneous WSS diverges from the main flow direction in a large fraction of the cardiac cycle, encouraging perturbed endothelial alignment. [3] [15] [23] [19]

- **RRT:** The Relative Residence Time (RRT) links the information given from the previous indexes.

    It is defined as:

$$RRT = \frac{1}{(1 - 2 \cdot OSI) \cdot TAWSS} = \frac{T}{\left| \int_0^T WSS(s,t) \cdot dt \right|}$$

RRT is inversely proportional to the magnitude of the time-averaged WSS vector and has influences on the biological mechanisms of atherosclerosis. RRT is used to calculate the residence time of particles near the wall and suggested as a single metric of low and oscillating shear stress. It was proved that residence time of particles near the wall is proportional to a mixture of TAWSS and OSI. [3] [15]

Hemodynamic Bulk Flow Descriptors

- **H:** Helicity (H) is a measure of the alignment of the velocity and vorticity vectors and quantifies the potential for helical flow to develop. Helical flow is the result of the composition of rotational and translational motion.

    It is defined as:

$$H(x,t) = \int_D v(x,t) \cdot \omega(x,t)dV = \int_D H_k(x,t)dV$$

where $v(x,t)$ and $\omega(x,t)$ are the velocity and vorticity vector and their internal product H_k is the helicity density and x is the position in space.

Many authors emphasize the existence of a relationship between helical flow patterns and transport phenomena affecting blood-vessel wall interaction. Helical flow could be used as a fluid dynamic signature for the identification of sites prone to fibro-intimal thickening and atherosclerotic plaque formation. [19]

- **LNH:** The Localized Normalized Helicity (LNH) indicates how the velocity and vorticity fields are respectively oriented through the cosine of the angle between the local velocity and vorticity vectors.

  It is defined as:

$$LNH(x,t) = \frac{v(x,t) \cdot \omega(x,t)}{|v(x,t)||\omega(x,t)|} = cos\varphi(x,t)$$

The absolute value of LNH can assume values between one, when the flow is totally helical, and zero, in presence of reflectional symmetry (e.g. Poiseuille flow). The sign of LNH is useful to understand the direction of the rotation. Hence, LNH is a useful to describe changes in the rotation's direction of flow into vessels during the cardiac cycle. [19]
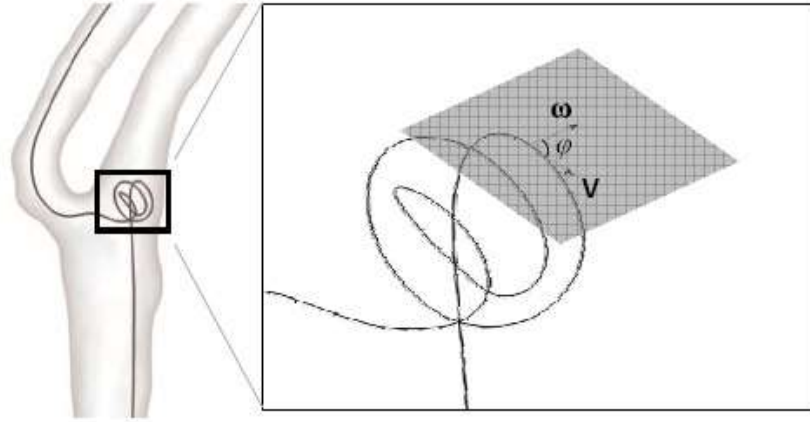
Figure 3.4: Angle between velocity and vorticity vectors. [3]

- **h1:** is the time-averaged value of the Helicity.

  It is defined as:

$$h_1 = \frac{1}{TW_i V_i} \int_{TW_i} \int_{V_i} H_k dV dt$$

were TW_i are properly defined time intervals of the cardiac cycle and V_i are volumetric fluid domains.

Therefore, h1 is equal to 0 in case of reflectional fluid domain symmetry or in case of no helicity and its sign depend on the predominant direction of rotation.

- **h2:** is the Helicity intensity.

    It is defined as:

$$h_2 = \frac{1}{TW_i V_i} \int_{TW_i} \int_{V_i} |H_k| dV dt$$

    and indicates the total amount of helical flow in the fluid domain, irrespective of direction.

- **h3:** is a non-dimensional quantity included between -1 and +1.

    It is defined as:

$$h_3 = \frac{h_1}{h_2}$$

    It is equal to -1 when only left-handed helical structures are present, 0 in reflectional symmetry and +1 in case of only right-handed helical structures.

- **h4:** describes the balance of counter-rotating structures without considering what is the major direction of rotation.

    It is defined as:

$$h_4 = |h_3|$$

    so, it ranges between 0 and +1.

3.2 CFD Processing

CFD gives an approximate solution of flow since it is a discretization method that converts integral-differential equations into algebraic, which can be computed by a computer; as the number of meshes elements increases, the approximation is smaller.

Firstly, the flow extensions have been calculated for each model. Then, a sensitivity analysis has been run for choosing the best mesh for studying the geometry. Lastly, the unsteady CFDs for each case have been run.

3.2.1 Motion Equations

For a 3D domain there are two fundamental equations that govern a fluid: the conservation of momentum and the conservation of mass.

The conservation of momentum equation is:

$$\rho \frac{\partial \vec{V}}{\partial t} + \rho (\vec{V} \cdot \nabla) \vec{V} = -\nabla p + \rho \vec{g} + \nabla \cdot \tau_{ij}$$

and the conservation of mass equation is:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \vec{V} \right) = 0$$

In the first equation the first member represents the inertial forces and the second member is the sum of pressure forces ($-\nabla p$), viscous forces ($\nabla \cdot \tau_{ij}$) and field forces ($\rho \vec{g}$).

In case of Newtonian and incompressible fluid the two equations become:

$$\begin{cases} \rho \left(\frac{\partial \vec{V}}{\partial t} + \vec{V} \cdot \nabla \vec{V} \right) = -\nabla p + \mu \nabla^2 \vec{V} + \rho \vec{g} \\ \nabla \vec{V} = 0 \end{cases}$$

The beginning of any numerical approach is the mathematical model that simplifies the exact conservation laws. As discretization method was chosen a finite volume method. The discretization of the geometric domain was obtained through a defined numerical grid composed of discrete locations in which the variables are calculated. As a result, the solution domain is composed by a finite number of sub-domains.

In CFD every finite element is considered by the software as a single control volume and equations are discretized in every single cell through the method chosen by the user. The solution is computed at the middle point of each cell and then interpolated at the grid points.

The software computes the solution of the algebraic equations by checking their convergence with an iterative method. For this project the residuals of continuity, and velocity components along the 3D axes are set to be under 10^{-4}, this satisfies the convergence criteria. Obviously, a more accurate solution is given by lower residuals.

For reaching the convergence every solution is compared with the previous and the percentage difference needs to be lower than a reverence value ε.

At the beginning of each time step values are initialised with those obtained before until the convergence is reached:

$$\frac{x_{k+1} - x_k}{x_k} < \epsilon$$

3.2.2 Flow extensions

For this thesis, the Boundary Conditions from *"Characterization of volumetric flow rate waveforms at the carotid bifurcations of older adults."* Steinman *et al. 2010* showed below in the image have been adopted.
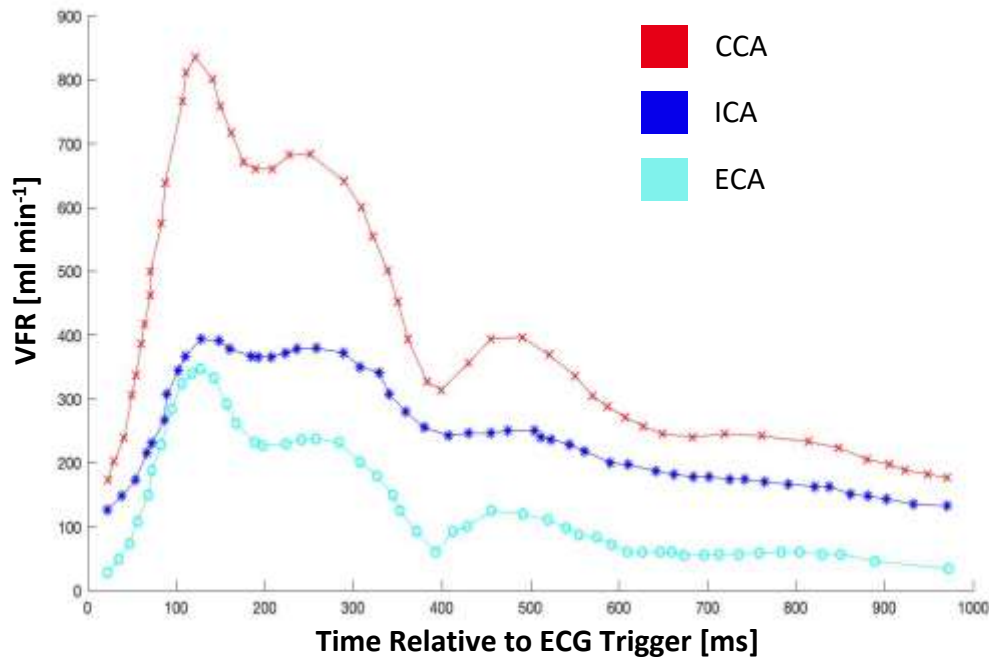
Chart 3.1: Mean carotid bifurcation VFR waveforms of older adults (N=94). CCA, ICA and ECA are shown at their corresponding mean times relative to the ECG trigger and respective mean VFR values. [24]

A Matlab script has been implemented to calculate the flow extensions for each branch (CCA, ICA, ECA) of each case. From the VFR (Volume Flow Rate), dividing by the values of inlet and outlet surfaces, has been calculated the velocity for each branch. After that, according with Reynold's definition the flow extension has been evaluated for each branch of each case.

Below the result obtained for pre and post-operative model, sorted for each branch (CCA, ICA, ECA) and case.

PRE-OPERATIVE				
CCA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.003451738	0.003028107	0.002755599	0.003608975
Area [m^2]	3.74E-05	2.88E-05	2.39E-05	4.09E-05
Mean Velocity [m/s]	1.91E-01	0.247730965	0.299151177	0.174403419
Reynolds N°	398.6142765	454.3801966	499.314992	381.2472726
F.E Length [m]	0.017886857	0.017886857	0.017886857	0.017886857
Extensive Ratio	5.181985594	5.906942555	6.491094896	4.956214544

Table 3.1: Values for CCA branch of pre-operative models.

ICA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.002466847	0.001535273	0.001252885	0.002056878
Area [m^2]	1.91E-05	7.40E-06	4.93E-06	1.33E-05
Mean Velocity [m/s]	0.217399657	0.561270723	0.842793251	0.312698836
Reynolds N°	324.839593	521.9461823	639.5876298	389.5853913
F.E Length [m]	0.010417286	0.010417286	0.010417286	0.010417286
Extensive Ratio	4.22291471	6.785300369	8.314639187	5.064610087

Table 3.2: Values for ICA branch of pre-operative models.

ECA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.002737187	0.001807371	0.002028392	0.001669687
Area [m²]	1.91E-05	1.03E-05	2.59E-05	8.76E-06
Mean Velocity [m/s]	0.09804689	0.224878741	1.29E-05	0.263495286
Reynolds N°	162.5571753	246.1860464	219.3606806	266.4867733
F.E Length [m]	0.005784343	0.005784343	0.005784343	0.005784343
Extensive Ratio	2.113243279	3.200418604	2.851688848	3.464328053

Table 3.3: Values for ECA branch of pre-operative models.

POST-OPERATIVE				
CCA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.003306835	0.002777636	0.00378615	0.003359777
Area [m²]	3.44E-05	2.42E-05	4.50E-05	3.55E-05
Mean Velocity [m/s]	0.20772927	0.294423331	0.158462766	0.201234284
Reynolds N°	416.0812017	495.3536398	363.4066446	204.7624127
F.E Length [m]	0.017886857	0.017886857	0.017886857	0.017886857
Extensive Ratio	5.409055622	6.439597317	4.724286379	5.323822729

Table 3.4: Values for CCA branch of post-operative models.

ICA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.002393937	0.001627607	0.00282147	0.002127126
Area [m²]	1.80E-05	8.32E-06	2.50E-05	8.71E-06
Mean Velocity [m/s]	0.230843696	0.499395039	0.166185281	0.292386377
Reynolds N°	334.7330083	492.3360564	241.5828854	376.7194974
F.E Length [m]	0.010417286	0.002892171	0.010417286	0.010417286
Extensive Ratio	4.351529108	6.400368733	3.692148509	4.897353466

Table 3.5: Values for ICA branch of post-operative models.

ECA	Case 1	Case 2	Case 3	Case 4
Radius [m]	0.002665317	0.001509671	0.001841809	0.001665021
Area [m²]	2.23E-05	7.16E-06	1.07E-05	1.42E-05
Mean Velocity [m/s]	0.103405813	0.322313421	0.216547843	0.26497422
Reynolds N°	166.9405001	294.7327717	284.0114237	267.233589
F.E Length [m]	0.010417286	0.010417286	0.005784343	0.005784343
Extensive Ratio	2.170226502	3.831526031	3.14057751	3.474036656

Table 3.6: Values for ICA branch of post-operative models

3.2.3 Sensitivity Analysis

For carrying out the sensitivity analysis Case 1 postoperative was chosen.
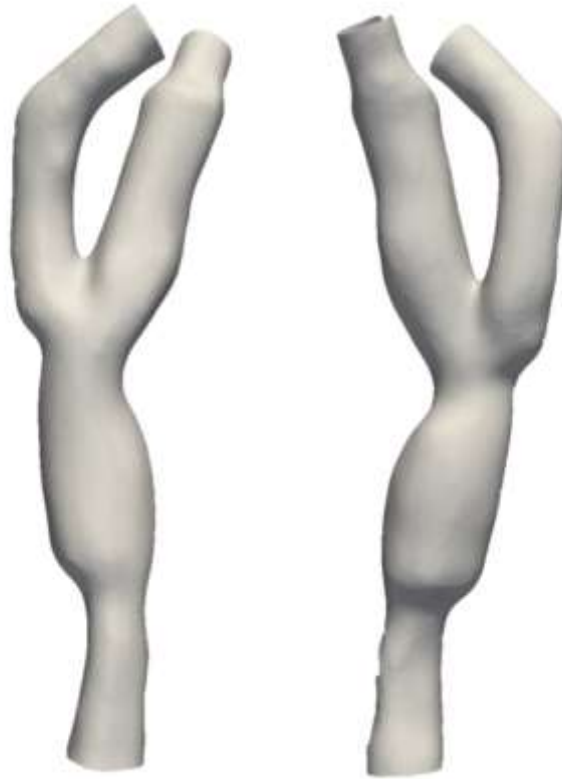
Figure 3.5: Case 1 chosen for the sensitivity analysis. The front and back side views respectively.

Ten different meshes have been created on ICEM with different dimensions of elements. All meshes are radial with tetrahedral elements.

Meshes	Total Elements	Surface Elements	Quality min Value	Quality max Value
1	1259665	130110	0.369058	0.999932
2	1992091	203066	0.371158	0.999983
3	3199442	317452	0.375339	0.999963
4	3653612	362178	0.200178	0.999973
5	4926174	480483	0.360723	0.99997
5bis	5328863	521456	0.200668	0.99999
6	5887844	564103	0.385678	0.999988
7	7107383	673357	0.200076	0.999991
7bis	7284441	698909	0.201192	0.999977
8	7806031	736634	0.338083	0.999982
9	8709439	814985	0.20033	0.999996
10	10682508	995075	0.200083	0.999998

Table 3.7: Meshes specifics.

Less Thin Meshes

1

2

3

4

5

6

7

8

9

10

Thinner Meshes
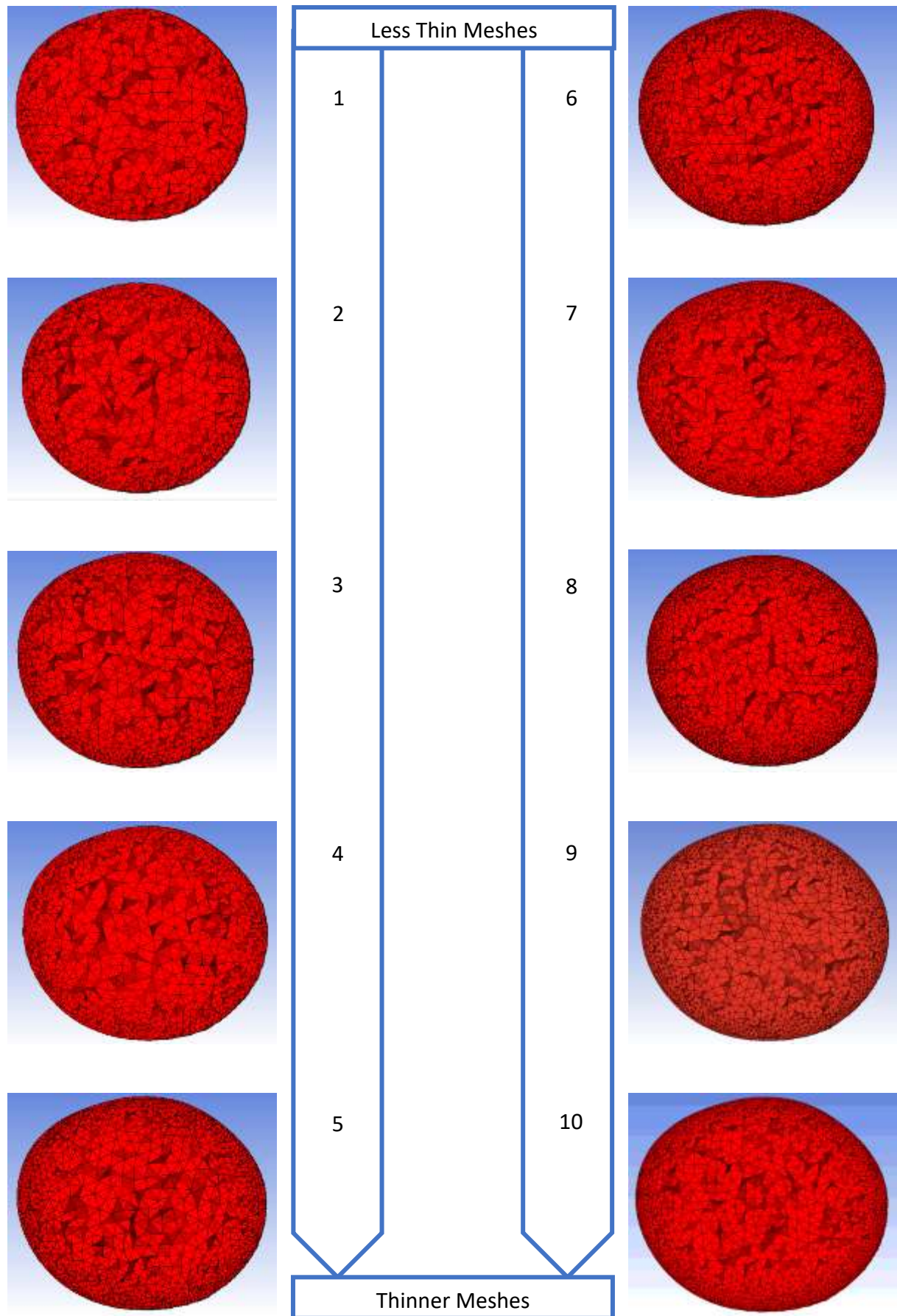
Figure 3.6: Meshes cut-planes.

A steady simulation has been run for each mesh in ANSYS Fluent with the computational settings showed in the table below and using the boundary conditions showed before. [24] Furthermore, an outflow values of 65% and 35% were imposed in ICA and ECA respectively.

QUANTITY	VALUE
ρ [kg/m^3]	1050
μ [Pa·s]	0.0035
A_{CCA} [m^2]	3.44e-5
A_{ICA} [m^2]	1.80e-5
A_{ECA} [m^2]	2.23e-5
Q_{CCA} [m^3/s]	6.88e-6
Q_{ICA} [m^3/s]	4.14e-6
Q_{ECA} [m^3/s]	2.23e-6
Mean v_{CCA} [m/s]	0.2
Mean v_{ICA} [m/s]	0.23
Mean v_{ECA} [m/s]	0.1

Table 3.8: Computational settings for sensitivity analysis.

The results of each simulation were referred to those obtained with Mesh 10 and the percentage differences have been calculated for WSS max, WSS mean and mean velocity.

The results are shown below in the graphs.
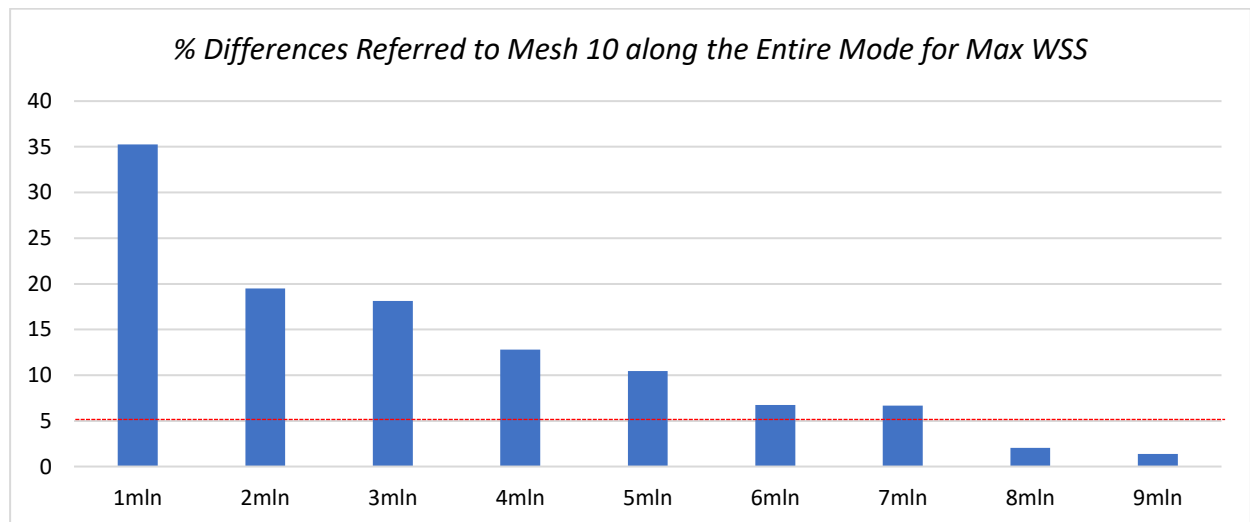
Chart 3.2: Percentage differences referred to Mesh 10 for Max WSS.
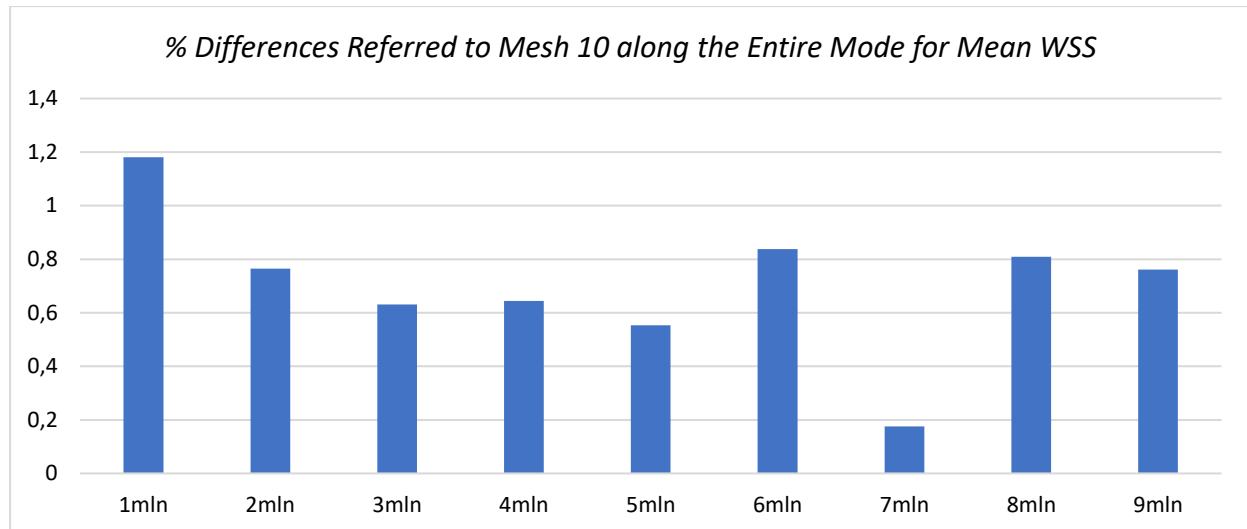
Chart 3.3: Percentage differences referred to Mesh 10 for Mean WSS.
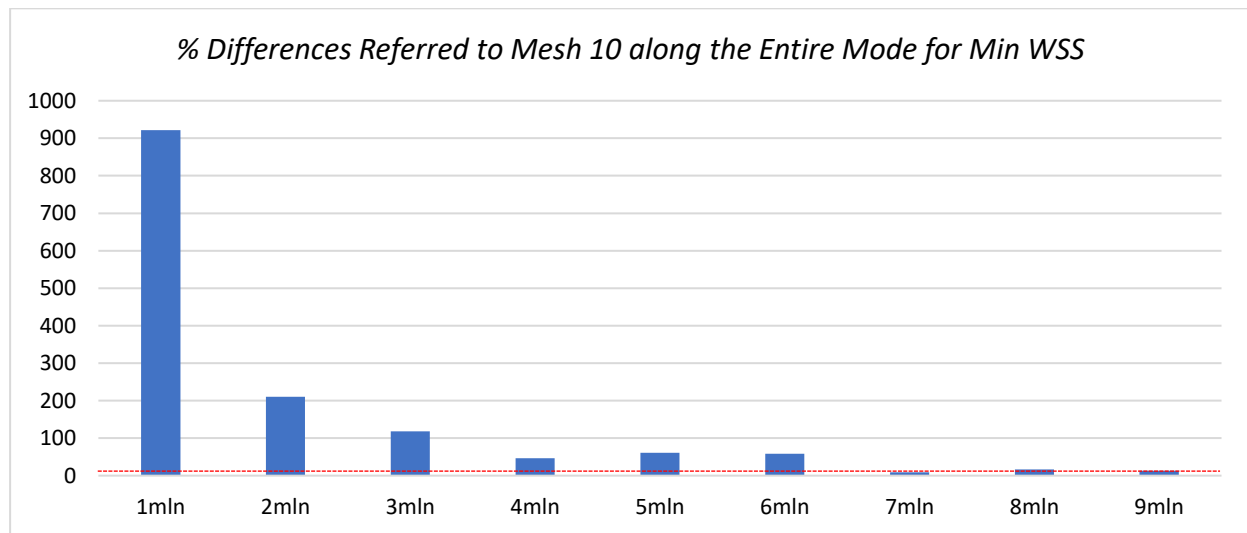
Chart 3.4: Percentage differences referred to Mesh 10 for Min WSS.
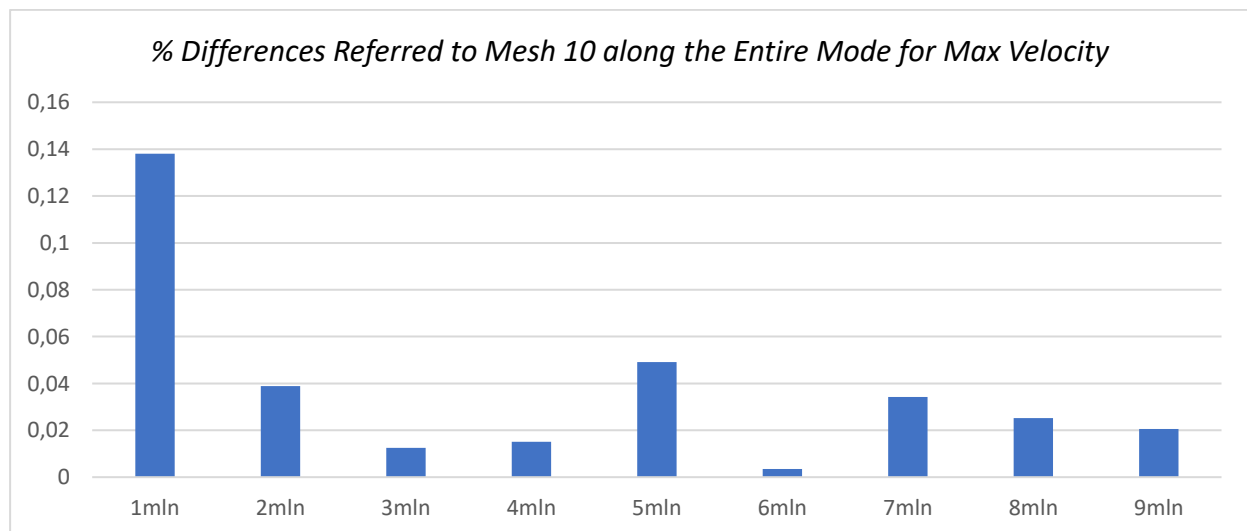
Chart 3.5: Percentage differences referred to Mesh 10 for Max Velocity.
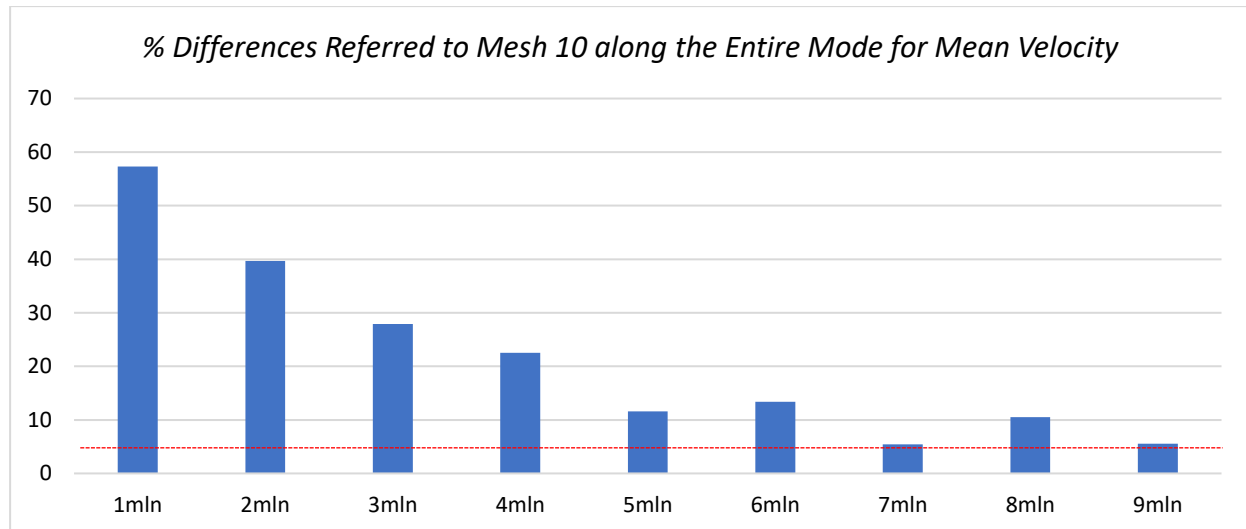
Chart 3.6: Percentage differences referred to Mesh 10 for Mean Velocity.

The results show that Mesh 7bis differs from Mesh 10 by approximately 5% for Max WSS, Min WSS and Mean Velocity, while for Mean WSS the differences are below 1%.

Mesh 7bis was obtained with the following parameters:

BULK		SURFACE	
Max Element Size	Min Element Size	Max Element Size	Min Element Size
0.39	0.29	0.108	0.055

Table 3.9: Mesh 7bis specifics.

3.2.4 CFD unsteady simulations

In order to simulate physiological behaviour of the models, unsteady simulations have been performed imposing come boundary conditions.

A mesh with the element's sizes shown above was created for each Case, both for pre-operative and post-operative.

Physical properties of blood have been imposed as for the sensitivity analysis. At the inflow section and at the ICA section it was imposed a time velocity profile, according to the flow rate described by Steinman et al. [24], scaling the velocity in accordance to the boundary area. The UDF (User-Defined Funtion) was obtained by interpolating samples given by [24] with a Fourier series:

$$f(t) = \sum_{n=1}^{N} [a_n \cos(2\pi nt) + b_n \sin(2\pi nt)]$$

where:

- a_n and b_n are the Fourier coefficients;

- N is the number of coefficients used to reconstruct the wave-form.

In the ECA outlet a 0 value pression was imposed.

A cardiac cycle of 1 s was set and subdivided into time steps of 2ms.

The wall was considered rigid and a no-slip condition was imposed, this guarantees a null relative-to-the-boundary velocity.

Lastly, before starting the simulations, all the variables and the motion field were set to zero.

Chapter 4
Results and discussion:
Patch Cases

The aim of this chapter is to show the different results obtained with morphometric analysis and CFD simulations for the two patch cases and it is divided in two parts. The first is about the results obtained from Morphometry Analysis and the second is about the results obtained from CFD simulations.

The Morphometric Analysis was made for both pre- and post-operative geometries while the CFD simulations were run only on post-operative.

4.1 Morphometric Analysis

4.1.1 Results

The scripts described in Chapter 2 has been used for evaluating the morphometric characteristics for Case 3 and Case 4.

Below, for each case, a picture shows the respective CCA_3, CCA_{max} and mean centerline cut between the two sections.
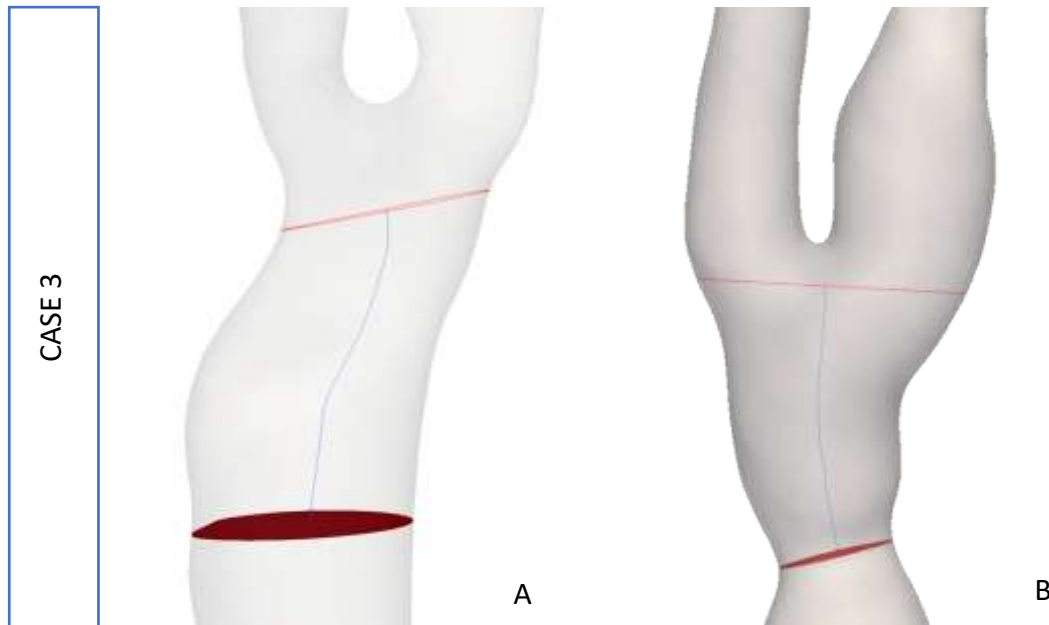
Figure 4.1: Case 3. A) Pre-operative geometry B) Post-operative geometry. Both with CCA_3, CCA_{max} and cut-mean centerline.
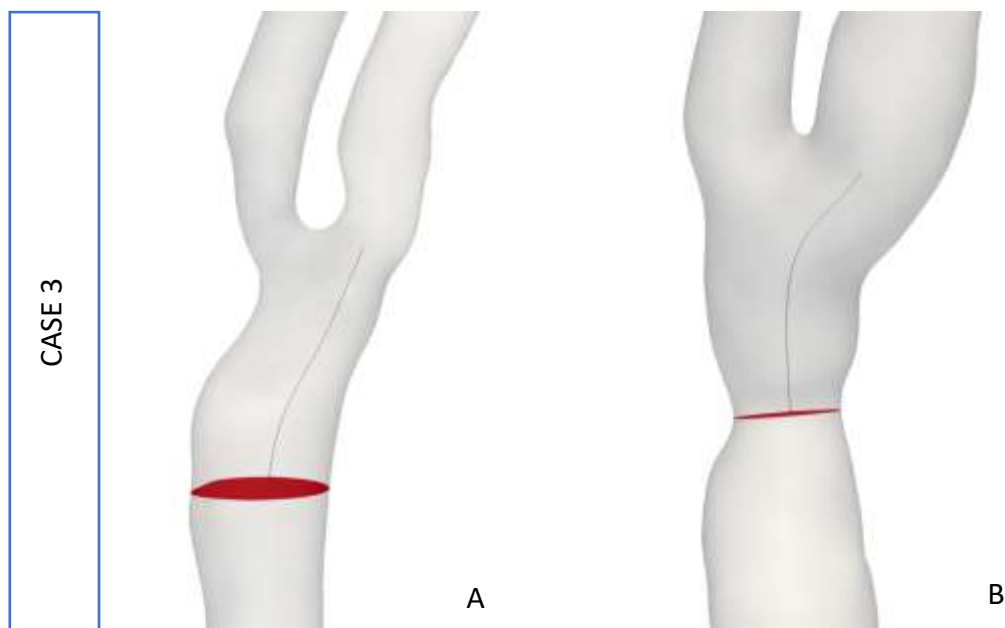
Figure 4.2: Case 3. A) Pre-operative geometry B) Post-operative geometry. Both with CCA_3 and ICA centerline cut between CCA_3 and the inflection-point.

CASE 3			
	Pre-Operative	Post-Operative	% Differences
Flare A	0.5469	3.1915	483.5962
Flare M	0.8420	2.1641	157.0191
Flare R	-0.0594	0.2578	534.0109
Tort 3D	0.0330	0.0578	75.3384
Tort 2D	0.0325	0.0557	71.3466
Curv 2D [mm^{-1}]	0.0874	0.0840	-3.8697

Table 4.1: Case 3. Flare and Tortuosity parameters for pre- and post-operative geometries with percentage differences.

Concerning Flare, it is possible to notice how all parameters increase after surgery. The maximum variation is observed for flare R and is around 534%.

Also for Tortuosity parameters is evident a general increase of values except for Curv 2D. In particular the maximum variation occurs for Tort 3D and is approximately 75%.
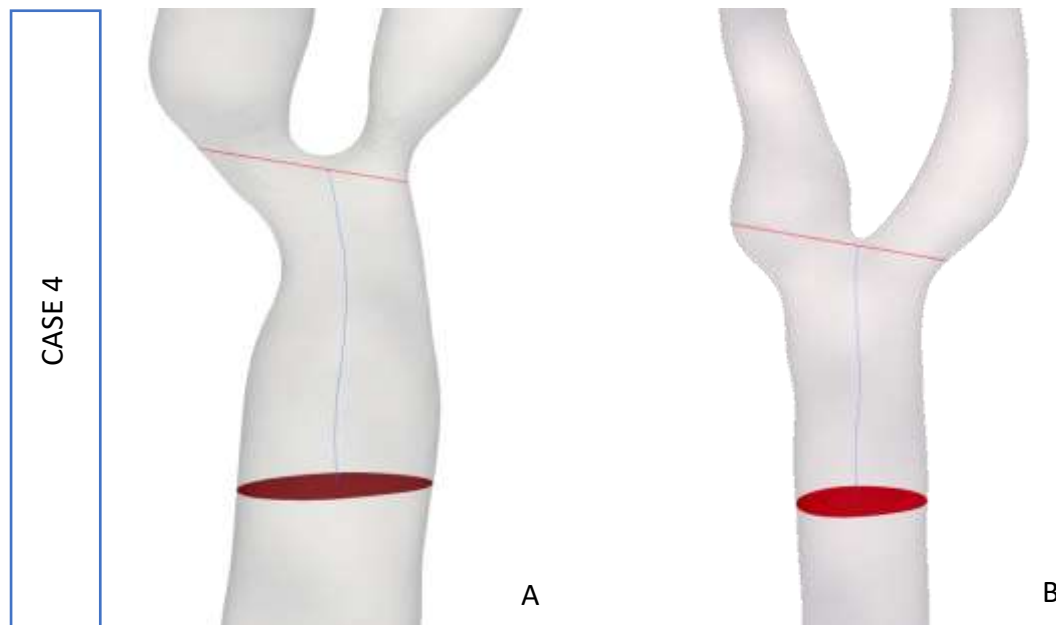
Figure 4.3: Case 4. A) Pre-operative geometry B) Post-operative geometry. Both with CCA$_3$, CCA$_{max}$ and cut-mean centerline.

CASE 4

A

B
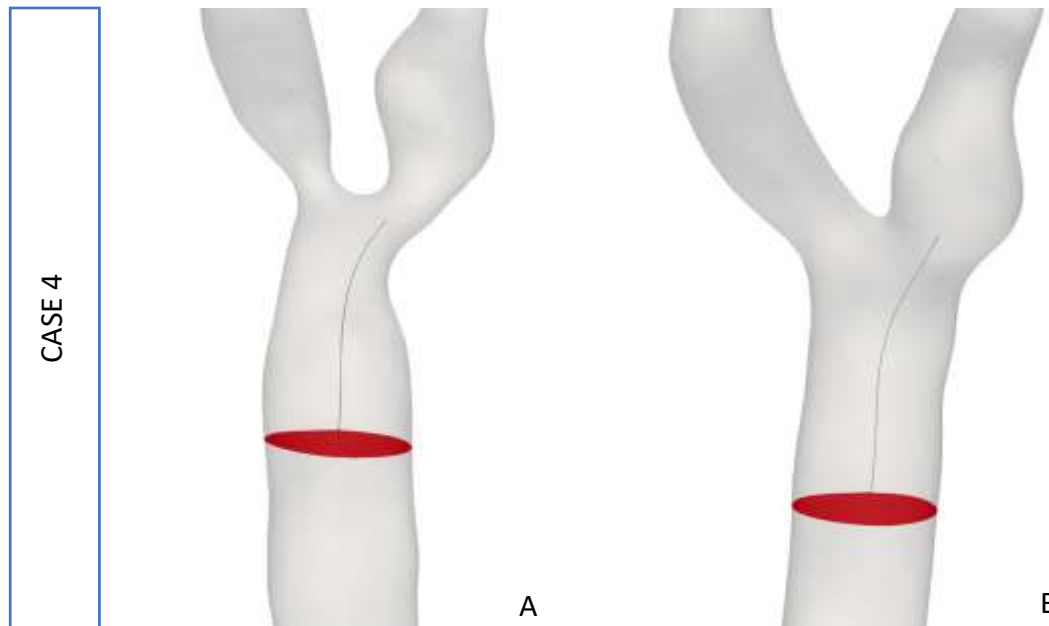
Figure 4.4: Case 4. A) Pre-operative geometry B) Post-operative geometry. Both with CCA3 and ICA centerline cut between CCA3 and the inflection-point.

CASE 4			
	Pre-Operative	Post-Operative	% Differences
Flare A	0.4326	1.1653	169.3639
Flare M	0.9397	1.6095	71.2792
Flare R	-0.0184	0.1531	933.7390
Tort 3D	0.0287	0.0312	8.7957
Tort 2D	0.0282	0.0311	10.2975
Curv 2D [mm^{-1}]	0.0661	0.0705	6.6299

Table 4.2: Case 4. Flare and Tortuosity parameters for pre- and post-operative geometries with percentage differences.

Concerning Flare, it is possible to notice how all parameters increase after surgery. The maximum variation is observed for flare R and is around 934%.

Also for Tortuosity parameters is evident a general increase of all values. In particular the maximum variation occurs for Tort 2D and is approximately 10%.

4.1.2 Discussion

Regarding Flare A, the increase of the values is the result of a restoring of a more physiological vessel lumen. Since the atherosclerotic plaque was located in correspondence of the CCA$_{max}$, after the endarterectomy Flare M increases because only the CCA$_{max}$ diameter increases. Regarding Flare R, the increase of the values shows a higher expansive rate of the vessels. Low values of Flare tend to describe a less atherogenic geometry, preventing from flow separation from the vessel's walls.

Regarding Tort 3D, Tort 2D and Curv 2D, the increase of the values can be associated to a more physiological geometry. Tort 3D focuses on the flow divider region and is calculated

with the ICA centerline between CCA_3 and the inflection point, Tort 2D is linked only to secondary flows driven by planar curvature and Curv 2D is more directly linked to the production of secondary flows.

4.2 CFD

4.2.1 Results

Unsteady CFD simulations were used for evaluating the percentage of areas exposed to risk values of TAWS, OSI, RRT. To do so, the 10th and 20th percentile of TAWSS values and the 80th and 90th percentile of OSI and RRT values were calculated. The results are shown below.

PERCENTILE		
	10th	20th
TAWSS [Pa]	0.42211593	0.5678807
	80th	90th
OSI	0.07976232	0.17989759
RRT [Pa^{-1}]	2.17536964	3.86766419

Table 4.3: Percentile.

Concerning bulk flow, helicity-based descriptors as h1, h2, h3 and h4 have been calculated from Helicity.

TAWSS

The areas exposed to values below 10th percentile present major risk of atherogenic phenomena and after surgery restenosis.

■ <10th percentile

■ <20th percentile

Figure 4.5: Case 3 - areas exposed to TAWSS values lower than 10th percentile (black) and 20th percentile (red).

PERCENTAGE OF RISK AREAS - TAWSS		
	10th	20th
CCA	13.137	17.458
ECA	0.4564	2.032
ICA	3.971	7.446
TOT	17.565	26.936

Table 4.4: Case 3 – percentage of areas exposed to TAWSS values lower than 10th percentile and 20th percentile.

The wider area exposed to risk is located in CCA branch and is around 17% of all-model area. The total amount of area exposed to TAWSS values lower than the 10th percentile is around 18% and increases to 27% considering the 20th percentile.

■ <10th percentile
■ <20th percentile
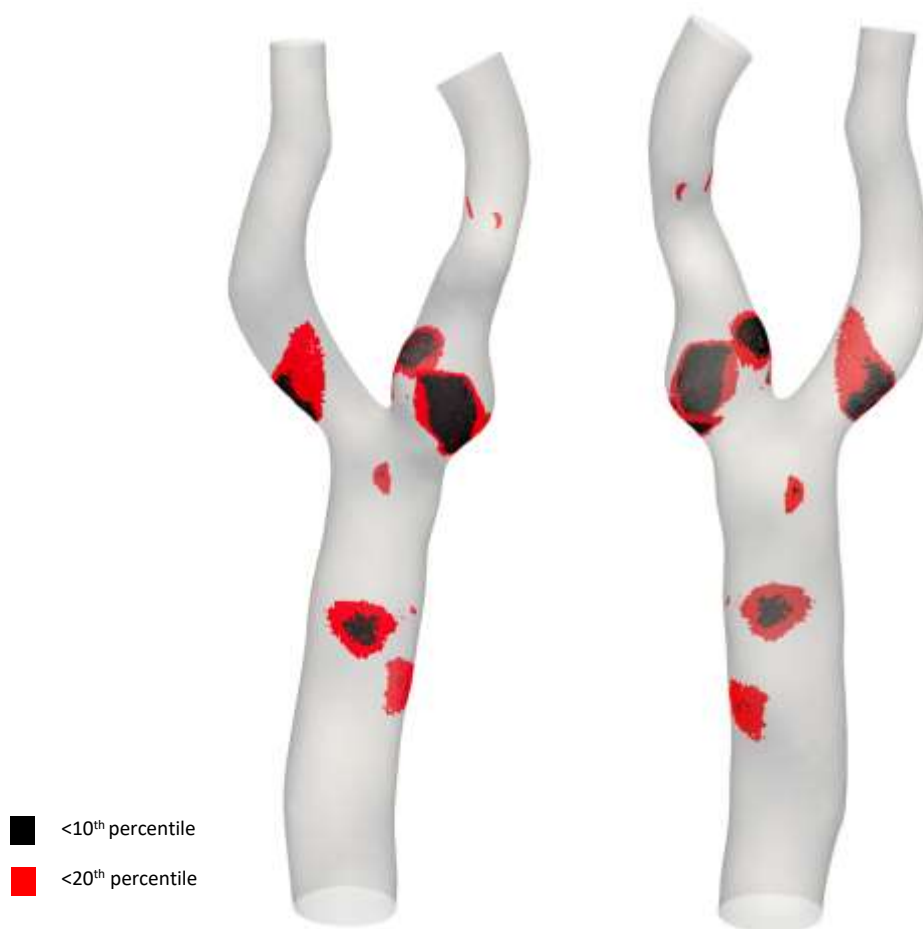
Figure 4.6: Case 4 - areas exposed to TAWSS values lower than 10th percentile (black) and 20th percentile (red).

PERCENTAGE OF RISK AREAS - TAWSS		
	10th	20th
CCA	0.314	1.850
ECA	0.875	1.954
ICA	1.928	3.254
TOT	3.117	7.057

Table 4.5: Case 4 – percentage of areas exposed to TAWSS values lower than 10th percentile and 20th percentile.

The wider area exposed to risk is located in ECA branch and is around 3% of all-model area. The total amount of area exposed to TAWSS values lower than the 10th percentile is around 3% and increases to 7% considering the 20th percentile.

OSI

The areas exposed to values over 90th percentile present major risk of atherogenic phenomena.

>90th percentile (black square)
>80th percentile (red square)

Figure 4.7: Case 3 - areas exposed to OSI values higher than 90th percentile (black) and 80th percentile (red).

PERCENTAGE OF RISK AREAS - OSI		
	80th	90th
CCA	11.692	5.167
ECA	4.803	1.478
ICA	7.481	1.21
TOT	23.975	7.855

Table 4.6: Case 3 – percentage of areas exposed to OSI values higher than 80th percentile and 90th percentile.

The wider area exposed to risk is located in CCA branch and is around 12% of all-model area. The total amount of area exposed to OSI values higher than the 90th percentile is around 8% and increases to 24% considering the 80th percentile.
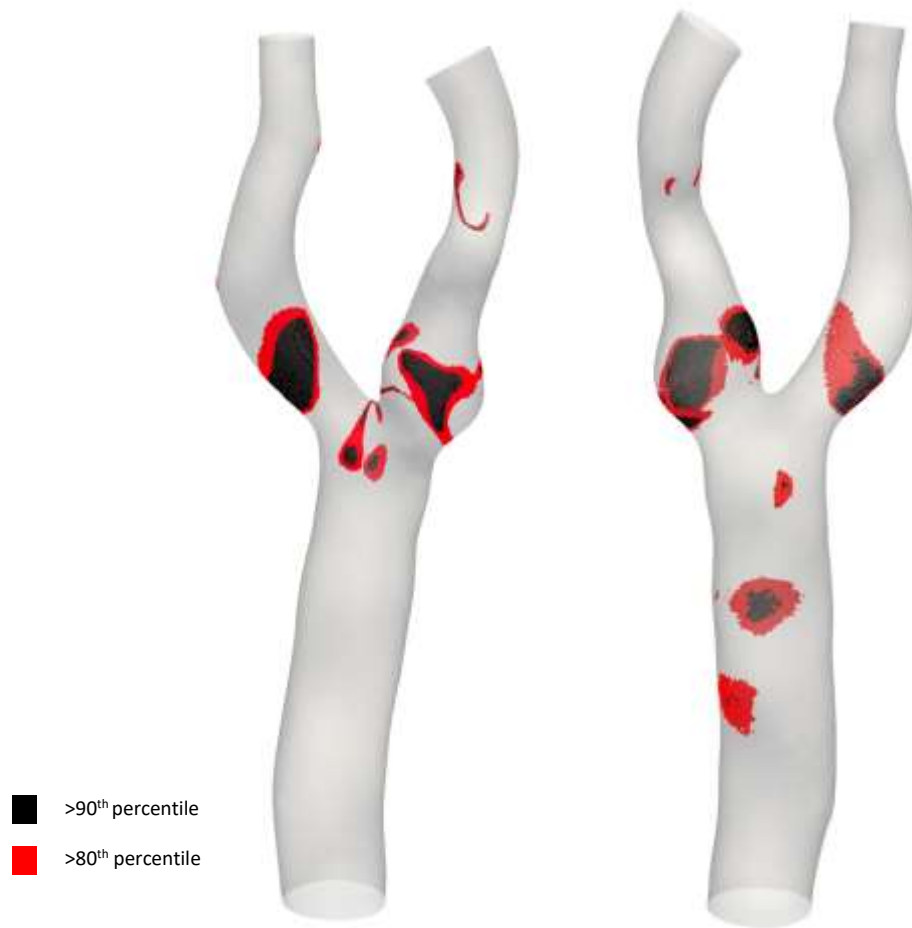
Figure 4.8: Case 4 - areas exposed to OSI values higher than 90th percentile (black) and 80th percentile (red).

PERCENTAGE OF RISK AREAS - OSI		
	80th	90th
CCA	0.257	0.056
ECA	2.789	1.759
ICA	2.192	0.9
TOT	5.238	2.714

Table 4.7: Case 4 – percentage of areas exposed to OSI values higher than 80th percentile and 90th percentile.

The wider area exposed to risk is located in ICA branch and is around 3% of all-model area. The total amount of area exposed to OSI values higher than the 90th percentile is around 3% and increases to 5% considering the 80th percentile.

RRT

The areas exposed to values over 90th percentile present major risk of atherogenic phenomena.

Figure 4.9: Case 3 - areas exposed to RRT values higher than 90th percentile (black) and 80th percentile (red).

PERCENTAGE OF RISK AREAS - RRT		
	80th	90th
CCA	16.138	9.396
ECA	2.482	0.877
ICA	6.912	2.105
TOT	25.532	12.379

Table 4.8: Case 3 – percentage of areas exposed to RRT values higher than 80th percentile and 90th percentile.

The wider area exposed to risk is located in CCA branch and is around 16% of all-model area. The total amount of area exposed to RRT values higher than the 90th percentile is around 12% and increases to 26% considering the 80th percentile.
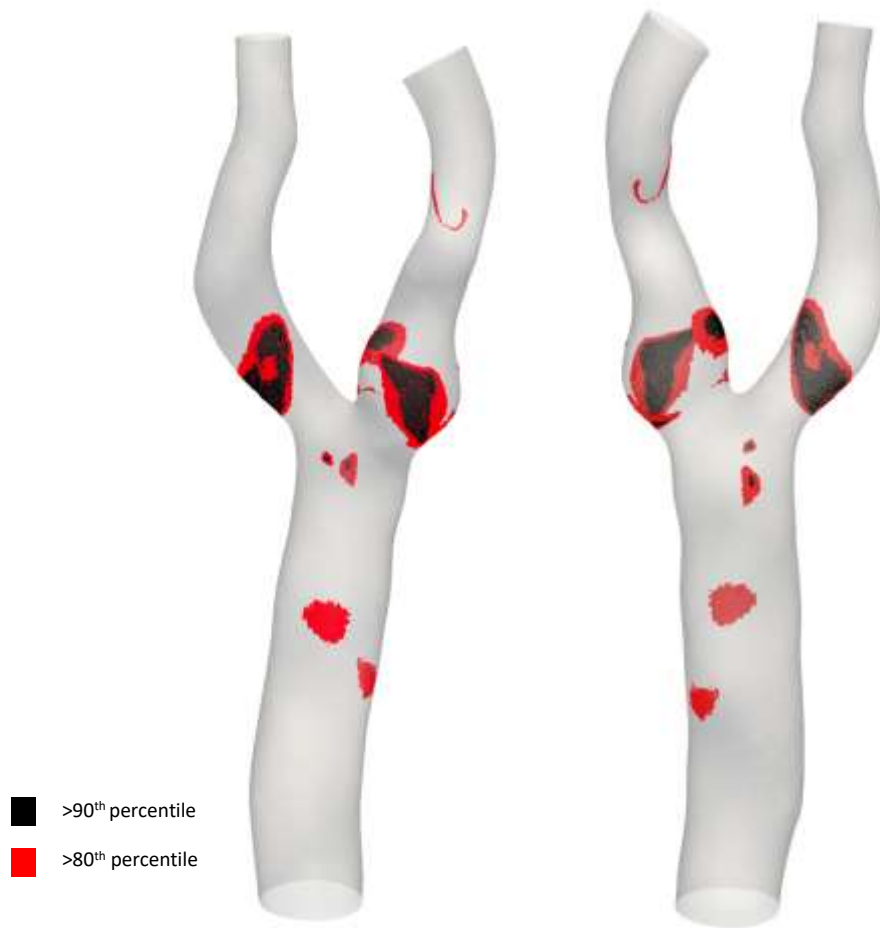
Figure 4.10: Case 4 - areas exposed to RRT values higher than 90th percentile (black) and 80th percentile (red).

PERCENTAGE OF RISK AREAS - RRT		
	80th	90th
CCA	0.934	0.022
ECA	2.147	1.417
ICA	2.84	1.181
TOT	5.921	2.62

Table 4.9: Case 4 – percentage of areas exposed to RRT values higher than 80th percentile and 90th percentile.

The wider area exposed to risk is located in ECA branch and is around 3% of all-model area. The total amount of area exposed to RRT values higher than the 90th percentile is around 3% and increases to 6% considering the 80th percentile.

Helicity-Based Descriptors

HELICITY- BASED DESCRIPTORS				
	h1 [m/s^2]	h2 [m/s^2]	h3	h4
Case 3	0.999	6.22	0.161	0.161

Table 4.10: Case 3 – Helicity-Based descriptors.

The table shows that the Helicity Intensity (h2) consistently higher than the Mean Helicity (h1). This is also shown by h3 value 0.16.
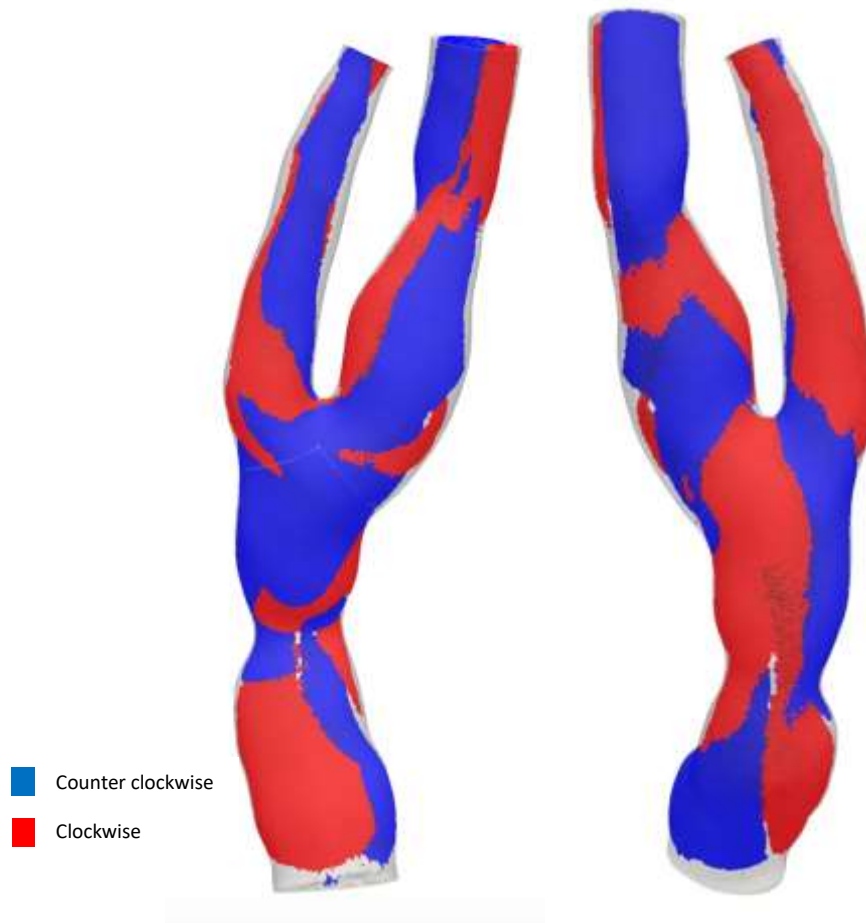
Figure 4.11: Case 3 - Flow structures derived from Mean LNH positive (counter clockwise) values (blue) and negative (clockwise) values (red).

HELICITY- BASED DESCRIPTORS			
	h1	h2	h3	h4
Case 4	2.822	15.21	0.186	0.186

The table shows that the Helicity Intensity (h2) consistently higher than the Mean Helicity (h1). This is also shown by h3 value 0.19.
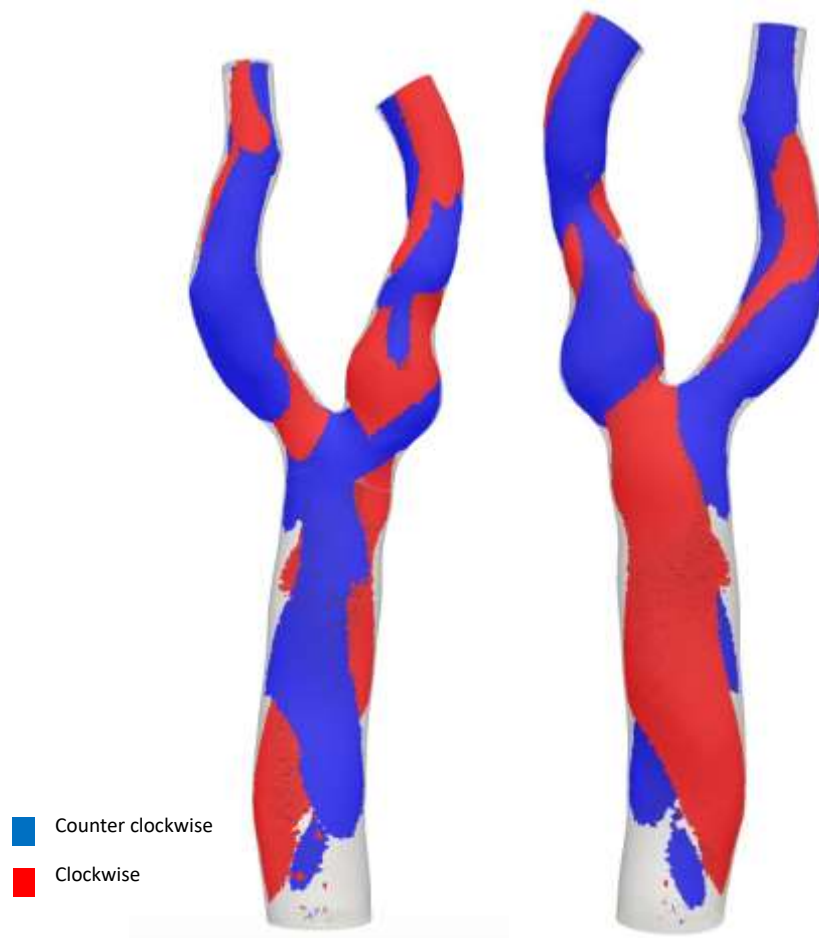
Figure 4.12: Case 4 - Flow structures derived from Mean LNH positive (counter clockwise) values (blue) and negative (clockwise) values (red).

4.2.2 Discussion

The value of 10th percentile for TAWSS matches with the value found in literature for indicating the possibility of stimulating a proatherogenic endothelial phenotype.
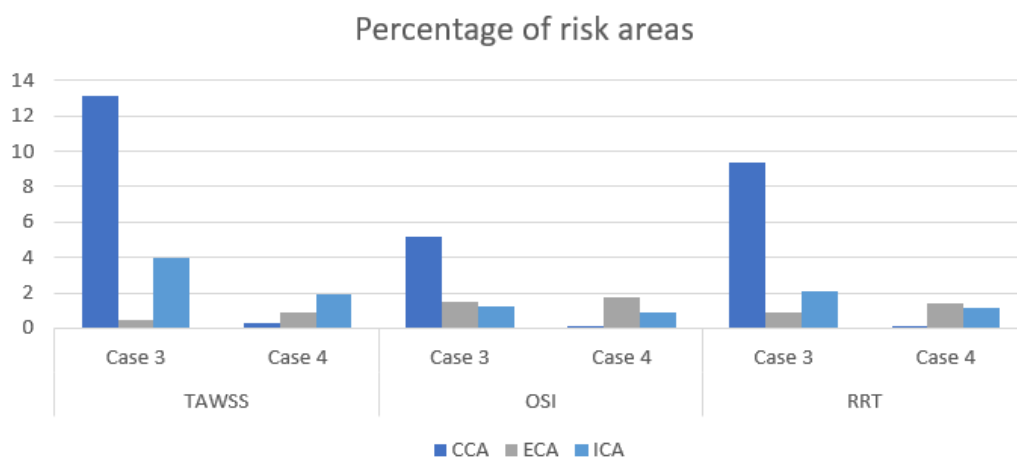
Chart 4.1: Percentage of risk areas for TAWSS, OSI and RRT.

As can be noticed form the chart above, the branch that present the wider area exposed to risk is the CCA for Case 3, while ECA and ICA for Case 4 present similar values of exposed areas.

The areas exposed to risk found analysing TAWSS and OSI are similar to those found analysing RRT since this parameter is directly proportional to OSI and inversely proportional to TAWSS.

As can be seen from Figures 4.11 and 4.12, the counter-rotating flow structures are mostly balanced; this is also proved by the value of h4.

Chapter 5
Conclusions

In this last chapter the conclusions about the thesis work are shown.

Carotid bifurcation stenosis is one of the most common cardiovascular diseases. There are many surgical techniques for try to restore the physiological lumen, the most popular are: atherectomy with patch and vessel stenting.

For these reasons the main purpose of this study was to compare two different techniques used to restore a physiological hemodynamic condition in patients affected by atherosclerotic plaque. This was guided by the purpose of preventing after surgery restenosis.

The study consists in a geometrical and hemodynamics analysis to characterize and better understand the behaviour of the patch atherectomy technique and compare the results with the ones of another study that analysed the stenting procedure [A. Sanino 2018].

Pre and post-operative Patient-specific models were reconstructed from CT scans and used for this study.

The geometrical analysis was carried on using a redefinition of the parameters introduced by Bijari *et al. 2012*.

Since low and oscillatory values of WSS can lead to the development of atherosclerosis in the zones where they occur, TAWSS, OSI and RRT have been analysed. The dimension of the risk areas with respect to the total area of the model are approximately the same both for the cases treated with stent and with patch. Another important hemodynamic descriptor is Helicity. Helical flow could be indeed used as a fluid dynamic signature for the identification of sites prone to fibro-intimal thickening and atherosclerotic plaque formation. To evaluate the balance between flow structures, helicity-based descriptors were calculated. For both techniques the results show a global balance of this structures.

In conclusion, from this work seems to emerge that the two techniques are similar in terms of after surgery hemodynamic. These conclusions can be confirmed, first of all, with the comparison of CFD results of preoperative and with the analysis of the possible variation of extension and location of risk areas. The only variation of the dimensions of risk areas, combined with a poor displacement, shows that the techniques prevents from the atherosclerotic-plaque formation after surgery. In order to demonstrate the similarity of the two techniques is necessary to expand the data set and to implement a statistic analysis of results. Concerning geometry, the comparison between pre- and postoperative geometries shows that endarterectomy plays a significant role in the variation of flare parameters, while the stenting procedure mostly influences tortuosity and curvature parameters. Endarterectomy affects geometry more than stenting; this can be addressed to the different procedure of the two techniques. As a matter of facts, during carotid stenting the plaque is not removed, on the other hand, the cutting of the vessel's wall and its consequent reconstruction applied during endarterectomy modify the carotid geometry. It is demonstrated that flare parameters are directly related to flow separations events and higher values of tortuosity and curvature are linked with disturbed blood-flow and the generation of secondary flows. It is also important to notice that high tortuosity can counteract the effects on flow induced by flare.

Future developments of this work primarily consist in the comparison with preoperative CFD results. Furthermore, is important to evaluate follow ups that are similar in terms of passed time after surgery in order to validate the comparison of the atherosclerotic processes and evolutions. A wider data set is necessary in terms of validation of the results of this work.

Appendix A – Flare Script

```python
#!/usr/bin/env python

import sys
import vtk
import math
import numpy as np
from vmtk import pypes
from vmtk import vmtkscripts
from vmtk import vtkvmtk

#################################### FUNCTIONS ######################################

#Function that reads an .stl file and returns polyData

def loadSTL(filenameSTL) :
        readerSTL=vtk.vtkSTLReader()
        readerSTL.SetFileName(filenameSTL)
        readerSTL.Update()
        polydata=readerSTL.GetOutput()
        if polydata.GetNumberOfPoints()==0 :
                raise ValueError("No point data could be loaded from")
                return None
        return polydata

#################################### OBTAINING FILES ##################################

##Stl to vtp conversion

command = 'vmtksurfacereader -ifile MS_C1.stl --pipe vmtksurfacewriter @vmtksurfacereader -ofile
surface.vtp'
arg = pypes.PypeRun(command)

##Clipping (press i once for every surface)

command = 'vmtksurfaceremeshing -ifile surface.vtp -area 0.05 -ofile surface.vtp --pipe
vmtksurfaceclipper -ifile surface.vtp -ofile surface.vtp'
arg = pypes.PypeRun(command)

##Centerlines

command = 'vmtkcenterlines -ifile surface.vtp -seedselector openprofiles -ofile centerlines.vtp'
arg = pypes.PypeRun(command)

##Flow Extensions

command = 'vmtkflowextensions -ifile surface.vtp -centerlinesfile centerlines.vtp -
adaptivelength 1 -extensionratio 5 -interactive 1 -ofile surface_extended.vtp --pipe
vmtksurfaceviewer -ifile surface_extended.vtp'
arg = pypes.PypeRun(command)

##Centerlines and CCA3

command = 'vmtksurfacereader -ifile surface_extended.vtp --pipe vmtkcenterlines -seedselector
openprofiles -resampling 1 -resamplingstep 0.005 -ofile centerlines.vtp --pipe
vmtkbranchextractor --pipe vmtkbranchclipper --pipe vmtkbifurcationsections -distancespheres 3
--pipe vmtkbranchclipper -i @vmtkbifurcationsections.o -groupids 0 -ofile CCA3.vtp --pipe
vmtksurfacewriter -ifile CCA3.vtp -ofile CCA3.dat -celldata 1'
arg = pypes.PypeRun(command)

##Remeshing (We remesh again our model with defined area element and then we want to compute
the volume of closed surface)

command = 'vmtksurfaceremeshing -ifile CCA3.vtp -elementsizemode "area" -area 0.005 -
preserveboundary 1 -ofile CCA3.vtp'
arg = pypes.PypeRun(command)

##ICA centerline
```

```
command = 'vmtkcenterlines -ifile surface_extended.vtp -seedselector openprofiles -resampling 1
-resamplingstep 0.005 -ofile ICA.vtp --pipe vmtkcenterlineattributes -ifile ICA.vtp -ofile
ICA.dat'
arg = pypes.PypeRun(command)

##ECA centerline

command = 'vmtkcenterlines -ifile surface_extended.vtp -seedselector openprofiles -resampling 1
-resamplingstep 0.005 -ofile ECA.vtp --pipe vmtkcenterlineattributes -ifile ECA.vtp -ofile
ECA.dat'
arg = pypes.PypeRun(command)

##Mean Centerline

with open ('ECA.dat','r') as f:
        c1=f.readlines()
c1=c1[1:]

with open ('ICA.dat','r') as f:
        c2=f.readlines()
c2=c2[1:]

l1=len(c1)
l2=len(c2)

if l1<l2:
        diff=l2-l1
        c2=c2[diff:]
        leng=l1
elif l1>l2:
        diff=l1-l2
        c1=c1[diff:]
        leng=l2

c_mean_x=[]
c_mean_y=[]
c_mean_z=[]
c1_x=[]
c1_y=[]
c1_z=[]
c2_x=[]
c2_y=[]
c2_z=[]
i=0

for i in range (leng):                  ##Extraction of the triplets x,y,z for the mean centerline
        c11=c1[i].split(" ")
        c1_x.append(float(c11[0]))
        c1_y.append(float(c11[1]))
        c1_z.append(float(c11[2]))
        c22=c2[i].split(" ")
        c2_x.append(float(c22[0]))
        c2_y.append(float(c22[1]))
        c2_z.append(float(c22[2]))
        mean_x=(float(c11[0])+float(c22[0]))/2
        c_mean_x.append(mean_x)
        mean_y=(float(c11[1])+float(c22[1]))/2
        c_mean_y.append(mean_y)
        mean_z=(float(c11[2])+float(c22[2]))/2
        c_mean_z.append(mean_z)

psource=[c_mean_x[0], c_mean_y[0], c_mean_z[0]]
ptarget=[c_mean_x[leng-1], c_mean_y[leng-1], c_mean_z[leng-1]]

index= len(c_mean_x)
points= vtk.vtkPoints()
points.SetNumberOfPoints(index)
lines= vtk.vtkCellArray()
lines.InsertNextCell(index)

for k in range(index) :                 ##Write the points in vtk
        points.SetPoint(k,c_mean_x[k],c_mean_y[k],c_mean_z[k])
        lines.InsertCellPoint(k)
polygon=vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)
```

```
writer=vtk.vtkXMLPolyDataWriter()
writer.SetFileName('centerline_mean.vtp') ##Saving the mean centerline
writer.SetInput(polygon)          ##Ceck the version on vtk, for more updated version
substitute SetInput with SetInputData
writer.Write()


command = 'vmtkcenterlineattributes -ifile centerline_mean.vtp -ofile centerline_mean.dat'
arg = pypes.PypeRun(command)


################
#SAMPLING#
#Resampling allows you to find intersection with RayCasting algorithm

length='0'
factor='0'
command = 'vmtkcenterlineresampling -ifile centerline_mean.vtp -length '+length+' -ofile
centerline_mean_resampled.vtp          --pipe          vmtkcenterlinegeometry          -ifile
centerline_mean_resampled.vtp -smoothing 1 -factor '+factor+' -outputsmoothed 1 -iterations 100
-ofile      centerline_mean_resampled.vtp     --pipe     vmtkcenterlineattributes     -ifile
centerline_mean_resampled.vtp -ofile centerline_mean_resampled.dat'
arg = pypes.PypeRun(command)
################


####################################                                         INTERSECTIONS
####################################

##CCA3 - Mean Centerline intersection

command = 'vmtksurfacereader -ifile CCA3.vtp --pipe vmtksurfacewriter @vmtksurfacereader -ofile
CCA3.stl'   ##Conversion of carotid model from .vtp to .stl
arg = pypes.PypeRun(command)

mesh = loadSTL("CCA3.stl")               ##Creation of an obbTree box containing the carotid
model
obbTreeMesh=vtk.vtkOBBTree()
obbTreeMesh.SetDataSet(mesh)
obbTreeMesh.BuildLocator()

with open ('centerline_mean_resampled.dat','r') as f:
      cm=f.readlines()
cm=cm[1:]

targetPointsIntersection=[]
i=0

for i in range (len(cm)-1):                        ##Ray Casting

      'https://pyscience.wordpress.com/2014/09/21/ray-casting-
            with-python-and-vtk-intersecting-linesrays-
with-surface-meshes/'
   cmm=cm[i].split(" ")
   psource=[float(cmm[0]), float(cmm[1]), float(cmm[2])]##The Ray Casting algorithm provides
the intersection points of a straight line with a surface. We implemented this algorithm for
every
   cmmm=cm[i+1].split(" ")                          ##couple of adiacent points along
the mean centerline.
   ptarget=[float(cmmm[0]), float(cmmm[1]), float(cmmm[2])]
   pointsVTKintersection = vtk.vtkPoints()
   code=obbTreeMesh.IntersectWithLine(psource, ptarget, pointsVTKintersection, None)
   pointsVTKIntersectionData=pointsVTKintersection.GetData()
   noPointsVTKIntersection=pointsVTKIntersectionData.GetNumberOfTuples()
   for idx in range (noPointsVTKIntersection):
       tup=pointsVTKIntersectionData.GetTuple3(idx)
       targetPointsIntersection.append(tup)

xs=tup[0]
ys=tup[1]
zs=tup[2]

##Carotid surface - Mean centerline intersection

command   =  'vmtksurfacereader  -ifile  surface_extended.vtp  --pipe  vmtksurfacewriter
@vmtksurfacereader -ofile surface_extended.stl'   ##Conversion of carotid model from .vtp to
.stl
arg = pypes.PypeRun(command)
```

```
mesh = loadSTL("surface_extended.stl")                    ##Creation of an obbTree box containing
the carotid model
obbTreeMesh=vtk.vtkOBBTree()
obbTreeMesh.SetDataSet(mesh)
obbTreeMesh.BuildLocator()


with open ('centerline_mean_resampled.dat','r') as f:
        cm=f.readlines()
cm=cm[1:]


targetPointsIntersection=[]
i=0


for i in range (len(cm)-1):                                ##Ray                        Casting
'https://pyscience.wordpress.com/2014/09/21/ray-casting-with-python-and-vtk-intersecting-
linesrays-with-surface-meshes/'
    cmm=cm[i].split(" ")
    psource=[float(cmm[0]), float(cmm[1]), float(cmm[2])] ##The Ray Casting algorithm provides
the intersection points of a straight line with a surface. We implemented this algorithm for
every couple
    cmmm=cm[i+1].split(" ")                               ##of adiacent points along the mean
centerline.
    ptarget=[float(cmmm[0]), float(cmmm[1]), float(cmmm[2])]
    pointsVTKintersection = vtk.vtkPoints()
    code=obbTreeMesh.IntersectWithLine(psource, ptarget, pointsVTKintersection, None)
    pointsVTKIntersectionData=pointsVTKintersection.GetData()
    noPointsVTKIntersection=pointsVTKIntersectionData.GetNumberOfTuples()
    for idx in range (noPointsVTKIntersection):
        tup=pointsVTKIntersectionData.GetTuple3(idx)
        targetPointsIntersection.append(tup)


xt=tup[0]
yt=tup[1]
zt=tup[2]


#####################################  CUTTING CCA  #####################################

##Reading centerline mean

with open('centerline_mean_resampled.dat', 'r') as f:
        cm=f.readlines()
cm=cm[1:]
l=len(cm)

cmx=[]
cmy=[]
cmz=[]
cma=[]
i=0

for i in range(l):
        cmm=cm[i].split(" ")
        cmx.append(float(cmm[0]))
        cmy.append(float(cmm[1]))
        cmz.append(float(cmm[2]))
        cma.append(float(cmm[16]))

##Reading CCA3 surface

with open ('CCA3.dat','r') as f:
        CCA3_dati=f.readlines()
CCA3_dati=CCA3_dati[1:2]
CCA3_dati=CCA3_dati[0].split(" ")
CCA3_area=CCA3_dati[11]                      ##Area CCA3 (index 11)
CCA3_area=float(CCA3_area)
CCA3_max_axis=CCA3_dati[13]            ##MaxSectionSize (index 13)
CCA3_max_axis=float(CCA3_max_axis)
CCA3_x=float(CCA3_dati[5])
CCA3_y=float(CCA3_dati[6])
CCA3_z=float(CCA3_dati[7])

##Cutting

dst=math.sqrt((xt - xs)**2 + (yt - ys)**2 + (zt - zs)**2) ##Distance source-target

distancetocenterlinemedia_s=[]
i=0
```

IV

```
for i in range (len(cm)):
    cmm=cm[i].split(" ")            ##Traformazione in numeri della riga iesima
    x=float(cmm[0])
    y=float(cmm[1])
    z=float(cmm[2])
    distance = math.sqrt((xs - x)**2 + (ys - y)**2 + (zs - z)**2)
    distancetocenterlinemedia_s.append(distance)

distance_min_s = np.amin(distancetocenterlinemedia_s)
indexs= np.argmin(distancetocenterlinemedia_s)

distancetocenterlinemedia_t=[]
i=0

for i in range (len(cm)):
    cmm=cm[i].split(" ")            ##Traformazione in numeri della riga iesima
    x=float(cmm[0])
    y=float(cmm[1])
    z=float(cmm[2])
    distance = math.sqrt((xt - x)**2 + (yt - y)**2 + (zt - z)**2)
    distancetocenterlinemedia_t.append(distance)

distance_min_t = np.amin(distancetocenterlinemedia_t)
indext= np.argmin(distancetocenterlinemedia_t)

cmx1=[]
cmy1=[]
cmz1=[]

if distance_min_t == 0:
        cmx1=cmx[indext:]
        cmy1=cmy[indext:]
        cmz1=cmz[indext:]

else:
        nD= math.sqrt((xs - cmx[indext])**2 + (ys - cmy[indext])**2 + (zs - cmz[indext])**2)
        if nD<dst:
                cmx1=cmx[indext:]
                cmy1=cmy[indext:]
                cmz1=cmz[indext:]
                cmx1.insert(0,xt)
                cmy1.insert(0,yt)
                cmz1.insert(0,zt)
        else:
                indext=indext+1
                cmx1=cmx[indext:]
                cmy1=cmy[indext:]
                cmz1=cmz[indext:]
                cmx1.insert(0,xt)
                cmy1.insert(0,yt)
                cmz1.insert(0,zt)

indexs=indexs-indext

if distance_min_s == 0:
        cmx1=cmx1[:indexs]
        cmy1=cmy1[:indexs]
        cmz1=cmz1[:indexs]

else:
        nD= math.sqrt((xt - cmx[indexs])**2 + (yt - cmy[indexs])**2 + (zt - cmz[indexs])**2)
        if nD<dst:
                cmx1=cmx1[:indexs]
                cmy1=cmy1[:indexs]
                cmz1=cmz1[:indexs]
                cmx1.append(xs)
                cmy1.append(ys)
                cmz1.append(zs)
        else:
                indexs=indexs+1
                cmx1=cmx1[:indexs]
                cmy1=cmy1[:indexs]
                cmz1=cmz1[:indexs]
                cmx1.append(xs)
                cmy1.append(ys)
                cmz1.append(zs)
```

```
index=len(cmx1)
points = vtk.vtkPoints()
points.SetNumberOfPoints(index)
lines = vtk.vtkCellArray()
lines.InsertNextCell(index)

for k in range(index) :              ##Write the points in vtk
     points.SetPoint(k,cmx1[k],cmy1[k],cmz1[k])
     lines.InsertCellPoint(k)

polygon=vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)
writer=vtk.vtkXMLPolyDataWriter()
writer.SetFileName('centerline_mean_cut.vtp') ##Saving the mean centerline
writer.SetInput(polygon)          ######Ceck the version on vtk, for more updated version
substitute SetInput with SetInputData#####
writer.Write()

command=        'vmtkcenterlineattributes      -ifile      centerline_mean_cut.vtp      -ofile
centerline_mean_cut.dat'
arg = pypes.PypeRun(command)

command=       'vmtkcenterlinesections      -ifile     surface_extended.vtp      -centerlinesfile
centerline_mean_cut.vtp -ofile CCA_sections_cut.vtp -ocenterlinesfile CCA_sections_cut.dat'
arg = pypes.PypeRun(command)



################################### CCA MAX  ###################################

with open ('CCA_sections_cut.dat', 'r') as f:
     CCA=f.readlines()
CCA3_i=len(CCA)-1
CCA=CCA[1:CCA3_i]       ##We don't want to consider CCA3
CCA_x=[]
CCA_y=[]
CCA_z=[]
CCA_area=[]
CCA_axis=[]
i=0

for i in range(len(CCA)):
     CCA_section=CCA[i].split(" ")
     CCA_x.append(float(CCA_section[0]))
     CCA_y.append(float(CCA_section[1]))
     CCA_z.append(float(CCA_section[2]))
     CCA_area.append(float(CCA_section[3]))
     CCA_axis.append(float(CCA_section[5]))
CCA_max_area=np.amax(CCA_area)
index=np.argmax(CCA_area)
CCA_max_axis=CCA_axis[index]
CCA_max_x=CCA_x[index]
CCA_max_y=CCA_y[index]
CCA_max_z=CCA_z[index]

################################### CUTTING CCA  ###################################

cmc=[]

with open ('centerline_mean_cut.dat', 'r') as f:
     cmc=f.readlines()
cmc=cmc[1:]

dst=math.sqrt((CCA_max_x - xs)**2 + (CCA_max_y - ys)**2 + (CCA_max_z - zs)**2)

cmcx=[]
cmcy=[]
cmcz=[]
abscissa=[]
distancetocenterlinemedia_m=[]
i=0

for i in range (len(cmc)):
     cmm=cmc[i].split(" ")            ##Traformazione in numeri della riga iesima
     x=float(cmm[0])
```

```
        y=float(cmm[1])
        z=float(cmm[2])
        asc=float(cmm[3])
        cmcx.append(x)
        cmcy.append(y)
        cmcz.append(z)
        abscissa.append(asc)
        distance = math.sqrt((CCA_max_x - x)**2 + (CCA_max_y - y)**2 + (CCA_max_z - z)**2)
        distancetocenterlinemedia_m.append(distance)

distance_min_m = np.amin(distancetocenterlinemedia_m)
indexm= np.argmin(distancetocenterlinemedia_m)
abscissacca3=abscissa[-1]

cmcx1=[]
cmcy1=[]
cmcz1=[]

if distance_min_m == 0:
        cmcx1=cmcx[indexm:]
        cmcy1=cmcy[indexm:]
        cmcz1=cmcz[indexm:]
        L=abscissacca3-abscissa[indexm]
        flag=1

else:
        flag=0
        print('Values may be wrong')
        nD= math.sqrt((xs - cmx[indexm])**2 + (ys - cmy[indexm])**2 + (zs - cmz[indexm])**2)
        if nD<dst:
                cmcx1=cmcx[indexm:]
                cmcy1=cmcy[indexm:]
                cmcz1=cmcz[indexm:]
                cmcx1.insert(0,CCA_max_x)
                cmcy1.insert(0,CCA_max_y)
                cmcz1.insert(0,CCA_max_z)
        else:
                indexm=indexm+1
                cmcx1=cmcx[indexm:]
                cmcy1=cmcy[indexm:]
                cmcz1=cmcz[indexm:]
                cmcx1.insert(0,CCA_max_x)
                cmcy1.insert(0,CCA_max_y)
                cmcz1.insert(0,CCA_max_z)

index=len(cmcx1)
points = vtk.vtkPoints()
points.SetNumberOfPoints(index)
lines = vtk.vtkCellArray()
lines.InsertNextCell(index)

for k in range(index) :                ##Write the points in vtk
        points.SetPoint(k,cmcx1[k],cmcy1[k],cmcz1[k])
        lines.InsertCellPoint(k)

polygon=vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)
writer=vtk.vtkXMLPolyDataWriter()
writer.SetFileName('centerline_mean_cut_final.vtp') ##Saving the mean centerline
writer.SetInput(polygon)              ######Ceck the version on vtk, for more updated version
substitute SetInput with SetInputData#####
writer.Write()

command=     'vmtkcenterlineattributes   -ifile   centerline_mean_cut_final.vtp   -ofile
centerline_mean_cut_final.dat'
arg = pypes.PypeRun(command)

command= 'vmtkrenderer --pipe vmtksurfaceviewer -ifile surface_extended.vtp -opacity 0.05 --
pipe  vmtksurfaceviewer  -ifile  centerline_mean_cut_final.vtp  -color  1  0  0  --pipe
vmtksurfaceviewer -ifile CCA3.vtp'
arg = pypes.PypeRun(command)

####################################  PARAMETERS  ####################################

##Flare A
```

```
flare_A=CCA_max_area/CCA3_area

##Flare M

flare_M=CCA_max_axis/CCA3_max_axis

##Flare R

if flag == 0:
        with open ('centerline_mean_cut_final.dat', 'r') as f:
                cmcf=f.readlines()
        cmcf=cmcf[1:]
        cmm=cmcf[-1].split(" ")
        L=float(cmm[3])

flare_R = ((CCA_max_axis - CCA3_max_axis)/(2*L))

##Print

print ('FLARE_A =' , flare_A)
print ('FLARE_M =' , flare_M)
print ('FLARE_R =' , flare_R)
print ('CCA3 Area =' , CCA3_area)

##Txt file

with open("Flare.txt","w") as out_file:

        out_string = ""
        out_string += str('FLARE_A')
        out_string += "=" + str(flare_A)
        out_string += "\n"
        out_string += str('FLARE_M')
        out_string += "=" + str(flare_M)
        out_string += "\n"
        out_string += str('FLARE_R')
        out_string += "=" + str(flare_R)
        out_file.write(out_string)
```

Appendix B – Tortuosity Script

```python
#!/usr/bin/env python

from __future__ import division, print_function
import sys
import vtk
import math
import numpy as np
import scipy
import matplotlib.pyplot as plt
import matplotlib as mpl
from mpl_toolkits.mplot3d import Axes3D
from vmtk import pypes
from vmtk import vmtkscripts
from vmtk import vtkvmtk
from scipy.interpolate import InterpolatedUnivariateSpline


################################### FUNCTIONS ###################################

#Function that reads an .stl file and returns polyData

def loadSTL(filenameSTL) :
        readerSTL=vtk.vtkSTLReader()
        readerSTL.SetFileName(filenameSTL)
        readerSTL.Update()
        polydata=readerSTL.GetOutput()
        if polydata.GetNumberOfPoints()==0 :
                raise ValueError("No point data could be loaded from")
                return None
        return polydata

#Function that detects peaks from
http://nbviewer.jupyter.org/github/demotu/BMC/blob/master/notebooks/DetectPeaks.ipynb

def detect_peaks(x, mph=None, mpd=1, threshold=0, edge='rising',
                kpsh=False, valley=False, show=False, ax=None):        #Function        that
detects peaks
        x = np.atleast_1d(x).astype('float64')
        if x.size < 3:
                return np.array([], dtype=int)
        if valley:
                x = -x
        # find indices of all peaks
        dx = x[1:] - x[:-1]
        # handle NaN's
        indnan = np.where(np.isnan(x))[0]
        if indnan.size:
                x[indnan] = np.inf
                dx[np.where(np.isnan(dx))[0]] = np.inf
        ine, ire, ife = np.array([[], [], []], dtype=int)
        if not edge:
                ine = np.where((np.hstack((dx, 0)) < 0) & (np.hstack((0, dx)) > 0))[0]
        else:
                if edge.lower() in ['rising', 'both']:
                        ire = np.where((np.hstack((dx, 0)) <= 0) & (np.hstack((0, dx)) > 0))[0]
                if edge.lower() in ['falling', 'both']:
                        ife = np.where((np.hstack((dx, 0)) < 0) & (np.hstack((0, dx)) >= 0))[0]
        ind = np.unique(np.hstack((ine, ire, ife)))
        # handle NaN's
        if ind.size and indnan.size:
                # NaN's and values close to NaN's cannot be peaks
                ind = ind[np.in1d(ind, np.unique(np.hstack((indnan, indnan-1, indnan+1))),
invert=True)]
        # first and last values of x cannot be peaks
        if ind.size and ind[0] == 0:
                ind = ind[1:]
```

```
        if ind.size and ind[-1] == x.size-1:
                ind = ind[:-1]
        # remove peaks < minimum peak height
        if ind.size and mph is not None:
                ind = ind[x[ind] >= mph]
        # remove peaks - neighbors < threshold
        if ind.size and threshold > 0:
                dx = np.min(np.vstack([x[ind]-x[ind-1], x[ind]-x[ind+1]]), axis=0)
                ind = np.delete(ind, np.where(dx < threshold)[0])
        # detect small peaks closer than minimum peak distance
        if ind.size and mpd > 1:
                ind = ind[np.argsort(x[ind])][::-1]  # sort ind by peak height
                idel = np.zeros(ind.size, dtype=bool)
                for i in range(ind.size):
                        if not idel[i]:
                                # keep peaks with the same height if kpsh is True
                                idel = idel | (ind >= ind[i] - mpd) & (ind <= ind[i] + mpd) \
                                & (x[ind[i]] > x[ind] if kpsh else True)
                                idel[i] = 0  # Keep current peak
                # remove the small peaks and sort back the indices by their occurrence
                ind = np.sort(ind[~idel])


        if show:
                if indnan.size:
                        x[indnan] = np.nan
                if valley:
                        x = -x
                _plot(x, mph, mpd, threshold, edge, valley, ax, ind)

        return ind


#Function useful for plotting from
http://nbviewer.jupyter.org/github/demotu/BMC/blob/master/notebooks/DetectPeaks.ipynb

def _plot(x, mph, mpd, threshold, edge, valley, ax, ind):
    """Plot results of the detect_peaks function, see its help."""
    try:
        import matplotlib.pyplot as plt
    except ImportError:
        print('matplotlib is not available.')
    else:
        if ax is None:
            _, ax = plt.subplots(1, 1, figsize=(8, 4))

        ax.plot(x, 'b', lw=1)
        if ind.size:
            label = 'valley' if valley else 'peak'
            label = label + 's' if ind.size > 1 else label
            ax.plot(ind, x[ind], '+', mfc=None, mec='r', mew=2, ms=8,
                    label='%d %s' % (ind.size, label))
            ax.legend(loc='best', framealpha=.5, numpoints=1)
        ax.set_xlim(-.02*x.size, x.size*1.02-1)
        ymin, ymax = x[np.isfinite(x)].min(), x[np.isfinite(x)].max()
        yrange = ymax - ymin if ymax > ymin else 1
        ax.set_ylim(ymin - 0.1*yrange, ymax + 0.1*yrange)
        ax.set_xlabel('Data #', fontsize=14)
        ax.set_ylabel('Amplitude', fontsize=14)
        mode = 'Valley detection' if valley else 'Peak detection'
        ax.set_title("%s (mph=%s, mpd=%d, threshold=%s, edge='%s')"
                     % (mode, str(mph), mpd, str(threshold), edge))
        # plt.grid()
        plt.show()

#################################                              OBTAINING               FILES
#################################


##Stl to vtp conversion

#command = 'vmtksurfacereader -ifile MS_C1ex.stl --pipe vmtksurfacewriter @vmtksurfacereader -
ofile surface_extended.vtp'
#arg = pypes.PypeRun(command)
```

```
##Clipping (press i once for every surface)

#command = 'vmtksurfaceremeshing -ifile surface.vtp -area 0.05 -ofile surface.vtp --pipe
vmtksurfaceclipper -ifile surface.vtp -ofile surface_clipped.vtp'
#arg = pypes.PypeRun(command)


##Centerlines

#command = 'vmtkcenterlines -ifile surface_clipped.vtp -seedselector openprofiles -ofile
centerlines.vtp'
#arg = pypes.PypeRun(command)


##Flow Extensions

#command = 'vmtkflowextensions -ifile surface_clipped.vtp -centerlinesfile centerlines.vtp -
adaptivelength 1 -extensionratio 1 -interactive 1 -ofile surface_extended.vtp --pipe
vmtksurfaceviewer -ifile surface_extended.vtp'
#arg = pypes.PypeRun(command)


##Centerlines and CCA3

#command = 'vmtksurfacereader -ifile surface_extended.vtp --pipe vmtkcenterlines -seedselector
openprofiles -resampling 1 -resamplingstep 0.005 -ofile centerlines.vtp --pipe
vmtkbranchextractor --pipe vmtkbranchclipper --pipe vmtkbifurcationsections -distancespheres 3
--pipe vmtkbranchclipper -i @vmtkbifurcationsections.o -groupids 0 -ofile CCA3.vtp --pipe
vmtksurfacewriter -ifile CCA3.vtp -ofile CCA3.dat -celldata 1'
#arg = pypes.PypeRun(command)


##Remeshing (We remesh again our model with defined area element and then we want to compute
the volume of closed surface)

#command = 'vmtksurfaceremeshing -ifile CCA3.vtp -elementsizemode "area" -area 0.005 -
preserveboundary 1 -ofile CCA3.vtp'
#arg = pypes.PypeRun(command)


##ICA centerline

#command = 'vmtkcenterlines -ifile surface_extended.vtp -seedselector openprofiles -resampling
1 -resamplingstep 0.005 -ofile ICA.vtp --pipe vmtkcenterlineattributes -ifile ICA.vtp -ofile
ICA.dat'
#arg = pypes.PypeRun(command)


##ECA centerline

#command = 'vmtkcenterlines -ifile surface_extended.vtp -seedselector openprofiles -resampling
1 -resamplingstep 0.005 -ofile ECA.vtp --pipe vmtkcenterlineattributes -ifile ECA.vtp -ofile
ECA.dat'
#arg = pypes.PypeRun(command)


##Mean Centerline

with open ('ECA.dat','r') as f:
      c1=f.readlines()
c1=c1[1:]

with open ('ICA.dat','r') as f:
      c2=f.readlines()
c2=c2[1:]

l1=len(c1)
l2=len(c2)

if l1<l2:
      diff=l2-l1
      c2=c2[diff:]
      leng=l1
elif l1>l2:
      diff=l1-l2
      c1=c1[diff:]
      leng=l2
```

```
c_mean_x=[]
c_mean_y=[]
c_mean_z=[]
c1_x=[]
c1_y=[]
c1_z=[]
c2_x=[]
c2_y=[]
c2_z=[]
i=0

for i in range (leng):                   ##Extraction of the triplets x,y,z for the mean centerline
        c11=c1[i].split(" ")
        c1_x.append(float(c11[0]))
        c1_y.append(float(c11[1]))
        c1_z.append(float(c11[2]))
        c22=c2[i].split(" ")
        c2_x.append(float(c22[0]))
        c2_y.append(float(c22[1]))
        c2_z.append(float(c22[2]))
        mean_x=(float(c11[0])+float(c22[0]))/2
        c_mean_x.append(mean_x)
        mean_y=(float(c11[1])+float(c22[1]))/2
        c_mean_y.append(mean_y)
        mean_z=(float(c11[2])+float(c22[2]))/2
        c_mean_z.append(mean_z)

psource=[c_mean_x[0], c_mean_y[0], c_mean_z[0]]
ptarget=[c_mean_x[leng-1], c_mean_y[leng-1], c_mean_z[leng-1]]

index= len(c_mean_x)
points= vtk.vtkPoints()
points.SetNumberOfPoints(index)
lines= vtk.vtkCellArray()
lines.InsertNextCell(index)

for k in range(index) :              ##Write the points in vtk
        points.SetPoint(k,c_mean_x[k],c_mean_y[k],c_mean_z[k])
        lines.InsertCellPoint(k)
polygon=vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)
writer=vtk.vtkXMLPolyDataWriter()
writer.SetFileName('centerline_mean.vtp') ##Saving the mean centerline
writer.SetInputData(polygon)                  ##Ceck the version on vtk, for more updated version
substitute SetInput with SetInputData
writer.Write()

command = 'vmtkcenterlineattributes -ifile centerline_mean.vtp -ofile centerline_mean.dat'
arg = pypes.PypeRun(command)

################
#SAMPLING#
#Resampling allows you to find intersection with RayCasting algorithm

length='0'
factor='0'
command = 'vmtkcenterlineresampling -ifile centerline_mean.vtp -length '+length+' -ofile
centerline_mean_resampled.vtp        --pipe        vmtkcenterlinegeometry        -ifile
centerline_mean_resampled.vtp -smoothing 1 -factor '+factor+' -outputsmoothed 1 -iterations 100
-ofile     centerline_mean_resampled.vtp     --pipe     vmtkcenterlineattributes     -ifile
centerline_mean_resampled.vtp -ofile centerline_mean_resampled.dat'
arg = pypes.PypeRun(command)

length='0.2'
factor='0.5'

command = 'vmtkcenterlineresampling -ifile ICA.vtp -length '+length+' -ofile ICA_resampled.vtp
--pipe vmtkcenterlinegeometry -ifile ICA_resampled.vtp -smoothing 1 -factor '+factor+' -
```

```
outputsmoothed 1 -iterations 100 -ofile ICA_resampled.vtp --pipe vmtkcenterlineattributes -ifile
ICA_resampled.vtp -ofile ICA_resampled.dat'
arg = pypes.PypeRun(command)
################

#####################################                                    INTERSECTIONS
#####################################

##CCA3 - Mean Centerline intersection

with open ('CCA3.dat','r') as f:
        CCA3_dati=f.readlines()
CCA3_dati=CCA3_dati[1:]

cca3px=[]
cca3py=[]
cca3pz=[]
i=0

for i in range (len(CCA3_dati)):
        cca3s=CCA3_dati[i].split(" ")
        x=float(cca3s[0])
        y=float(cca3s[1])
        z=float(cca3s[2])
        cca3px.append(x)
        cca3py.append(y)
        cca3pz.append(z)

CCA3_dati=CCA3_dati[0:1]
CCA3_dati=CCA3_dati[0].split(" ")
CCA3_x=float(CCA3_dati[5])
CCA3_y=float(CCA3_dati[6])
CCA3_z=float(CCA3_dati[7])

with open ('centerline_mean_resampled.dat','r') as f:
        cm=f.readlines()
cm=cm[1:]

cmx=[]
cmy=[]
cmz=[]
distancetocenterlinemedia=[]
i=0

for i in range (len(cm)):
        cmm=cm[i].split(" ")
        x=float(cmm[0])
        y=float(cmm[1])
        z=float(cmm[2])
        cmx.append(x)
        cmy.append(y)
        cmz.append(z)
        distance = math.sqrt((x - CCA3_x)**2 + (y - CCA3_y)**2 + (z - CCA3_z)**2)
        distancetocenterlinemedia.append(distance)

distancetocenterlinemedia=np.array(distancetocenterlinemedia)
distance_min = np.amin(distancetocenterlinemedia)
indexs= distancetocenterlinemedia.argmin()

##Carotid surface - Mean centerline intersection

command = 'vmtksurfacereader -ifile surface.vtp --pipe vmtksurfacewriter @vmtksurfacereader -
ofile surface.stl'    ##Conversion of carotid model from .vtp to .stl
arg = pypes.PypeRun(command)

mesh = loadSTL("surface.stl")                    ##Creation of an obbTree box containing the
carotid model
obbTreeMesh=vtk.vtkOBBTree()
obbTreeMesh.SetDataSet(mesh)
obbTreeMesh.BuildLocator()
```

VI

```
with open ('centerline_mean_resampled.dat','r') as f:
      cm=f.readlines()
cm=cm[1:]

targetPointsIntersection=[]

i=0
for i in range (len(cm)-1):                         ##Ray                    Casting
'https://pyscience.wordpress.com/2014/09/21/ray-casting-with-python-and-vtk-intersecting-
linesrays-with-surface-meshes/'
    cmm=cm[i].split(" ")
    psource=[float(cmm[0]), float(cmm[1]), float(cmm[2])] ##The Ray Casting algorithm provides
the intersection points of a straight line with a surface. We implemented this algorithm for
every couple
    cmmm=cm[i+1].split(" ")                              ##of adiacent points along the mean
centerline.
    ptarget=[float(cmmm[0]), float(cmmm[1]), float(cmmm[2])]
    pointsVTKintersection = vtk.vtkPoints()
    code=obbTreeMesh.IntersectWithLine(psource, ptarget, pointsVTKintersection, None)
    pointsVTKIntersectionData=pointsVTKintersection.GetData()
    noPointsVTKIntersection=pointsVTKIntersectionData.GetNumberOfTuples()
    for idx in range (noPointsVTKIntersection):
        tup=pointsVTKIntersectionData.GetTuple3(idx)
        targetPointsIntersection.append(tup)

xd=tup[0]
yd=tup[1]
zd=tup[2]

##CCA3 - ICA

command = 'vmtksurfacereader -ifile CCA3.vtp --pipe vmtksurfacewriter @vmtksurfacereader -ofile
CCA3.stl'   ##Conversion of carotid model from .vtp to .stl
arg = pypes.PypeRun(command)

mesh = loadSTL("CCA3.stl")                   ##Creation of an obbTree box containing the carotid
model
obbTreeMesh=vtk.vtkOBBTree()
obbTreeMesh.SetDataSet(mesh)
obbTreeMesh.BuildLocator()

with open ('ICA_resampled.dat','r') as f:
      ica=f.readlines()
ica=ica[1:]

targetPointsIntersection=[]
i=0

for i in range (len(ica)-1):                     ##Ray                          Casting
'https://pyscience.wordpress.com/2014/09/21/ray-casting-with-python-and-vtk-intersecting-
linesrays-with-surface-meshes/'
    icaa=ica[i].split(" ")
    psource=[float(icaa[0]),  float(icaa[1]),  float(icaa[2])]##The  Ray  Casting  algorithm
provides the intersection points of a straight line with a surface. We implemented this algorithm
for every
    icaaa=ica[i+1].split(" ")                              ##couple of adiacent points along
the mean centerline.
    ptarget=[float(icaaa[0]), float(icaaa[1]), float(icaaa[2])]
    pointsVTKintersection = vtk.vtkPoints()
    code=obbTreeMesh.IntersectWithLine(psource, ptarget, pointsVTKintersection, None)
    pointsVTKIntersectionData=pointsVTKintersection.GetData()
    noPointsVTKIntersection=pointsVTKIntersectionData.GetNumberOfTuples()
    for idx in range (noPointsVTKIntersection):
        tup=pointsVTKIntersectionData.GetTuple3(idx)
        targetPointsIntersection.append(tup)

xs=tup[0]
ys=tup[1]
zs=tup[2]
```

```
###################################                INFLECTION                POINT
###################################

with open('ICA_resampled.dat', 'r') as f:
      l_ICA=f.readlines()
l_ICA=l_ICA[1:]
l=len(l_ICA)


x_ICA=[]
y_ICA=[]
z_ICA=[]
asc_ICA=[]
i=0

for i in range(l):
      l_ICAq=l_ICA[i].split(" ")
      x_ICA.append(float(l_ICAq[0]))
      y_ICA.append(float(l_ICAq[1]))
      z_ICA.append(float(l_ICAq[2]))
      asc_ICA.append(float(l_ICAq[17]))

##Manually compute curve coordinate

h=np.column_stack((np.diff(x_ICA), np.diff(y_ICA), np.diff(z_ICA)))
hh=np.power(h,2)
i=0
hhh=[]
for i in range (len(hh)) :
      hh1=np.sum(hh[i,:])                    ##Point to point distance
      hhh.append(hh1)
hhh=np.sqrt(hhh)
#hhh=np.array(np.cumsum(np.column_stack(hhh)))      ##In one array we have all cumulative sum
of the distance for                                   every points and we add at the
begining one more point equal to                             zero          (origin
point).When we will find the point where we cut the
      centerline we have the distance of the centerline in that point.
                  Using the spline we reconstruct the curve every 5 points.
hhh=np.array(np.cumsum(hhh))
s=np.lib.pad(hhh,1,'constant',constant_values=0)
s=s[:len(s)-1]
l=s

##Call scipy cubic spline interpolator

f1=scipy.interpolate.InterpolatedUnivariateSpline(l, x_ICA)
f2=scipy.interpolate.InterpolatedUnivariateSpline(l, y_ICA)
f3=scipy.interpolate.InterpolatedUnivariateSpline(l, z_ICA)
xxs=f1(l)                                      #Compute the interpolation point
yys=f2(l)
zzs=f3(l)
d1sx=f1.derivative()                           #First derivate
d1sy=f2.derivative()
d1sz=f3.derivative()
dd1sx=d1sx(l)
dd1sy=d1sy(l)
dd1sz=d1sz(l)
d2sx=d1sx.derivative()                         #Second derivate
d2sy=d1sy.derivative()
d2sz=d1sz.derivative()
dd2sx=d2sx(l)
dd2sy=d2sy(l)
dd2sz=d2sz(l)
d3sx=d2sx.derivative()                         ##Third derivate
d3sy=d2sy.derivative()
d3sz=d2sz.derivative()
dd3sx=d3sx(l)
dd3sy=d3sy(l)
dd3sz=d3sz(l)
p1=np.multiply(dd1sy,dd2sz)-np.multiply(dd1sz,dd2sy)
p2=np.multiply(dd1sz,dd2sx)-np.multiply(dd1sx,dd2sz)
p3=np.multiply(dd1sx,dd2sy)-np.multiply(dd1sy,dd2sx)
```

```
##Curvature

num1=np.sqrt(np.power(p1,2)+np.power(p2,2)+np.power(p3,2))
den1=np.power(dd1sx,2)+np.power(dd1sy,2)+np.power(dd1sz,2)        ##num1= Module of the product
of the second and first derivate
den1=np.power(den1,1.5)                                          ##den1= Module of the first derivate
^3
curvature=np.divide(num1,den1)

##Torsion

num2=np.multiply(dd3sx,p1)+np.multiply(dd3sy,p2)+np.multiply(dd3sz,p3)
den2=np.power(p1,2)+np.power(p2,2)+np.power(p3,2)
torsion=np.divide(num2,den2)

##Plotting

dpi=75
fig=plt.figure(dpi=dpi,facecolor='#617f8a')
threeDPlot=fig.add_subplot(111,projection='3d')
fig.subplots_adjust(left=0.03,bottom=0.02,right=0.97,top=0.98)
mpl.rcParams['legend.fontsize']=10
threeDPlot.scatter(x_ICA,y_ICA,z_ICA,color='r')
threeDPlot.plot3D(xxs,yys,zzs, label='Curve_Fit',color='b',linewidth=1)
threeDPlot.legend()
#plt.show()
plt.figure('Figure 1')
plt.subplot(211)
plt.title('Curvature')
plt.plot(l,curvature)
plt.subplot(212)
plt.title('Tortion')
plt.plot(l,torsion,'r')
#plt.show()

##Computing the distance between the bifurcation and each peak. The minimum value will give us
the inflection point

torsion=abs(torsion)          #Because the torsion can be in different directions
ind = detect_peaks(torsion)   #Wa obtain the index of every peak, we must then search between
those points the inflection point (prob the max value) and

i=0
n=0
d=0

for i in range(len(ind)):
        j=ind[i]
        xp=x_ICA[j]
        yp=y_ICA[j]
        zp=z_ICA[j]
        distance=np.sqrt(np.power((xp-xd),2)+np.power((yp-yd),2)+np.power((zp-zd),2))
        if n==0:
                d = distance
                n = n+1
        else:
                if distance < d:
                        d = distance
                        xi = xp                #Inflection point cohordinates
                        yi = yp
                        zi = zp
                        indext = j

####################################  CUTTING ICA   ####################################

icax=[]
icay=[]
icaz=[]
distancetoica=[]
i=0
```

```
for i in range (len(ica)):
      icaa=ica[i].split(" ")
      x=float(icaa[0])
      y=float(icaa[1])
      z=float(icaa[2])
      icax.append(x)
      icay.append(y)
      icaz.append(z)
      distance= math.sqrt((xs - x)**2 + (ys - y)**2 + (zs - z)**2)
      distancetoica.append(distance)

distance_min = np.amin(distancetoica)
indexs= np.argmin(distancetoica)

ipcca3= math.sqrt((xi - xs)**2 + (yi - ys)**2 + (zi - zs)**2)
icax1=[]
icay1=[]
icaz1=[]

if distance_min == 0:
    icax1=icax[indexs:indext+1]
    icay1=icay[indexs:indext+1]
    icaz1=icaz[indexs:indext+1]

else:
    ipica= math.sqrt((xi - icax[indexs])**2 + (yi - icay[indexs])**2 + (zi - icaz[indexs])**2)
    if ipica<ipcca3:
         icax1=icax[indexs:indext+1]
        icay1=icay[indexs:indext+1]
        icaz1=icaz[indexs:indext+1]
        icax1.insert(0,xs)
        icay1.insert(0,ys)
        icaz1.insert(0,zs)
    else:
        indexs=indexs+1
        icax1=icax[indexs:indext+1]
        icay1=icay[indexs:indext+1]
        icaz1=icaz[indexs:indext+1]
        icax1.insert(0,xs)
        icay1.insert(0,ys)
        icaz1.insert(0,zs)

index=len(icax1)
points = vtk.vtkPoints()
points.SetNumberOfPoints(index)
lines = vtk.vtkCellArray()
lines.InsertNextCell(index)

for k in range(index) :              ##Write the points in vtk
      points.SetPoint(k,icax1[k],icay1[k],icaz1[k])
      lines.InsertCellPoint(k)

polygon=vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)
writer=vtk.vtkXMLPolyDataWriter()
writer.SetFileName('ICA_cut.vtp')       ##Saving the mean centerline
writer.SetInputData(polygon)            ##Ceck the version on vtk, for more updated version
substitute SetInput with SetInputData#####
writer.Write()

command= 'vmtkcenterlineattributes -ifile ICA_cut.vtp -ofile ICA_cut.dat'
arg = pypes.PypeRun(command)

#################################### PARAMETERS ####################################

##ICA cut datas

with open('ICA_cut.dat', 'r') as f:
      l_IC=f.readlines()
```

X

```
l_IC=l_IC[1:]
l=len(l_IC)

c_a=[]
i=0

for i in range (l):
      l_ICi=l_IC[i].split(" ")
      xc=float(l_ICi[0])
      yc=float(l_ICi[1])
      zc=float(l_ICi[2])
      abscissa=float(l_ICi[3])
      c_ai=[xc,yc,zc,abscissa]
      c_a.append(c_ai)

c_a=np.array(c_a)

##Tort 3D

dist=np.sqrt(np.power((xi-xs),2)+np.power((yi-ys),2)+np.power((zi-zs),2))
tort3D=(abscissa/dist)-1
print(abscissa)
print(dist)
print(tort3D)

##Best fit plane

R2=[]
R=c_a[:,0:3]

matrice=[R[:,0],R[:,1],np.ones(len(R))]
matrice=np.array(matrice)
matrice=np.transpose(matrice)
z=R[:,2]
z=np.array(z)
z=np.transpose(z)
coeff=np.linalg.lstsq(matrice,z)
coeff= coeff[0] ## the coefficients | z = coeff(1) * x + coeff(2) * y + coeff(3)

p=[0,0,float(coeff[2])]
n_1=[float(coeff[0]),float(coeff[1]),-1] ##normal vector
n_1=np.array(n_1)

####coeff(1) * x + coeff(2) * y - z + coeff(3) = 0
a=float(coeff[0])
b=float(coeff[1])
c=-1
d=float(coeff[2])

##Points pojection

proiezioni=[]
for i in range(len(R)):

    x=R[i,0]
    y=R[i,1]
    z=R[i,2]

    t=(-d-a*x-b*y-c*z)/(a**2+b**2+c**2)

    xp=float(a*t+x)
    yp=float(b*t+y)
    zp=float(c*t+z)
    rigamatrice = [xp ,yp, zp]

    proiezioni.append(rigamatrice)

proiezioni=np.array(proiezioni)

##Rotazione punti proiettati sul best fit plane, cosi' da ottenere punti da un piano 3D ad uno
2D, ove calcolare il Fit Circle
```

```
Coor_3D=proiezioni
N = len(Coor_3D)
origin = Coor_3D[0,:]
localz = np.cross(Coor_3D[1,:]-origin, Coor_3D[2,:]-origin)
#normalize it
unitz = localz/np.linalg.norm(localz,2)
#calculate local x vector in plane
localx = Coor_3D[1,:]-origin
unitx = localx/np.linalg.norm(localx,2)
#calculate local y
localy = np.cross(localz, localx)
unity = localy/np.linalg.norm(localy,2)
#assume transformation matrix
T1=np.array([unitx[0],unitx[1],unitx[2],np.zeros(1)])
T2=np.array([unity[0],unity[1],unity[2],np.zeros(1)])
T3=np.array([unitz[0],unitz[1],unitz[2],np.zeros(1)])
T4=np.array([origin[0],origin[1],origin[2],np.ones(1)])
T=np.column_stack((T1,T2,T3,T4))
C = np.column_stack((Coor_3D, np.ones(len(Coor_3D))))
Coor_2D = np.linalg.lstsq(T,np.transpose(C))[0]
Coor_2D = np.transpose(Coor_2D[:3,:])

_x= Coor_2D[:,0]
_y= Coor_2D[:,1]

##Cerchio che fitta i punti

a1= np.column_stack((_x,_y, np.ones(len(_x))))
a2=np.transpose(-np.array([np.power(_x,2)+np.power(_y,2)]))

a=np.linalg.lstsq(a1,a2)[0]
a0=a[0]
a1=a[1]
a2=a[2]
R=math.sqrt((a0**2+a1**2)/4 - a2)
curvatura_cerchio=float(1/R)                ## curvatura cerchio

indice=len(proiezioni)
points = vtk.vtkPoints()
points.SetNumberOfPoints(indice)
lines = vtk.vtkCellArray()
lines.InsertNextCell(indice)
for k in range(indice):
    riga=proiezioni[k]
    x=riga[0]
    y=riga[1]
    z=riga[2]
    points.SetPoint(k, x, y, z)
    lines.InsertCellPoint(k)

polygon = vtk.vtkPolyData()
polygon.SetPoints(points)
polygon.SetLines(lines)

writer = vtk.vtkXMLPolyDataWriter();
writer.SetFileName("proiezioni.vtp")
writer.SetInputData(polygon)
writer.Write()

#command = 'vmtkrenderer --pipe vmtksurfaceviewer -ifile surface_extended.vtp -opacity 0.25 --
pipe vmtksurfaceviewer -ifile ICA_cut.vtp --pipe vmtksurfaceviewer -ifile CCA3.vtp -color 1 0
0'
#arg = pypes.PypeRun(command)


command = ''' vmtkcenterlineattributes -ifile proiezioni.vtp -ofile proiezioniabscissa.dat'''
pype = pypes.PypeRun(command)

with open('proiezioniabscissa.dat', 'r') as f:
        proiezioniabscissa = f.readlines()  #generazione list da file input .dat
```

XII

```
proiezioniabscissa = proiezioniabscissa[1:]          #eliminare primo elemento della list (X Y Z
...)

L_proiezioni=[]
for i in range(len(proiezioniabscissa)):
    centerlineriga = proiezioniabscissa[i]        #estrazione righe della lista
    centerlineparole = centerlineriga.split()    #splittare le parole della riga

    x_ = centerlineparole[0]
    y_ = centerlineparole[1]
    z_ = centerlineparole[2]
    centerline3 = centerlineparole[3]             #quarta parola (abscissas)

    x_ =float(x_)
    y_ =float(y_)
    z_ =float(z_)
    centerline3 = float(centerline3)
    rigamatrice = [x_,y_,z_,centerline3]

    L_proiezioni.append(rigamatrice)

L_proiezioni= np.array(L_proiezioni)                  ## Path length della centerline proiettata,
per il Tort2D,

L_proiezioni_abscissa=L_proiezioni[(len(L_proiezioni)-1),3]

D_proiezioni=math.sqrt((L_proiezioni[0,0]   -   L_proiezioni[(len(L_proiezioni)-1),0])**2   +
(L_proiezioni[0,1]   -   L_proiezioni[(len(L_proiezioni)-1),1])**2   +   (L_proiezioni[0,2]   -
L_proiezioni[(len(L_proiezioni)-1),2])**2)

##Tort2D
TORT_2D = (L_proiezioni_abscissa/D_proiezioni)-1
print(L_proiezioni_abscissa)
print(D_proiezioni)

print ('INDICI')
print ('TORT_3D =' , float(tort3D))
print ('TORT_2D =' , float(TORT_2D))
print ('CURV_2D =' , curvatura_cerchio)

#### SALVATAGGIO IN FILE .TXT

with open("Tortuosity.txt","w") as out_file:

        out_string = ""
        out_string += str('TORT_3D')
        out_string += "=" + str(float(tort3D))
        out_string += "\n"
        out_string += str('TORT_2D')
        out_string += "=" + str(float(TORT_2D))
        out_string += "\n"
        out_string += str('CURV_2D')
        out_string += "=" + str(curvatura_cerchio)
        out_string += "\n"
        out_file.write(out_string)
```

Bibliography

[1] P. Buddiga, "Medscape," 20 August 2014. [Online]. Available: https://emedicine.medscape.com/article/1948510-overview#a2.

[2] "OER Services," [Online]. Available: https://courses.lumenlearning.com/suny-ap2/chapter/structure-and-function-of-blood-vessels/.

[3] U. Morbiducci, *BSCV - Politecnico di Torino,* 2017-2018.

[4] P. Maslak, "American Society of Hematology," 7 December 2004. [Online]. Available: https://imagebank.hematology.org/image/2673/rouleaux-formation--1?type=upload.

[5] "Kenhub," [Online]. Available: https://www.kenhub.com/en/dashboard.

[6] H. V. Carter and H. Grey, "Anatomy of the Human Body," 1918. [Online]. Available: https://en.wikipedia.org/wiki/Aortic_arch.

[7] "National Heart, Lung and Blood institute," [Online]. Available: https://www.nhlbi.nih.gov/health-topics/atherosclerosis.

[8] A. Ringer, "Mayfield Clinic," April 2018. [Online]. Available: https://www.mayfieldclinic.com/PE-CarotidStenosis.htm.

[9] M. Domanin, D. Bissacco, D. Le Van and C. Vergara, "Computational fluid dynamic comparison between patch-based and primary closure techniques after carotid endarterectomy.," *Jornal of vascular Surgery,* vol. 67, no. 3, pp. 887-897, 2018.

[10] N. Uemiya, C. Lee, S. Ishihara, F. Yamane, Y. Zhang and Y. Qian, "Analysis of restenosis after carotid artery stenting: Preliminary results using computational fluid dynamics based on three-dimensional angiography," *Journal of Clinical Neurosience: Official Journal of the Neurosurgical Society of Australasia,* vol. 20, no. 11, pp. 1582-7, 2013.

[11] "National Heart Lung and Blood Insitute (NIH)," 12 November 2013. [Online]. Available: https://en.wikipedia.org/wiki/Carotid_stenting#/media/File:Cad_stentplacement.jpg.

[12] M. H. Friedman, "Geometric Risk factors for arteriosclerosis," 1983.

[13] M. H. Friedman, O. J. Deters, F. F. Mark and G. M. Hutchins, "Arterial geometry affects hemodynamics A potential risk factor for atherosclerosis," *Atherosclerosis,* vol. 46, no. 2, pp. 225-231, 1983.

[14] A. Malek, S. Alper and S. Izumo, "Hemodynamic shear stress and its role in atherosclerosis.," *JAMA,* vol. 282, no. 21, pp. 2035-42, 1999.

[15] U. Morbiducci, A. Kok, B. Kwak, P. Stone, D. Steinman and J. Wentzel, "Atherosclerosis at arterial bifurcations: evidence for the role of haemodynamics and geometry.," *Thrombosis and Haemostasis,* vol. 115, no. 3, pp. 484-92, 2016.

[16] "VMTK," Orobix, [Online]. Available: www.vmtk.org.

[17] P. B. Bijari, L. Antiga, D. Gallo, B. A. Wasserman and D. A. Steinman, "Improved prediction of disturbed flow via hemodynamically-inspired geometric variables," *Journal of Biomechanics,* vol. 45, no. 9, pp. 1632-37, 2012.

[18] U. Morbiducci, *BMF - Politecnico di Torino,* 2016-2017.

[19] D. Gallo, G. Isu, D. Massai, F. Pennella, M. Deriu, R. Ponzini, C. Bignardi, A. Audenino and U. Morbiducci, "A Survey of Quantitative Descriptors of Arterial Flow.," 2014.

[20] C. Cheng, D. Tempel, R. Van Haperen, A. Van Der Baan, F. Groosveld, M. Daemen, R. Krams and R. De Crom, "Atherosclerotic lesion size and vulnerability are determined by patterns of fluid shear stress.," *Circulation,* vol. 113, no. 23, pp. 2744-53, 2016.

[21] D. Ku and D. Giddens, "Pulsatile flow in a model carotid bifurcation.," *Arteriosclerosis,* vol. 3, no. 1, pp. 31-9, 1983.

[22] D. Ku, D. Giddens, C. Zarins and S. Glagov, "Pulsatile flow and atherosclerosis in the human carotid bifurcation. Positive correlation between plaque location and low oscillating shear stress.," *Arteriosclerosis,* vol. 5, no. 3, pp. 293-302, 1985.

[23] C. Chiastra, S. Morlacchi, D. Gallo, U. Morbiducci, R. Càrdenes, I. Larrabide and F. Migliavacca, "Computational fluid dynamic simulations of image-based stented coronar bifurcation models.," *Journal of the Royal Society, Interface,* vol. 10, no. 84, 2013.

[24] Y. Hoi, B. Wasserman, Y. Xie, S. Najjar, L. Ferrucci, E. Lakatta, G. Geerstenblith and D. Steinman, "Characterization of volumetric flow rate waveforms at the carotid bifurcations of older adults.," *Physiological Measurement,* vol. 31, no. 3, pp. 291-302, 2010.

[25] D. Gallo, D. Steinman, P. Bijari and U. Morbiducci, "Helical flow in carotid bifurcation as surrogate marker of exposure to disturbed shear.," *Journal of Biomechanics,* vol. 45, pp. 2398-2404, 2012.

Ringraziamenti

Prima di tutto vorrei ringraziare il mio relatore prof. Umberto Morbiducci e il mio correlatore prof. Diego Gallo, per la possibilità fornitami nell'affrontare la ricerca in questo campo. Inoltre, vorrei anche ringraziare il prof. Michalis Xenos per la disponibilità durante il periodo di ricerca affrontato nell'università di Ioannina e il postdoc Raptis Tasos per il supporto fornito.

Ringraziamenti vanno anche alla phD Paola Tasso per l'aiuto e la pazienza avuta in questo progetto.

Vorrei anche ringraziare Andrea, collega ideale, paziente e volenteroso lungo questi mesi di lavoro.

Vorrei ringraziare i miei genitori, a loro devo tutto. Grazie per avermi supportato ogni giorno, in questo lungo cammino e per avuto la pazienza di incoraggiarmi e credere ogni giorno in me.

Vorrei anche ringraziare Andrea, collega ideale, paziente e volenteroso lungo questi mesi di lavoro.

Vorrei ringraziare mia sorella, fonte di sostegno familiare lontano da casa, esempio di gentilezza ed eleganza.

Vorrei anche ringraziare nonno Tore per avermi incoraggiato e aiutato nell'ambientarmi lontano da casa. Grazie per avermi insegnato ad essere umile, gentile e disponibile con tutti.

Vorrei anche ringraziare nonno Concetto per avermi insegnato che dal sacrificio si può creare un impero e renderlo fonte di sostentamento per tutta la famiglia.

Vorrei anche ringraziare le mie nonne, per essersi interessate ad ogni mio problema a distanza ed avere avuto la gentilezza di volermi incoraggiare sempre.

Vorrei ringraziare tutti i ragazzi del Borsellino, la mia seconda casa avuta in questi anni, per ogni giorno di risate passato insieme. Grazie per esserci stati ogni giorno.

Vorrei ringraziare Vittoria per la voglia di ascoltare i miei lamenti e per essermi stata vicina nell'ultimo periodo di questo lungo progetto di tesi.

Infine, vorrei ringraziare ogni vero amico che Torino mi ha regalato e con cui ogni giorno lontano da casa, lo hanno reso un giorno a casa. Grazie a Gigi, Gianclè, Eugenio, Gimmi, Danilo, Flavio e Mattia.