POLITECNICO DI TORINO

POLITECNICO DI TORINO

SCHOOL OF ENGINEERING

Master's degree course in Aerospace Engineering

Master's Degree Thesis

Cartesian Monolithic Models

2D Simulation of Type B Aortic Dissection

Supervisor
Prof. Domenic D'Ambrosio

Candidate
Matteo Fois

Internship Tutor at
Inria Research Center (Bordeaux)
Doct. Michel Bergman
Prof. Angelo Iollo

April 2020

Ad Alessia
Alla mia famiglia

Abstract

The main purpose of this thesis is that of presenting, from a deeply theoretical and exhaustive point of view, one of the milestones of team MEMPHIS's research activity at INRIA, Bordeaux, primarily concerning the employment of the so-called Cartesian Monolithic models in order to study fluid-structure interaction based phenomena, characterising several physical problems, and nowadays recognised as more and more interesting because of their own fascinating complexity. More specifically, as far as this thesis work is concerned, a type B aortic dissection case of study has been chosen to be tackled, therefore with the purpose of designing and performing a bidimensional and geometrically simplified numerical simulation, aimed at showing the versatility and effectivity of those above mentioned methods, when applied to the most various research fields, such as biomedical or bio-engineering ones.

Principale obbiettivo di questa attività di tesi è quello di presentare, da un punto di vista teoricamente approfondito ed esaustivo, uno dei punti cardine dell'attività di ricerca svolta dal team MEMPHIS di INRIA, Bordeaux, riguardante prevalentemente l'impiego di modelli cosiddetti cartesiani e monolitici, finalizzati allo studio dei fenomeni di interazione fluido-struttura caratterizzanti numerosi campi della fisica, ed oggigiorno sempre più oggetto di interesse dovutamente alla loro intrinseca ed affascinante complessità. Nello specifico, nell'ambito della presente tesi di laurea magistrale, si è scelto di affrontare un caso di studio riguardante una dissezione aortica di tipo B, realizzando a tale scopo una simulazione numerica di un caso bidimensionale e dalla geometria semplificata, al fine di mostrare la versatilità e l'efficacia dei metodi sopra indicati quando applicati ai campi più disparati, come quello della biomedicina e della bio-ingegneria.

Summary

The contents of the present master's thesis are organized as follows. In the first two chapters we aim to introduce those physiological and mathematical tools useful in order to devise and understand, afterwards, the numerical treatment and the model development concerning the tackled problem of a 2D simulation of the blood flowing within a dissected aortic channel. Specifically, the first chapter is devoted to the presentation of the physiological background regarding human cardiovascular system, basic hemodynamic knowledge and principles, as well as some of the most relevant examples related to numerical simulations of blood flows and aortic dissections found in the literature. Then, the second chapter is aimed at introducing and deriving all those mathematical methods and models needed in the following, starting from the fundamental formulation of the governing equations of fluid mechanics, to proceed, then, with their specialization for incompressible problems, which are the case when blood flows and low speeds are considered, and to finally conclude with the presentation of the so-called multiscale models, which are basically formulated in order to enrich our computational higher-dimensional domain, and to provide it with more suitable boundary conditions.

Furthermore, the third chapter of this thesis is dedicated to the derivation of the most commonly employed numerical methods concerning PDEs discretization, both in time and space, then focusing more deeply on the introduction of the most important method almost always adopted when dealing with incompressible problems: the fractional step projection method. With regard to this argument, a wide investigation covering the issue of boundary conditions treatment is carried out and clearly proposed to the readers, representing one of the hugest contributions of this work.

The fourth chapter of the present thesis includes the presentation of the newly devised powerful methods representing its main subject: fully eulerian and cartesian monolithic models. Within this part of the work, a brief overview of some widely adopted methods is provided for each category of the faced problems (freely moving interfaces localization, boundary conditions imposition on displacing and deforming immersed bodies, fluid-structure interaction equations' manipulation, etc.), and, moreover, a complete and exhaustive introduction is given for those specifically adopted for this thesis work, together with their numerical formulations. Lastly, a sum up of the implemented algorithm is proposed at the end of this chapter.

Within the last fifth chapter, all most relevant numerical results are presented. We begin with some simple test cases concerning a moving membrane invested by a stationary Poiseuille's flow, to proceed, then, with the more interesting case of a fixed but freely floating membrane immersed in a pulsating flow. Then, multiscale models are taken

into consideration, and the issue of boundary conditions choice and problem posedness is discussed, showing all related results. Finally, more complete models are proposed concerning moving external aortic walls capable of mimicking some relevant human vessels' features, and three complete and interesting final cases are reported, to conclude the work.

Lastly, after the conclusions, one can find two more appendixes regarding parallel computing and linear solvers, in particular Krylov's space solvers are presented, as they played a fundamental role in this work.

Acknowledgements

Writing this master's thesis has been a long, challenging, but enormously satisfying job, which allowed me to learn much more than I expected concerning my specific application field, as well as about my own growth as a man, enjoying the possibility they offered me when I accepted the internship position at Inria. Since the very beginning of my staying in Bordeaux, I felt a huge sense of excitement mixed together with anxiety and deep responsibility connected to the goal I was trying to reach: it was the first step out from my university, towards the real external world, where I was finally supposed to start building my own career, my professional personality, my working knowledge, ...

First of all, I owe this thesis to those persons who made my internship programme possible, and who continuously performed a scrupulous monitoring activity, allowing for my work to take its final shape, step by step, always providing sustaining, reference and advisement. This is the reason why I would really like to thank Professor Domenic D'Ambrosio, my internal tutor in Turin, who taught me the fundamentals of Computational Fluid Dynamics and who was able to transmit us the passion for this extremely interesting subject, proposing us in addition the open position at Inria for which I applied. Then, A second special thank goes to Professor Angelo Iollo, MEMPHIS team leader at Inria, and one of my external supervisors for this job, for his kindness and willingness always shown to me in any occasion during my period of stage, and to Michel Bergman, my second and primary external supervisor at Inria, to whom a special acknowledgement is reserved, since he is the author of the code I inherited when I started my internship experience at Inria, and he is the one who has always been helping and advising me during the whole duration of my stage, teaching me all needed methods developed by their team, and offering me also the chance to improve my programming skills by observing and studying their pioneer codes.

Secondly, a special thank is reserved for all persons I met in Bordeaux, and at the IMB in particular: among colleagues and friends, and my new lovely french family, I really wish to mention Stephanie, Mathilde, Arturo, Giulia, Mirko, Pietro, Sergio, , Silvia, Costanza, Giuliano, Cédrick, Floriane, Pedro, Lei, Luigi, Antoine, Sébastien, Guillaume, Gwladys, Océane, Anne-Sophie, Cynthia, Chiara and Erwan. And, of course, I cannot forget the guys of the Haut-Carré. In conclusion, it's been really a pleasure to work with you, to spend six months of my life all together, and to share this experience with such amazing people. I love you all.

Eccellente, terminati i convenevoli in lingua straniera, è finalmente giunto il momento di omaggiare anche chi, d'oltralpe, ha contribuito in maniera più o meno sostanziale al conseguimento di questo mio personale traguardo, per il quale, lo anticipo fin dal principio, sono profondamente debitore ad ogni signolo individuo abbia incontrato lungo questo interminabile, tortuoso e faticosissimo cammino, ma d'altra parte anche fonte di insostituibili ed indimenticabili emozioni, ricordi, gioie ed irripetibili momenti di condivisione. Giungere al termine di cinque lunghi anni di università, in una facoltà di ingegneria, e tentare di voltarsi indietro per sforzarsi di apprezzare il tragitto compiuto, dando un qualche senso di ordine a tutte le emozioni provate ed ai traguardi raggiunti, è impresa assai ardua, se non addirittura irrealizzabile: le esperienze e le prove superate insieme ai propri compagni, commilitoni, le ansie e le paure vinte grazie al sostegno immancabile di amici e parenti, le difficoltà fronteggiate nonostante il costante ed opprimente timore del fallimento, la pressione palpabile esercitata dal tempo che scorre veloce ed inesorabile, e ancora, l'incertezza per l'avvenire al termine di questo viaggio. Non esiste uomo o donna al mondo in grado di poter dipanare una simile matassa con le sue sole forze, ed è esattamente per questa ragione che, entro le poche righe qui presenti, tenterò di gratificare anche solo il minima parte il vostro inquantificabile lavoro di sostegno, di supporto, di fiducia, di fedeltà, di lealtà, di presenza, e di amore.

Prima fra tutte, riprendendo direttamente la dedica di questa tesi, ringrazio la mia intera famiglia, per quanto di eccezionale in questi anni ha saputo donarmi, sopportandomi e motivandomi incessabilmente, attraversando insieme a me ogni fase di questo infinito cammino ormai giunto al suo epilogo. Grazie, dunque, a mia madre e mio padre, al mio fratellone ed a tutti i miei nonni per essere sempre stati qui, siete tutti voi il mio tesoro più grande. Vi prego, non stancatevi mai di credere in me.

Proseguendo, sempre con riferimento alla dedica di questa tesi, ringrazio in particolar modo Alessia, presenza costante e fondamentale lungo tutto questo mio percorso, per essere stata sempre al mio fianco, giorno dopo giorno, per aver ascoltato pazientemente ogni mio sfogo, per aver sanato ogni mia crisi, per aver condiviso ogni successo, e per avermi ispirato costantemente, spronandomi a non arrendermi per nessuna ragione al mondo. Sei stata la mia forza, ed un grazie speciale lo dedico anche a tutta la tua meravigliosa famiglia. Vi porto nel cuore.

Come in tutte le battaglie più aspre, non possono certo mancare i compagni d'arme, coloro coi quali ho condiviso le gioie ed i dissapori più profondi direttamente in prima linea, e che non mi hanno mai abbandonato (seppur talvolta lo meritassi davvero, lo ammetto), spalleggiandomi ed incitandomi continuamente, fianco a fianco, tra alti e bassi, sempre a spada tratta. Insieme abbiamo studiato, abbiamo ripassato, abbiamo sclerato, abbiamo mangiato, abbiamo bevuto, abbiamo viaggiato, qualcuno ha addirittura sciato, ci siamo laureati una volta, ci siamo divisi, ed ora ci apprestiamo a condividere ancora quest'emozione, magari non quest'oggi, non tutti all'unisono, ma molto presto sarà così. Grazie ragazzi, Cristian, Davide, Mattia, Pietro, Giuseppe, Tommaso, Giacomo, Nicolò, Marco e tutti gli altri. Volevo sapeste che è stato un onore essere vostro compagno, e lo è ancor più oggi essere vostro collega. Non riuscirete mai a liberarvi di me.

Giuro, ho quasi terminato con questa manfrina strappalacrime. Tuttavia, viene proprio ora la parte più complicata, temevo questo momento fin dal primo giorno in cui

ho cominciato la stesura di questa tesi: gli amici di sempre. Due parole, innanzitutto, sono d'obbligo. Per tutta la vita mi sono sempre sentito irrimediabilmente fin troppo legato al mio territorio, un campanilista da manuale, si potrebbe dire. Fin dal momento in cui ho iniziato i miei studi a Torino, ho sempre pensato che un giorno sarei riuscito a trovare il mio posto nel mondo del lavoro in una città che mi consentisse di rientrare a casa mia, nella mia Biella, con una cadenza sufficientemente regolare. Il mio desiderio, in breve, era quello di avviare la mia carriera, pezzo dopo pezzo, senza dover rinunciare alla vita meravogliosa che tanto faticosamente mi ero costruito negli anni, e dalla quale non avrei mai potuto prescindere. Tutt'oggi, a valle di un'esperienza semestrale in Francia, la quale mi ha consentito di apprendere molto come uomo, come individuo, come studioso, ed anche come amico, mi sento di dire che questo mio sogno, talmente radicato e caparbio, ne sia risultato rinnovato, rinforzato, ma anche esteso a più ampi orizzonti. Non posso più esludere, ad occhi bendati, la possibilità di dovermi un giorno allontanare dal mio tanto amato territorio, di dover trascorrere nuove e più durature fasi della mia vita in altre realtà, ho imparato ad accettarlo. Ma la più grande e rassicurante novità è che ora sono certo che tutti voi, tutte le persone cui tengo di più al mondo, quelle da cui ho imparato tutto ciò so, con le quali ho diviso e condiviso tutto quel che avevo, ogni avventura e disavventura, ogni follia, ogni guaio, voi tutti sarete sempre pronti a riaccogliermi a braccia aperte, a percuotermi con la vostra solita indelicatezza, e ad ascoltarmi come se nemmeno un solo giorno fosse trascorso. Siete tanti, troppi per elencarvi tutti, ed è per questa ragione che includo ciascuno di voi in questo sintetico ma immenso pensiero: vi voglio bene ragazzi.

Contents

List of Tables

List of Figures

Chapter 1

Introduction and Physiological Background

1.1 Aortic Dissection

Among aortic diseases, aortic dissections are certainly the ones of most interest from the Fluid Mechanics point of view, especially because of their challenging involvement with the Fluid Structure Interaction field (hereafter FSI). In this first section of the chapter, we briefly aim to introduce this particular kind of pathology, focusing on the most spread symtomps and risk factors, and also the most common medical and surgical treatments normally associated with it.

1.1.1 Description and Classification

As a reference to the following sum up, one can consider the work [38]. In it, aortic dissections are described as a rare but life-threatening condition which can affect up to 4-6 cases over 100 000 people, but depending on their age, it could even reach up to 30 cases over the same total amount for people older than 65 years. This risky kind of syndrome consists of the formation of a tear on the inner wall of the aortic vessel, the largest of the entire human, directly connected to the left ventricle of the heart through the homonymous aortic valve and devoted to the transport of oxygenated blood tissues from that heart chamber to all the peripheral organs and tissues. When such a laceration appears on the inner layer of the aortic wall, commonly called the *intima layer*, the pumped blood is therefore allowed to flow through it and between the latter and the other aortic wall's layers, the *media layer* and finally the outer *adventitia layer*, as shown in picture 1.1.

Depending on the anatomical location of such tears along the intima wall's length, at least two classifications are possible to be made in order to enumerate the different known configurations of aortic dissections. The latter, as one can notice from picture 1.2, can be therefore divided in two groups, the first of which, according to the Standford system, distinguishes between:

- **type A dissection:** those involving also the ascending part of the aorta, that is from the aortic valve and up to the higher summit aortic arch, and then eventually

down along the descending aorta;

- **type B dissection:** those involving only the descending aorta, after the summit aortic arch.

On the other hand, another kind of classifications is possible, according to the DeBakey system, which distinguishes between:

- **type I dissection:** those involving both the ascending and the descending part of the aorta, and therefore the summit aortic branches too;
- **type II dissection:** those involving only the ascending aorta, and then letting the summit branches alone;
- **type III dissection:** those involving only the descending aorta, again excluding the summit branches.

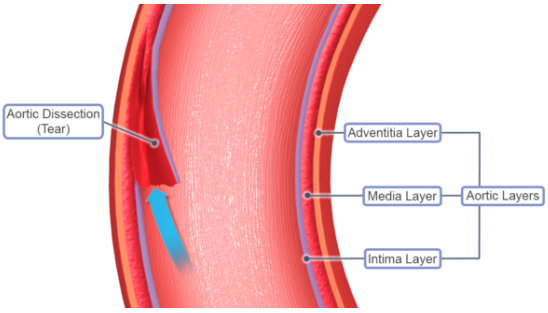


Figure 1.1: Formation of an intimal tear and, consequently, of an aortic dissection, reference at [1].

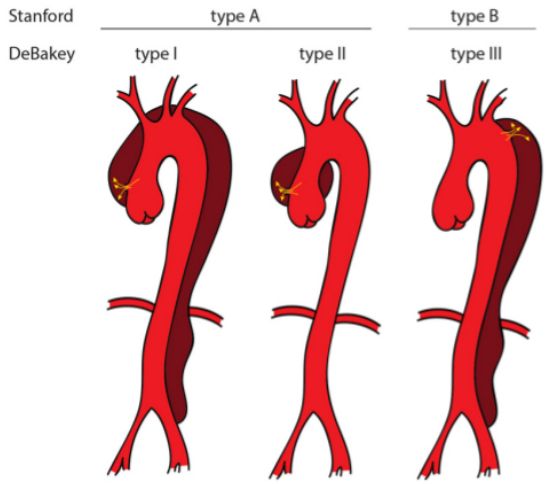


Figure 1.2: Classification of different aortic dissections configurations, reference in [38].

Depending on the time from the onset of the symptoms instead, another sub-classification is possible, which lists, in order of severity, acute dissections (presentation at the emergency department within one week), subacute dissections (from one week to one month) and chronic (more than one month).

Referring to the Standford classification, as far as the medical treatment is concerned, for each type of syndrome detected in a certain patient, most of acute type A cases are surgically repaired in order to immediately reduce the mortality risk, whereas those patients presenting a type B dissection are generally medically treated and monitored, unless worse complications occur, such as malperfusions, aortic regurgitation, cardiac failure or even, occasionally, stroke. Obviously, the main medical aim in order to reduce pain and the impact of this syndrome is to decrease as much as possible the wall shear stress, at least in the area surrounding the involved part of the aorta, generally by reducing blood pressure and heart rate. However, as it regards mortality of aortic dissection medical cases, it remains quite high, though: about 26% for acute type A dissection patient who receive surgical treatments, but up to 58% in the opposite case (because of advanced age or other motivations which inhibit surgery), while those values are significantly smaller for patients with type B dissections, for which the mortality rate could drop to 11%, although it could in turn still increase up to 31% again if complications occur.

1.1.2 Pathogenesis and Risk Factors

As it concerns the pathogenesis of such disease, intimal tears tend to appear in those areas of the inner aortic wall where the shear stress is the most intense. The blood flow through that laceration of the wall always leads to the formation of a second channel right beside the main aortic one, thus called *false lumen*, in contrast with the original so-called *true lumen*. Normally, this false lumen extends until a second tear appears, thus allowing the blood to come back to the true channel, linking them two again. In in the worst case, and especially in patients with type A dissection, the blood flowing within the false lumen could be capable of breaking both the remaining external layer of the vessel, thus leading the stream to leave the aorta and diffuse into the external media, with terrible consequences.

Among the most common risk factors which can lead to the occurrence of an aortic dissection, first one has to recognize that male cases are more probable than female ones, for both type A and B dissections, and in addition one has also to take into account the age of the subjects: the risk to have an aortic dissection increases as people become older, and the mean age at which that kind of disease mostly occurs is about 65 years, considering also that types A dissection usually involves younger subjects than type B one. Furthermore, patients with particular connective tissue malfunctions, or who exhibit a bicuspid aortic valve, are more exposed to the risk of facing such aortic troubles, even before 40 years of age. Aortic aneurysms often lead to aortic dissection's presentation, and the risk is also greater in those subjects who experienced cardiac surgery, or in those who suffer from other sicknesses such as atherosclerosis. The most important risk factor to be highlighted is, of course, hypertension, since it is present in about 80% aortic dissection cases. At last, also the aortic diameter plays an important role in the occurrence of this kind of pathology, and it seems that aortic dilatation could progressively increase the risk

of developing an aortic dissection.

It is now clear enough how much importance should be given to the timely detection of such aortic malfunctions, and this is exactly why the diagnosis process is one of the most important steps to this aim. Due to the randomness formation of an aortic dissection, in terms of its specific location all along the aortic channel, one needs a wide range of clinical alternatives in order to support the diagnosis process, which are now briefly listed below. One of the most spread symptoms associated with the involved patients is the sudden feeling of an intense and sharp chest or back pain, almost always followed by pulse deficit, murmur of aortic regurgitation, hypotension and so on, all clinically revealed in monitored patients. For those concerned with a type B dissection, especially, abdominal pain is more common. Whenever an aortic dissection was suspected, one should immediately proceed with the evaluation of the internal condition of the aortic vessel, throughout some devices and techniques which represent the state of the art of the present scanning and monitoring medical instruments. Among the latter, one can find X-ray scans, 12-lead electrocardiograms, transthoracic echocardiographies and, the most preferred in this sense, contrast-enhanced computed tomographies and magnetic-resonance imaging.

1.2 Fundamentals of Human's Cardiovascular System

Within this section we aim to give a quick but exhaustive presentation of the human cardiovascular system, in terms of its dynamics and its functional principles. In fact, it is essential for the well comprehension of this work to have a basic knowledge of the physiological behaviour of such system from the biological point of view, as well as from an engineering one, giving all the instruments and the mathematical tools commonly used to describe its functions and to classify its fundamental outcomes. All what it follows below is extracted from the work [71], presented by prof. S. Scarsoglio at Polytechnic of Turin as part of Aerospace Engineering master's classes about Biofluid Dynamics and Space Medicine.

1.2.1 Systemic and Pulmonary Circulations

The main function of the cardiovascular system is to deliver oxygen to all the body's tissues and organs, to exchange it with carbon dioxide via pulmonary respiration and, last but not least, to transport nutrients all over the human body, together with thermoregolation, fluid balancing, etc. The media allowing these substances to be transported and delivered is the blood, which is in turn moved by the heart, the central organ of the system which acts like a pump, sustaining the blood circulation over the whole human body, and pumping the fluid along all the vessels which compose the two main circulation compartments in which the entire network can be divided: the systemic circulation and the pulmonary one as shown in figure 1.3. While in the arterial compartments the blood flow is essentially pulsating, with relative pressure values comprised and oscillating between 80 and 120 mmHg in an healthy individual, as we proceed towards the smaller vessels this pulsating behaviour is dumped by the characteristic resistance of the vessels themselves, and by their elastic behaviour as well, therefore resulting in a continuous and non pulsating

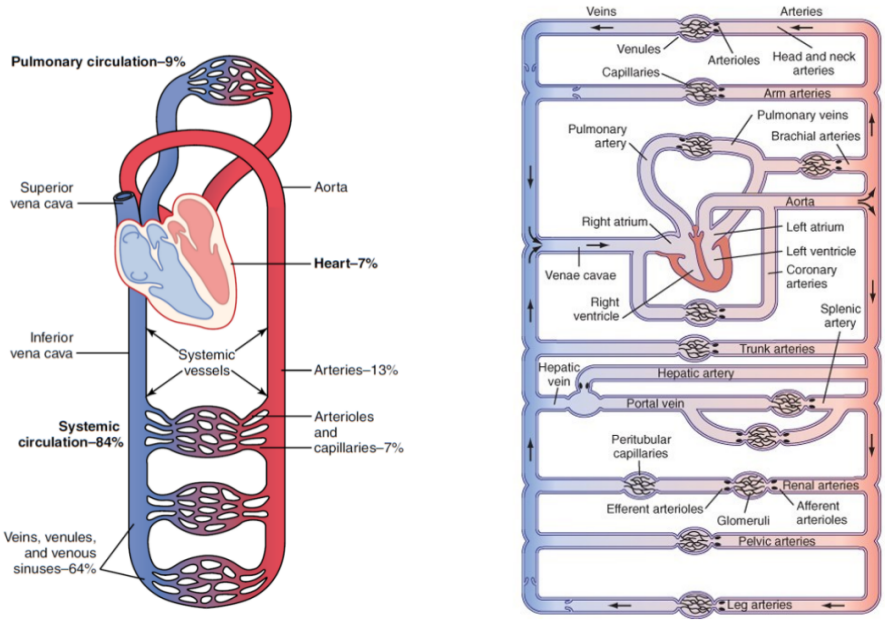


Figure 1.3: Systemic and pulmonary circulations.

flows in the venous compartments, with a much lower value of relative pressure. These fundamental features are reported in picture 1.4

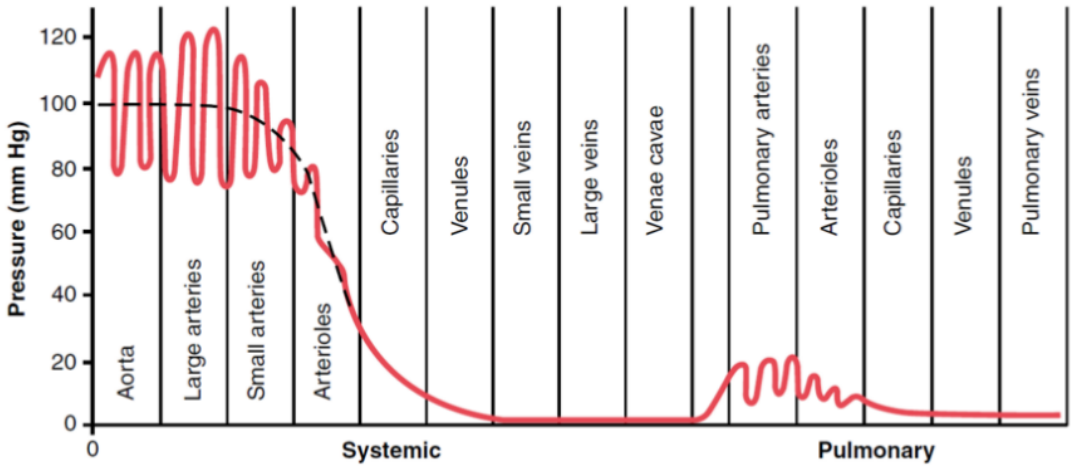


Figure 1.4: Pressure distribution over systemic and pulmonary circulations.

1.2.2 Heart and Blood

The main pump which guarantees the blood circulation described above, as stated previously, is the heart. This central and fundamental organ is essentially composed of four chambers, two of them are called atriums, and the other two are known as ventricles.

Each of these plays an important role in what is commonly called a *cardiac cycle*, typically composed of a first part denominated *systole*, and a second one called *diastole*. An entire cardiac cycle lasts in average about 0.80 s, which means that a healthy heart shows a *cardiac frequency* (or *heart rate*, HR) of about 75 bpm (beats per minute). Each atrium is connected to the relative ventricle through an intern valve (the *tricuspid* and the *mitral* valves), whereas the corresponding ventricle sends blood tissue towards either the aortic or the pulmonary arteries through their own *semilunar* valves, respectively the *aortic* and the *pulmonary* valve. As far as this work is concerned, we will focus only on the left ventricle dynamics, and on the linked behaviour of its aortic valve, since it allows the blood to flow into the aortic vessel, which we are mainly interested in. The temporal behaviour of the main hemodynamics variables for the left heart ventricle in reported in picture 1.5, which represents the well known Wiggers diagram for the left heart.

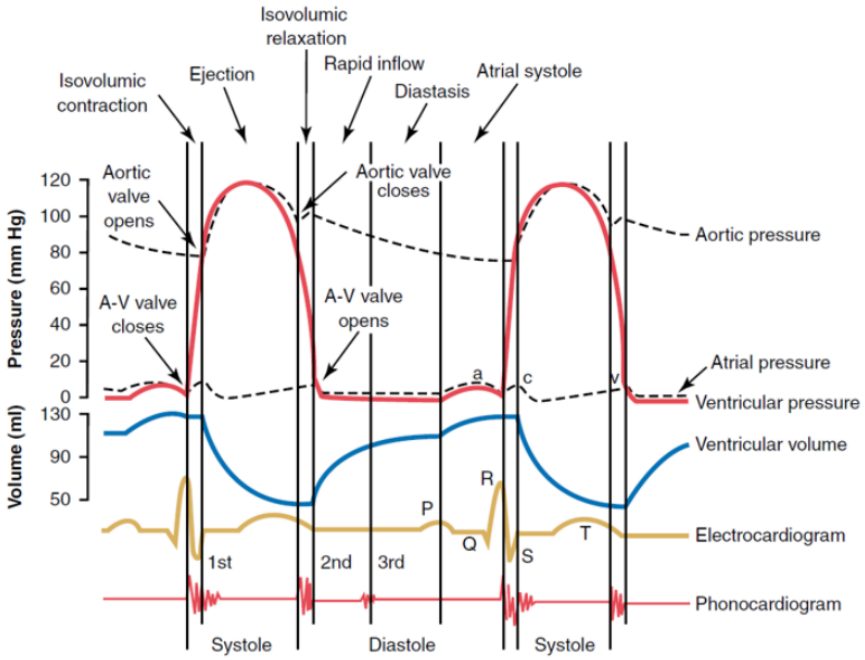
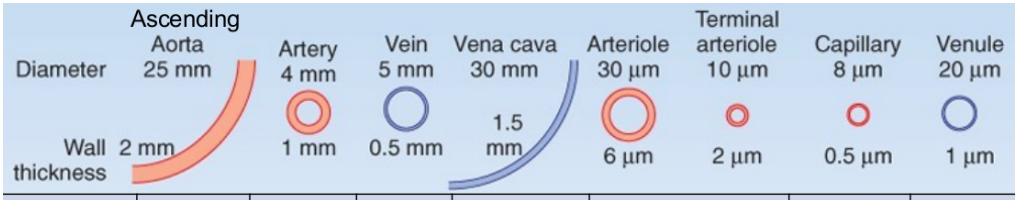


Figure 1.5: Wiggers diagram for the left heart.

Some of the main features of the arterial and venous systems are resumed and depicted in figure 1.6, which could be very useful for the settings of our numerical simulations, presented and then carried out in the following parts of the present work. Moreover, in order to provide useful information for the settings of the numerical simulations related to this work, in table 1.1 are summarised and listed some physical properties of the fluid which will be involved in all of them: the blood.



Length	5 cm	10 cm	3.5 cm	30 cm	0.15 cm	0.06 cm	0.15 cm
Tot section area	5 cm^2	20 cm^2	40 cm^2	18 cm^2	400 cm^2	4500 cm^2	4000 cm^2
Reynolds* (peak)	1500 (9400)	200 (1000)	100	1400 (3000)	0.1	0.001	0.01
Velocity (peak)	15 cm/s (110)	10 cm/s (50)	5 cm/s	20 cm/s (50)	0.7 cm/s	0.07 cm/s	0.3 cm/s

Figure 1.6: Main arterial and venous flows properties.

Table 1.1: Blood properties.

Property	**Symbol**	**Value**
Density	$\rho\,[kg\,m^{-3}]$	1.06×10^3
Dynamic Viscosity	$\mu\,[Pa\,s]$	4.0×10^{-3}

1.3 Numerical Simulations in Literature

A lot of different examples of numerical simulations are present in literature concerning the subject of hemodynamics, and some of them have been chosen and reported in this section to show their main and most characteristic results and features. Performing a numerical simulation of blood flow into a human vessel is not only a mere demonstration of the power and of the wideness of applicability of Computational Fluid Dynamics' tools, but instead it could even and above all represent a very effective monitoring instrument at medical disposal, since it could help in predicting whenever a case of aortic aneurysm should lead to the formation of an aortic dissection, or vice versa, or furthermore it could be useful in treating patients affected by type B aortic dissections, predicting their development when the tear's position and dimensions are known (see [23] for further details in this sense). In the perspective of realising a numerical simulation of the blood flow into a vessel, which consists of what is commonly known as Computational Hemodynamics, a big amount of efforts have been dedicated to those simulations regarding the very physiological problem we are particularly interested in, that is, aortic dissections. In this section are therefore reported those which have been considered the most relevant features and settings of such numerical simulations, which could turn out to be very useful and enlightening for our specific work. With this aim, this review section is therefore organised so that we present the main results obtained by the most relevant authors who faced the problem, for each feature of such simulation which needs to be set, that are:

- Geometry Definition;
- Fluid Model;

- Mesh Generation;
- Boundary Conditions.

Thus, starting from the choice of the model's geometry, we now focus on the investigation of the most relevant solution adopted in literature to set up such numerical hemodynamics simulations, under different physiological conditions.

1.3.1 Geometry Definition

The first feature to specify whenever a numerical simulation has to be carried out is the geometry one would like to assign to his problem, and consequently to his computational domain. On this choice will depend indeed any other feature of such simulation, from the generation of the computational mesh, up to the mathematical model adopted to mimic the physical phenomenon onto the domain itself. The first global distinction which is possible to be done refers to the geometrical space in which one wants to carry out his simulation, in other words it is now necessary to choose if the problem will be either 2 - dimensional or 3 - dimensional.

With regard to 2D simulations, it is clear enough that these simulations will not be that realistic, as no human vessel could ever exhibit such a geometrical configuration. These cases of study will therefore be the simplest and the least meaningful in terms of the actual physiological behaviour of the human cardiovascular system, but on the other hand, adopting some specific geometrical hypothesis such as the axisymmetry of the domain, whenever it should be possible, the resulting outcomes could turn out to be as useful as the ones from other more specific 3D simulations. This is not our case, obviously, since aortic dissections are anything but axisymmetric, although some interesting results could be found for example in [81], on the subject of aortic aneurysms, which could in turn give rise to aortic dissections, and vice versa, as presented in [75]. For instance, some results reported in the work cited above are shown in figure 1.7. In addition, and above all, the current work will focus right on a 2D case simulation, as it will be introduced in the following chapters.

Among the wide range of 3D simulations easily available in literature, one first needs to distinguish between the two fundamentals ways of defining a vessel's geometry, that is whether to represent an ideal geometry, basically consisting of a pipe containing the blood flow, with several outflow subsections if necessary, or instead a more realistic domain, based on specific reconstruction of an actual human vessel, conducted by those techniques listed in the previous section 1.1. To clearly describe each of this two possible alternatives, we will discuss them separately:

- **Ideal Geometry:** there are few examined cases belonging to this class of geometries, due to their relative less importance in comparison to the more complicate but detailed Patient-Specific ones , but some fundamental results could be collected even from these simpler simulations. They can be used for example as a preliminary study for the subsequent mesh generation, as reported in [12], or to perform other simulations with different and more general purposes, for instance those involving extracorporeal circulation, as shown in [13]. The geometries depicted in figure 1.8 are taken from the two works cited above.

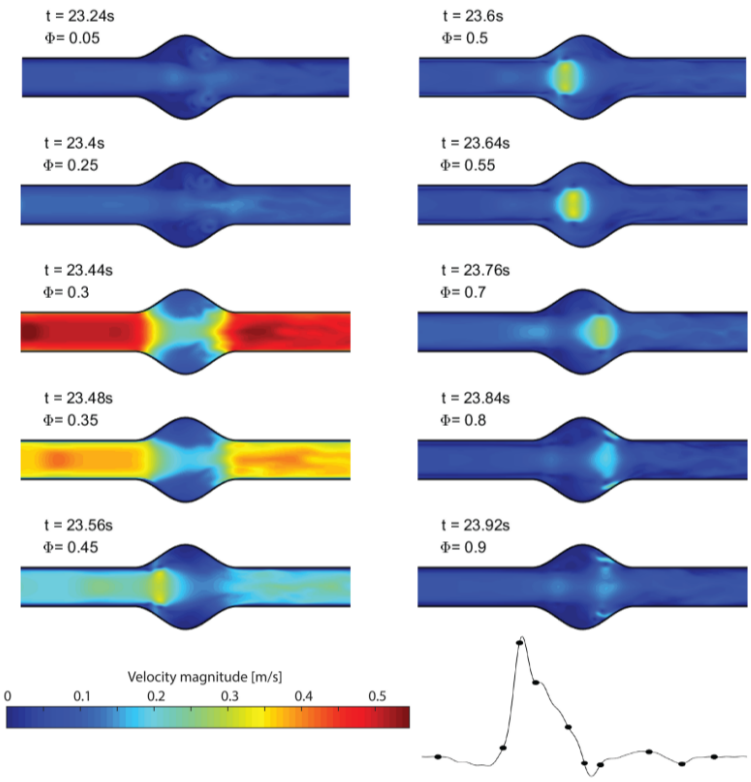


Figure 1.7: Example of 2D simulation of an aortic aneurysm (ϕ describes the cardiac cycle's completion).

- **Patient-Specific Geometry:** undoubtedly the most appealing class of geometric representations, to which belong almost all the simulations found in the literature (see cases presented in [23, 22, 75, 18, 12, 64, 83, 37, 17, 81, 39]). The main advantage is the fact of having a realistic, detailed and human-faithful aortic domain reconstruction where to perform our numerical simulation, which thus will be as exhaustive as possible, but as main drawbacks it is necessary to consider the extremely specific meaning of such simulations, which will refer just and only to the patient whose aortic channel has been virtually reconstructed this way, and the difficulties to face in order to obtain such geometries themselves throughout the medical devices mentioned in section 1.1, that are mainly PC-MRI (Phase-Contrast Magnetic Resonance Imaging) and CT (Computed Tomography), to begin with. As examples of these latter kinds of geometries, in figures 1.9 and 1.10 some cases of aortic domain are shown, virtually created either via PC-MRI or CT techniques, some of which are also exhibiting a type B aortic dissection configuration with the two typical lumen described in section 1.1.

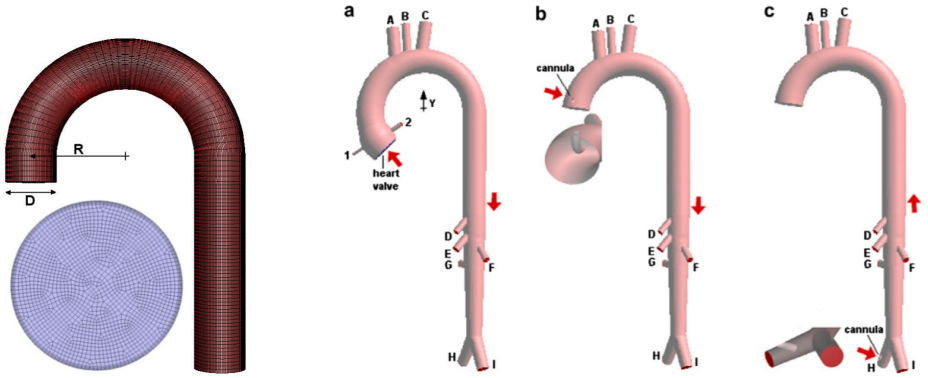


Figure 1.8: Ideal geometries defining an aortic channel.

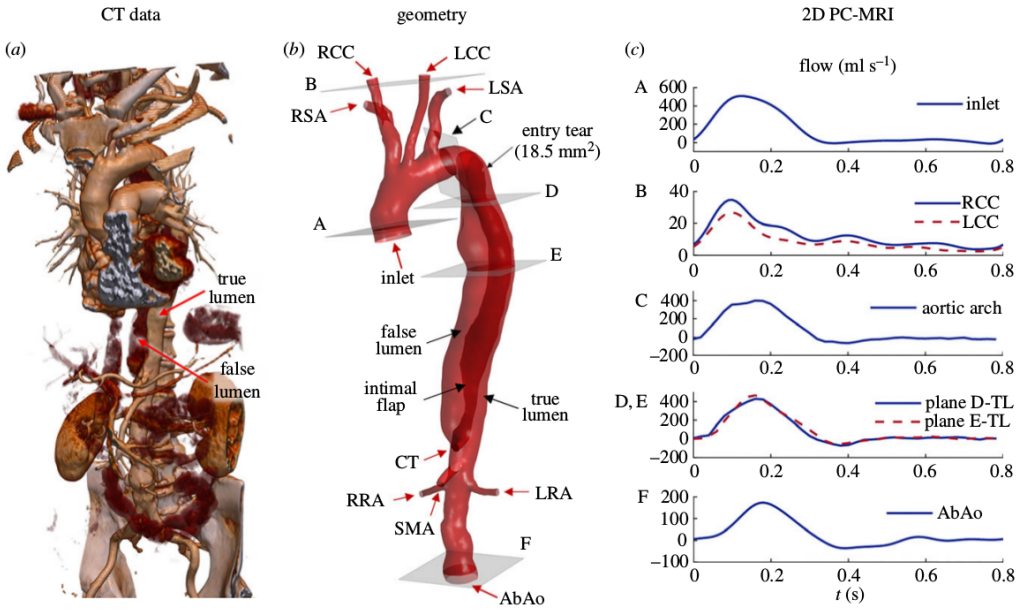


Figure 1.9: Patient-Specific geometries defining an aortic channel obtained via CT or PC-MRI scanning (first example).

1.3.2 Fluid Model

The second aspect which requires special care with the aim of setting up a numerical simulation is to establish a model for the fluid involved. As discussed previously, blood will be the actual fluid to mimic, with respect to its physical parameters, listed in table 1.1, and to its given stream properties. Depending on the vessel in which it is actually flowing, shown in figure 1.6, some features are easy to be highlighted. Since stream velocities characterising all different configurations of human vessels are never too high, compared to the speed of sound, the adopted model will always be considered as an incompressible one.

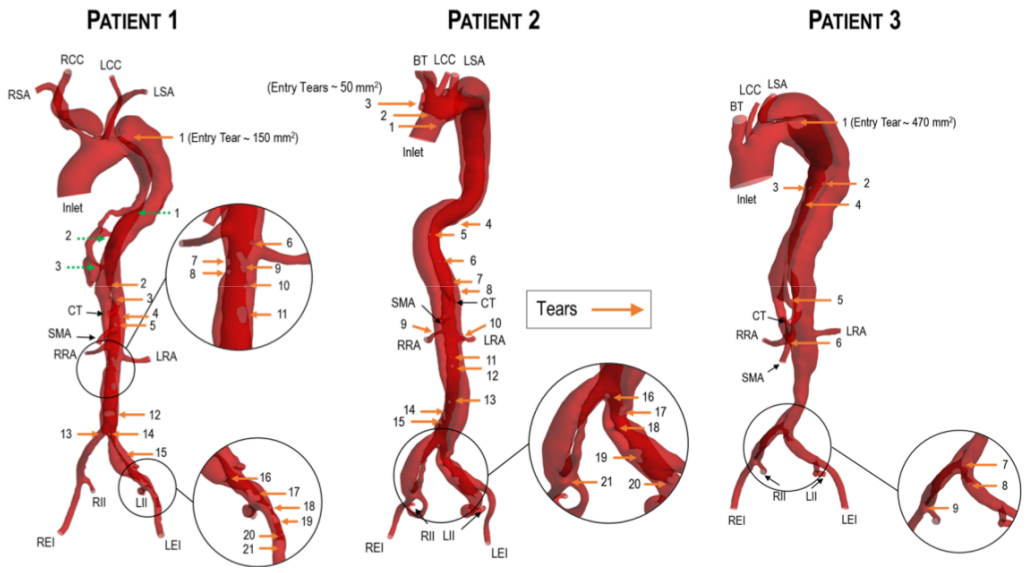


Figure 1.10: Patient-Specific geometries defining an aortic channel obtained via CT or PC-MRI scanning (second example).

For a deep analysis of the different kinds of fluid models and their respective consequences on the simulations, as well as their own requirements to be exhaustively defined, one could refer to the work [55]. The latter puts in evidence three major aspects concerning a numerical simulation involving blood flow: the choice between the use of a newtonian or a non-newtonian fluid model, the imposition of proper boundary conditions which allow to take into account wave reflections and all the effects related to the remaining part of the downstream cardiovascular system (an issue deeply faced later), and also the precise estimation of energy losses due to the complicate geometries which human vessels could show. In general, most of the works concerning blood flows tend to consider newtonian fluid models, in order to ensure the highest simplicity, but for different solutions regarding non-newtonian viscosity models, such as the so-called *Carreau-Yasuda* model, the *Power Law* or even the *Casson* model, one can refer to [18], to [17], and also to [39]. See figure 1.11 for a comparison. Another issue related to the definition of a proper fluid model is that regarding its laminar or turbulent behaviour in the context of an aortic flow: examples of application of different turbulence models can be found in works [12] and [13], in which the use of the so-called *Shear Stress transport* model is shown, or again in [17] for other different ones.

Finally, the last distinction regards the temporal model adopted for the simulation, that is either to choose a steady-states or pulsating stream model: the former could be useful in case of extracorporeal circulation (see [12] and [13]) or to provide preliminary outcomes useful for example to allow the setting up of the parameters involved in multi-scale models (see [18]), widely discussed later. Some results are shown in figure 1.12.

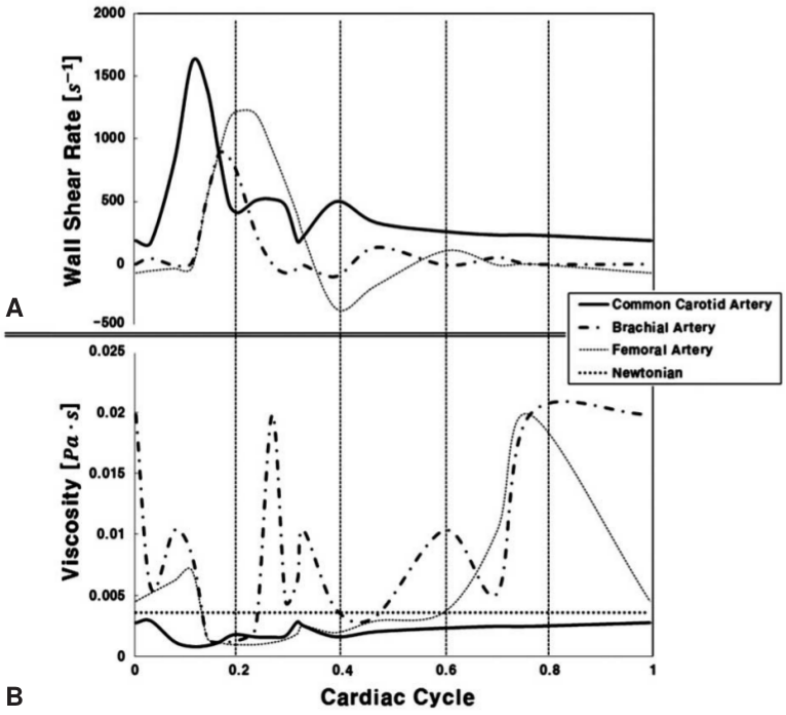
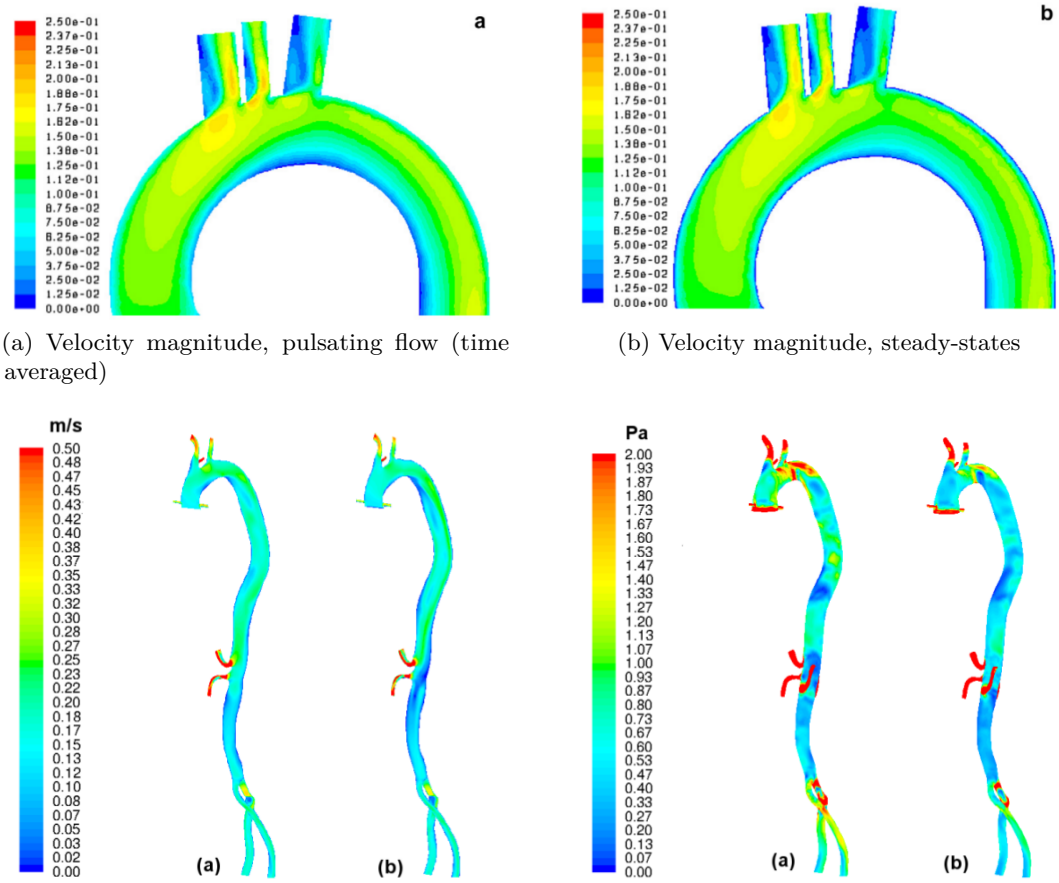


Figure 1.11: Comparison between newtonian and non-newtonian (Yarreau-Casuda) viscosity models for different human vessels, over a cardiac beat.

1.3.3 Mesh Generation

All the considered works rely on a Finite Volume Method discretization in order to achieve their numerical results, due to their huge flexibility, their immense adaptivity even to very complicated geometries and, above all, because of their well appreciated property of conservativity. Therefore, non structured grids are usually adopted and generated into the computational domain of interest, with cells exhibiting different shapes and extended up to cover any single detail of the reconstructed vessel, together with its branches, bifurcations, or external interventions. The generally adopted cell's geometries can include hexahedric, tetrahedric or even prismatic configurations, with these latter typically used to cover those regions very close to the walls of the domain, i.e. the ones where boundary layers will rise, characterised by their strong gradients confined in very thin layers, usually regarding few cells, which have necessarily to be as small and regular as possible, in order to ensure the best spatial resolution and, thus, the minimal numerical error affecting the solution. An example of wall mesh refinement is reported in [22], in which the authors suggest to use not less than at least 10 prismatic cells to cover the region just beside the channel's wall, in order to describe the stream's boundary layer with the best accuracy. To this aim, a refinement guideline is also given, stating the upper bound for the cell's height, which should not overcome the maximal value of 2 wall units, that is:

$$y^+ < 2$$



(a) Velocity magnitude, pulsating flow (time averaged)

(b) Velocity magnitude, steady-states

(c) Velocity magnitude, pulsating (left) and steady-states (right)

(d) Pressure distribution, pulsating (left) and steady-states (right)

Figure 1.12: Velocity magnitude and pressure distributions over an aortic summit arch ideal recontruction (a and b) and over an entire patient-specific aortic channel (c and d).

where

$$y^+ = \frac{y}{\delta_v} = y\frac{u_\tau}{\nu} = \frac{y}{\nu}\sqrt{\frac{\tau_w}{\rho}}$$

with

$$\tau_w = \tau(x=0) = \mu\frac{\partial u}{\partial y}\bigg|_{y=0}$$

Some examples of body-fitting and unstructured mesh as those described above are reported in figure 1.8, already presented in the previous section, and in figure 1.13.

Some words deserve to be spent with regard to this kind of mesh: body-fitting, unstructured and not-moving in time. Computational hemodynamics often deals with problems and domains characterised by complex interactions between the fluid involved in the studied phenomenon and the structure itself within which it is confined. For this reason, those behaviours should not be ignored, but on the other hand, taking them

into consideration would require an additional computational cost which could in turn compromise the entire feasibility of the simulation. In fact, considering the interaction exerted by the fluid on the structure would mean computing a displacement field for the latter, thus leading to a new geometrical configuration, and therefore to a completely new solution inside the domain, and so all over again. At each time step, a new mesh has to be generated since the geometrical configuration has changed, in order to obtain the new solution by time integration of the governing equations. This would represent what is more commonly denoted as a non-conservative problem. In the context of Fluid-Structure Interaction, which we are mainly concerned with, one needs to specify either a displacement model for the structure representing the deforming domain under the action of the fluid contained in it, or to find a new and powerful method, correlated with some numerical tool, able to represent it effectively.

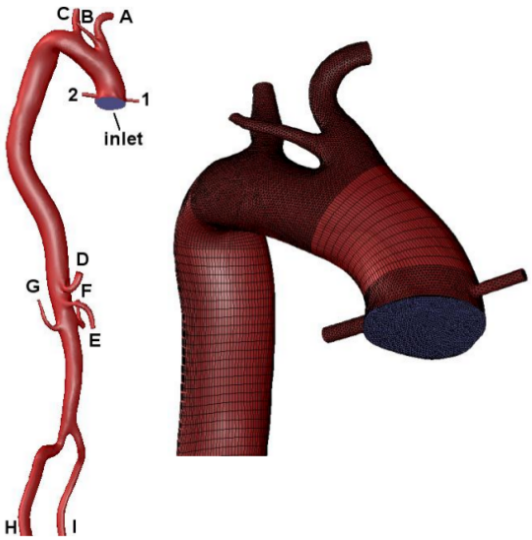


Figure 1.13: Example of unstructured and body-fitting mesh on a patient-specific aortic geometry.

1.3.4 Boundary Conditions

This last part of the section dedicated to the wide review of the collected works on the subject of hemodynamics numerical simulations is probably the most important and crucial, since it provides the missing information needed in order to set up consistent and faithful mathematical models, able to effectively represent the phenomenon of flowing blood into human vessels, even in presence of an aortic dissection. Whenever such a simulation needs to be performed, one has to provide for it appropriate boundary conditions, which allow the model to be said as *well posed*, that is, it must have a unique solution, which in turn must depend continuously on the data assigned to the problem itself (an exhaustive definition of a *well posed* problem could actually be found in any book on mathematical modeling, numerical analysis, applied mathematics, etc.). Furthermore, once imposed

appropriate boundary conditions for the considered problem, one needs also to deal with their compatibility with the imposition of the so-called initial condition, another issue which has to be taken into account in order to ensure the well posedness of the defined problem. There are really a lot of different settings available in the literature to be considered as suggestions to impose proper boundary conditions for numerical simulations concerning such hemodynamics problems, and some of the most popular will therefore be reported here. To clearly let the reader understand all different possibilities, the following main classification is adopted, based on the region of the physical domain where such conditions will be applied:

- Inflow section;
- Walls;
- Outflow section(s).

Inflow section: Different solutions are proposed in the works by the considered authors, ranging from the imposition of well known, traditional velocity profiles as inlet spatial distributions (Poiseuille profile, more complicate Womersley profile, polynomial or flat profiles, see for example [13]), to more realistic and faithful distributions directly extracted from patient-specific data via PC-MRI scanning, with all the related complications already discussed due to the intrusivity of such medical techniques (see for this case [18] and [37]). Oftentimes, they only apply a measured flow rate, extracted from its waveform which is obtained again via MRI technique, and prescribe it to a traditional profile as simple but realistic inflow conditions. To resume all the possibilities in this sense, one could refer to the much more than exhaustive work [83], whose main results state that there is no need to require such a complicate image scanning in order to impose a specific profile shape as inflow boundary condition, unless the specific individual we are interested in suffers from some particular disease concerning the aortic valve, thus influencing strongly the flow ejected by the heart into the aortic channel.

Walls: the issue concerning the choice of the most appropriate boundary condition to be applied on aortic walls, either considering them as outer or inner walls, that is those corresponding to the intima layer membrane now floating in the middle of the channel, under the condition of aortic dissection, and so separating the true lumen from the false lumen, is not as problematic as for the other regions of the domain. Indeed, having considered a model of realistic fluid, and therefore of a viscous fluid, the application of a no-slip condition is the only possible solution, representing both a straightforward and clear way to proceed. On the other hand, the imposition of such condition onto moving or deforming walls and surfaces, in case of a fluid-structure interaction problem, could be no longer as easy as for the previous case.

Outflow section(s): to this last division it will be given the most of the attention, due to its relevant importance with respect to the well posedness of a numerical simulation based on the related mathematical model. On this particular choice the authors have

introduced most of the alternatives today present in the literature, and still adopted in order to set up solid and consistent models, meant to provide faithful simulations. Following the review work presented in [64], here are listed the main possibilities suggested by the different authors with respect to the imposition of appropriate boundary conditions on the outflow section of the domain, or on all of them, if there are more than only one exit section, such as in the case of aortic branches starting from the central aortic vessel. In table 1.2 one can find a useful summary of the various configurations presented.

Table 1.2: List of various possible outflow boundary conditions suggested by the authors.

Boundary Condition	**Pros**	**Cons**
Constant pressure [75, 22, 64]	Simple to implement, good results in terms of velocity distribution, flow rates and wall shear stress.	Very bad in terms of pressure distribution, nonsense in presence of multiple outflow sections.
Flow rate/pressure via PC-MRI [12, 64]	Easy to implement, good results when applied to the smaller outflow sections of branches.	Hard to measure from in-vivo patients, excessive intrusivity.
Stress-free condition [37]	Easy to implement, good results in general.	Still not providing information about wave reflections from the downstream system.
Loss coefficient model [13]	Easy to implement, good results in general.	Still not providing information about wave reflections from the downstream system.
Windkessel models [17, 18, 20, 39, 64]	Excellent results in terms of physiological faithfulness, providing information about wave reflection Reference in [17]from downstream.	Quite hard to set up and tune, necessity of additional simulations to be properly calibrated.

As shown in table 1.2, there are various and extremely different possibilities regarding outflow boundary conditions, but not all of them will turn out to be suitable for the problem we aim to deal with. Taking as first example the easiest way to proceed, that is, imposing a constant and uniform value of pressure on the whole outflow section, typically a zero-pressure condition, one would find out that the numerical results obtained that way are not that far from those coming from PC-MRI scans, for example, in terms of velocities and shear stress distributions, but as far as the pressure field is concerned, what one would see is a completely unfaithful distribution of that variable, especially in the presence of multiple exit sections such as aortic branches. That condition in practise consists of a free-exit condition, as if the human vessel was directly communicating with the external environment, but also other pressure values corresponding to different situations are possible to be set. One needs to keep in mind the fact that bad representations of pressure fields inside the aortic domain, from the fluid structure interaction point of view,

would mean a completely distorted prediction of its displacement field, thus leading to not realistic solutions in time.

Another pair of possibilities which are advised by the cited authors is that regarding the prescription of either a flow rate time dependent value on each exit section of the domain, or a uniform pressure one (thus no longer constant in time), both extracted from their respective waveforms made available thanks to PC-MRI scans, once more. This would therefore represent an extremely valid and pertinent solution, but on the other hand, such waveform information are too much patient-specific and hardly measurable as well. Then, two new kinds of outflow boundary conditions follow, which exhibit the advantage of being totally new, easy-implementing, extremely flexible and surprisingly effective. The first of these two is the well known *stress-free condition*, also reported in some works as the *do-nothing condition* (see also [19] for deeper details), which basically consists of eliminating all the stress tensor terms from the motion equations, thus imposing something which could read like:

$$p\boldsymbol{n} - \mu\nabla\boldsymbol{V}\cdot\boldsymbol{n} = 0$$

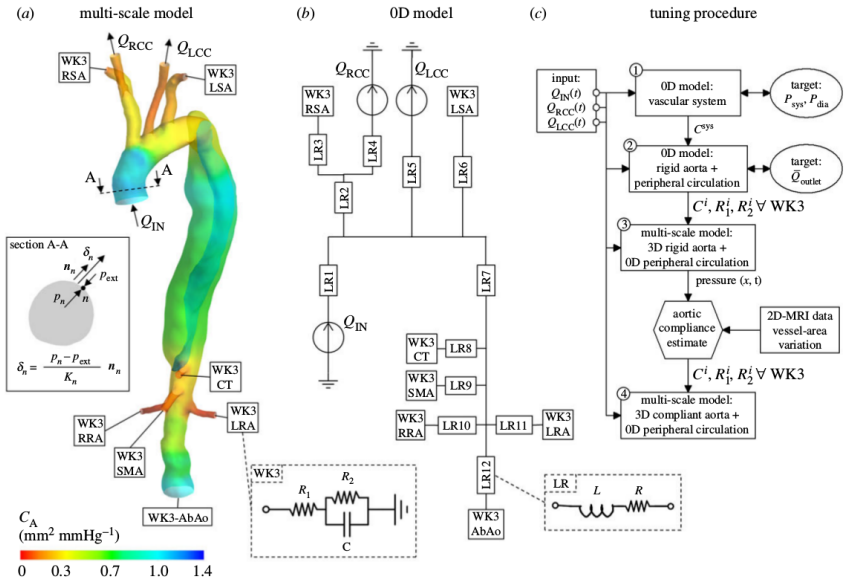
Clearly, imposing only a viscous stress-free condition on the velocity field at the outflow of the domain, one gets simply a homogeneous Neumann-like condition for the velocity vector field. The second alternative advised by those authors is a new, straightforward, *loss coefficient model* boundary condition, which basically means that one aims to find a coherent pressure value to be prescribed on the outflow section, and which could also take into account all the load losses due to the downstream system linked to each exit section, through a simple and easy-implementing model as the following one:

$$p_i - p_b = \xi_i \frac{1}{2}\rho\boldsymbol{V_i}^2$$

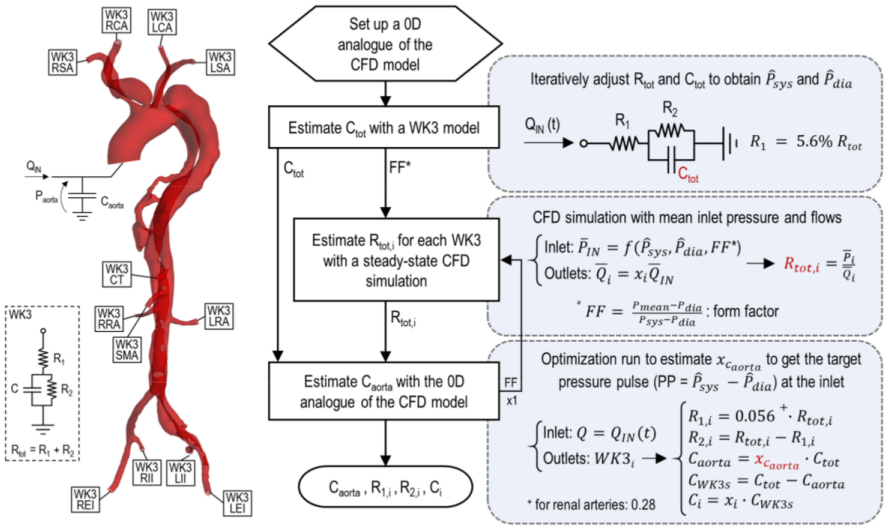
Where the subscript i refers to the i-th outlet, whereas ξ is the loss coefficient corresponding to each of them, and which needs to be estimated, and finally the pressure p_b is meant as a reference value corresponding to a downstream pressure, for example linked to some organs or typical venous pressures.

As last solution among those introduced here, we have left the most suitable one. By now, it is more and more common for researchers who aim to perform numerical simulations concerning blood flows and Hemodynamics to make use of multiscale models in order to provide appropriate boundary conditions to their models, which can in turn ensure consistency and faithfulness to them. The most spread of these are the well known Windkessel models, that are basically electric circuits composed of different circuital elements, thus non-dimensional models, linked to the dimensional domain through its boundaries and to which it provides the proper conditions depending on both what exits from the domain itself and how that 0-dimensional model is calibrated. These multidimensional schemes could become as more detailed and complicated as one wants them to be, in order to mimic the effect and the behaviour of the not represented downstream part of the cardiovascular system on the reconstructed aortic computational domain. In other words, a realistic pressure waveform is obtained by the 0-dimensional model, starting from a flow rate signal which comes from the multidimensional domain

(or vice versa), and it is prescribed as the new boundary condition at each time step. This aspect moves from a fluid dynamics-electrical analogy. For a clear and exhaustive explanation of the Windekessel effect and modeling, see [20], among the others cited, but we point out that this thesis work will focus deeply on this subject in the following chapters. This is therefore currently the state of the art of Computational Hemodynamics,



(a) Reference in [17].



(b) Reference in [18].

Figure 1.14: 3D aortic dissection domain with Windessel-like boundary conditions, and tuning procedure.

the only drawback of such multiscale models is the necessity for them to be properly tuned and calibrated, and thus the requirement for some specific procedures to be devised. Finally, some examples are shown in figure 1.14.

Chapter 2

Mathematical Models

In the context of this chapter, all the mathematical models exploited to build our numerical simulations will be introduced. Starting from the derivation of the fundamental governing equations of Fluid Mechanics, in their incompressible formulation, a brief survey of hyperelatic structural models will be addressed, as they will be those used to represent the structural behaviour of the aortic biological tissues, and finally a more complete and deep analysis of multiscale models, with special regard to Windkessel's 0-dimensional ones, will be also presented.

2.1 Classification of PDEs

In this first section a quick review of *Partial Differential Equations* (PDEs) classification is intended to be given, with the aim of better understanding the numerical methods which best suit each case of interest, depending on the nature of those equations themselves. All what it follows could be found in any Differential Analysis text, we address to A. Quarteroni's one in [66] to provide a very reliable example.

2.1.1 Definitions and Examples

With the term *partial differential equation* one intends to refer to a differential equation containing derivatives of the unknown variable, such as the velocity component, (or variables, in case of systems of equations) with respect to several variables, such as time and space, in contrast with the so-called *ordinary differential equations*, which considered only the derivative of the unknown variable with respect to one variable, such as time or space. A PDE is an object which could look like:

$$F\left(\boldsymbol{x}, t, \phi, \frac{\partial \phi}{\partial t}, \frac{\partial \phi}{\partial x_1}, \ldots, \frac{\partial \phi}{\partial x_n}, \frac{\partial^2 \phi}{\partial x_1 \partial x_2}, \ldots, g\right) = 0 \tag{2.1}$$

Usually one classifies those equations depending on their order of derivation, on their linearity properties, and on the nature of the equations themselves, which could be listed as:

- elliptic equations;

- parabolic equations;
- hyperbolic equations.

Some examples of PDEs frequently appearing in Physics and Applied Mathematics are now provided, starting from the popular *transport equation*, or *advection equation*:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (c\phi) = 0 \tag{2.2}$$

The *potential equation*, or *Poisson's equation*:

$$-\nabla^2 \phi = g \tag{2.3}$$

The *heat equation*, or *diffusion equation*:

$$\frac{\partial \phi}{\partial t} + k\nabla^2 \phi = 0 \tag{2.4}$$

And, finally, the *wave equation*:

$$\frac{\partial^2 \phi}{\partial t^2} + c\nabla^2 \phi = 0 \tag{2.5}$$

A first way to classify those equations often advised by many texts is to compare them with a standard second order PDE of the form:

$$L\phi = A\frac{\partial^2 \phi}{\partial x_1^2} + B\frac{\partial^2 \phi}{\partial x_1 \partial x_2} + C\frac{\partial^2 \phi}{\partial x_2^2} + D\frac{\partial \phi}{\partial x_1} + E\frac{\partial \phi}{\partial x_2} + F\phi = 0 \tag{2.6}$$

Now, depending on the quantity indicated with $\Delta = B^2 - 4AC$, one can distinguish three different cases:

- if $\Delta = B^2 - 4AC < 0$ the equation is said to be elliptic;
- if $\Delta = B^2 - 4AC = 0$ the equation is said to be parabolic;
- if $\Delta = B^2 - 4AC > 0$ the equation is said to be hyperbolic.

From this last classification and with regard to the standard PDE 2.6, it is pretty much straightforward now to establish the nature of the equations presented above, and for this purpose one can observe that equations 2.2 and 2.5 are both hyperbolic, equation 2.4 is parabolic, while equation 2.3 is elliptic.

Following the standard procedure reported in [66], it is now possible to look for a change of variables for equation 2.6 which allows us to write it in the new form:

$$L'\phi = -4A\Delta\frac{\partial^2 \phi}{\partial \xi \partial \eta} = 0 \tag{2.7}$$

This notation is actually permitted only if we deal with the hyperbolic case, that is if $\Delta > 0$ as we said before, and it is easy to verify that this last equation 2.7 admits a solution of the type:

$$\phi = p(\xi) + q(\eta) \tag{2.8}$$

which corresponds to the case of having two lines, respectively the line $\xi = const.$ and the one $\eta = const.$, along which the functions p and q remain constant too. The latter are therefore called *characteristic lines* of 2.7, ore more shortly *characteristics*. In case we had instead $\Delta = 0$, that is the parabolic case, one could find another form of 2.7, for which the solution would be of the type:

$$\phi = p(\xi) + \eta q(\xi) \tag{2.9}$$

Which would mean that only one family of characteristics exists for this second case, the one where $\xi = const.$, and no others. Finally, the last case with $\Delta < 0$, that is the elliptic case, the form of 2.7 which could be reached would not admit any family of characteristic lines this time. It is exactly from these last considerations that the well known *Characteristics Method* comes, which could therefore be used in order to solve exactly parabolic and hyperbolic problems marching in time along their characteristic lines.

2.1.2 Numerical Solution

Once illustrated the wide world of PDEs, one needs to admit the fact that, apart from some very particular and rare cases where incredibly strong simplifications and assumptions can be made, it is almost impossible to obtain their analytical and explicit solution in what is commonly named as closed form. Oftentimes one has to bind the analytical investigation to the proof of the existence, the uniqueness and the possible degree of regularity of those solutions, without explicitly determining them. It is exactly for this reason that numerical methods and tools have been devised and invented, in order to find a numerical approximation of the solutions to those equations for which it would be almost impossible to find a closed and analytical one. Beside the necessity of determining a faithful numerical solution, hereafter indicated with ϕ_N, one needs also to establish a way to measure the error between the latter and the exact solution ϕ, in order to evaluate the numerical properties of the method, such as what we will call its *consistency*, *stability*, *convergence* and, of course, *accuracy*.

Thus indicating with ϕ_N the numerical approximation of the exact solution ϕ to the problem $\mathcal{P}$, whose approximation will be denoted as $\mathcal{P}_N$, and which will depend on the set of data x, approximated as x_N, one obtains that the differential problem to be solved would be turned into a numerical one of the type:

$$\mathcal{P}(\phi, x) = 0 \qquad \rightarrow \qquad \mathcal{P}_N(\phi_N, x_N) = 0$$

As it regards the properties listed above, they refer to the methods and the algorithms used in order to approximate and solve such problems. A simple definition of each of them is now here briefly reported:

- **consistency:** it refers to the numerical method adopted to approximate the differential problem, thus, the former is said to be consistent if, as the *discretization* of the latter tends to dispose of up to an infinite number of grid points N, the discretized problem itself tends to resemble the continuous original one, that is:

$$\lim_{N \to \infty} \Big(\mathcal{P}_N(\phi_N, x_N) - \mathcal{P}(\phi, x) \Big) = 0$$

- **stability:** it is a property of the algorithm used to solve the numerical problem, and we say that we have stability when small perturbations on the data do not affect too much the solution, so that also the latter will be little perturbed (this is the case, for instance, of round-off errors, etc.), thus given:

$$\forall \varepsilon > 0,\ \exists \delta(\varepsilon) > 0 : \forall \delta_{x_N} : ||\delta_{x_N}|| \leq \delta \implies \delta_{\phi_N} \leq \varepsilon,\ \forall N \geq 1$$

  therefore we still have:

$$\mathcal{P}_N(\phi_N + \delta_{\phi_N}, x_N + \delta_{x_N}) = 0$$

- **convergence:** in order for a given method to be said convergent, it has to provide a numerical solution ϕ_N which tends to the exact solution of the continuous problem ϕ as the number of discretisation points tends to infinity, that is as the numerical model tends to resemble the continuous original one, for a consistent method. This means that:

$$\lim_{N \to \infty} ||\phi - \phi_N|| = 0$$

To conclude this section, we recall the fundamental theorem which links together all these properties ensuring the convergence of a method whether its stability and consistency can be proven:

Lax-Richtmyer equivalence theorem. *If a method is consistent, then it is convergent if and only if it is stable.*

2.2 Derivation of the Navier-Stokes Equations

The present section is dedicated to the introduction of the main mathematical model which any Computational Fluid Dynamics simulation is based on, i.e. the governing equations of Fluid Mechanics. There are many different ways of deriving and writing these equations, as well as the name they are given: equations of motion, equations of Fluid Mechanics, or, as we will always refer to them, *Navier-Stokes equations*. In the literature one can find a very wide collection of works on this omni-comprehensive subject, here are reported, indicatively, those in [46, p. 7-24] and, above all, [65] for a complete description of such equations and the challenging approaches developed in order to face them in their incompressible formulation. As it regards their derivation here showed, we will refer to the slides in [28], presented by professor Domenic D'Ambrosio, Polytechnic of Turin, as part of Computational Fluid Dynamics classes. This section is therefore organised as follows: the three principles of conservation will be introduced from a heuristic point of view, in their integral form, then each of their differential forms will be derived.

2.2.1 Conservation of Mass

Starting from the balance equation of mass, or *continuity equation*, as often it is referred to, before starting writing down its straightforward integral form, which is nothing but a mere mass conservation law, we had better introduce some notations which will turn out

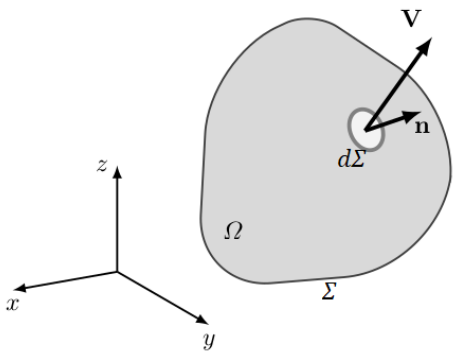


Figure 2.1: Control volume Ω with its external control surface $\partial\Omega = \Sigma$ in a cartesian reference frame.

to be useful also for the next subsections. Denoting with t the time and with $\boldsymbol{x} = (x, y, z)^t$ the space coordinates, with ρ the density, i.e. the mass per unit volume, and with $\boldsymbol{V} = (u, v, w)^t$ the velocity field, moreover indicating with $\Omega \subset \mathbb{R}^3$ a control volume which is fixed in time and does not move with the fluid, but instead is crossed by the latter which flows through its external surface denoted by $\partial\Omega = \Sigma$, see figure 2.1, then we have that if:

$$\int_\Omega \rho(t, \boldsymbol{x})\, d\Omega$$

The object above could be referred as the mass of fluid contained within the control volume Ω at time t, taking its time derivative we can get the mass-rate related to that volume, which must be equal to the difference between the total amount of fluid entering the volume and its amount leaving the volume, that are (indicating with $\boldsymbol{n}$ the normal unit vector externally oriented with respect to the volume's surface):

$$\int_\Sigma \rho(t, \boldsymbol{x}) \boldsymbol{V}(t, \boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x})\, d\Sigma$$

Thus, the equation of balance of mass is now pretty easy to write as:

$$\frac{\mathrm{d}}{\mathrm{d}t} \int_\Omega \rho(t, \boldsymbol{x})\, d\Omega = -\int_\Sigma \rho(t, \boldsymbol{x}) \boldsymbol{V}(t, \boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x})\, d\Sigma \tag{2.10}$$

The minus sign in front of the integral is due to the assumption of considering as positive a net mass flow exiting from the volume (which would imply a reduction of the mass contained into it in time), and as negative a net mass flow entering the volume. Equation 2.10 is normally referred to as the integral form of the continuity equation in Fluid Mechanics, written in its conservative form (where the conservative variable ρ appears together with its flux $\rho\boldsymbol{V}$). By exploiting the *divergence theorem* (or *Gauss'*), according to which, given a vector field $\boldsymbol{W}$ and closed volume D, one has that:

$$\int_{\partial D} \boldsymbol{W} \cdot \boldsymbol{n}\, d\partial D = \int_D \nabla \cdot \boldsymbol{W}\, dD$$

It could be worth recalling at this point the definition of the *nabla* operator, that is:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right)$$

Applying now such theorem to 2.10, and the grouping all terms into the same volume integral, one obtains the so-called differential form of the continuity equation, still in its conservative version:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho(t, \boldsymbol{x})\boldsymbol{V}(t, \boldsymbol{x})) = 0 \tag{2.11}$$

Finally, by expanding the divergence operator which appears in this last expression 2.2.2, one gets:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho(t, \boldsymbol{x})u(t, \boldsymbol{x}))}{\partial x} + \frac{\partial(\rho(t, \boldsymbol{x})v(t, \boldsymbol{x}))}{\partial y} + \frac{\partial(\rho(t, \boldsymbol{x})w(t, \boldsymbol{x}))}{\partial z} = 0 \tag{2.12}$$

2.2.2 Conservation of Momentum

The second fundamental equation that we will introduce is the momentum balance equation, which is, in other words, nothing but a generalised way of writing the second Newton's Dynamics law, which reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}(m\boldsymbol{V}) = \boldsymbol{f}$$

In fact, again as left-hand side we have the time derivative of the linear momentum, that is the quantity $\rho\boldsymbol{V}$ per unit volume, thus:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \rho\boldsymbol{V}\, d\Omega$$

Omitting the undermeant dependence on both time and space of the field variables, for the sake of brevity. On the right-hand side actually we should write all forces acting on the control volume Ω, which could be in general volume forces or surface ones, plus the net momentum flow crossing the control surface $\partial\Omega = \Sigma$. Starting right from the latter, we will deal with a vectorial equation, whose components are three balance equation respectively for each cartesian component of the velocity field $\boldsymbol{V}$, in a cartesian frame. Thus the net flux for each of these components will read:

$$\int_{\Sigma} \rho u\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma \quad , \quad \int_{\Sigma} \rho v\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma \quad \text{and} \quad \int_{\Sigma} \rho w\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma$$

Concerning volume forces, they will not be taken into consideration for this specific work, since vector fields such as electro-magnetic ones, or the gravitational field are absolutely negligible in the considered scenario. Much more important is, on the other hand, the model adopted to write the surface forcing terms, that are summarized in a so-called second order *stress tensor* $\boldsymbol{\sigma}$, which in turn can be decomposed into an isotropic component and a non-isotropic one. Thus we have an object which looks like:

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{xz} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{yz} \\ \sigma_{xz} & \sigma_{zy} & \sigma_{zz} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\sigma}_x \\ \boldsymbol{\sigma}_y \\ \boldsymbol{\sigma}_z \end{pmatrix}$$

Where all the subscripts refer to the normal of the face and to the coordinate direction along which the stress tensor's component acts, with respect to the elementary element of fluid in a cartesian frame, as shown in figure 2.2. It is now pretty clear the way those

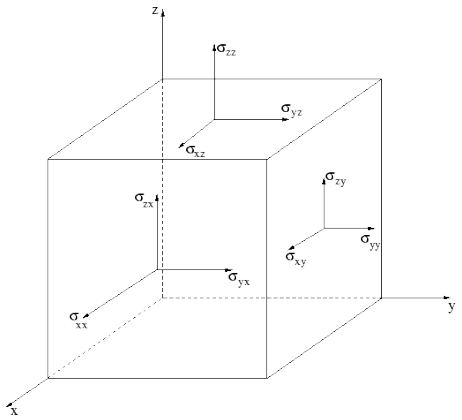


Figure 2.2: Cartesian stress tensor's components on an elementary element of fluid.

subscripts work, that is, considering the generic component of the stress tensor σ_{ij}, then the first index i refers to the normal direction of the face on which that component acts, while the second one, j, refers instead the actual direction of the stress component on that face. As anticipated, it is possible to split this object in two different tensors whose contribute is physically easier to understand, that are a static pressure and a viscous stress ones. Therefore, one can write a relationship of the type:

$$\boldsymbol{\sigma} = -p\boldsymbol{I} + \boldsymbol{\tau} \tag{2.13}$$

Where:

$$p\boldsymbol{I} = \begin{pmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\tau} = \begin{pmatrix} \tau_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \tau_{yy} & \tau_{yz} \\ \tau_{xz} & \tau_{zy} & \tau_{zz} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau}_x \\ \boldsymbol{\tau}_y \\ \boldsymbol{\tau}_z \end{pmatrix}$$

The decomposition 2.13 can also be written according to the index notation as:

$$\sigma_{ij} = -p\delta_{ij} + \tau_{ij}$$

Where the symbol δ_{ij} refers right to the so-called Kronecker's delta function ($\delta_{ij} = 1$ if $i = j$, otherwise $\delta_{ij} = 0$). To determine those new terms that have been introduced in such way, one can consider the following definitions, respectively, for the static pressure and the viscous stress components:

$$p = -\frac{1}{3}\sum_{i=1}^{3}\sigma_{ii} \tag{2.14}$$

and:

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \delta_{ij}\lambda\sum_{k=1}^{3}\frac{\partial u_k}{\partial x_k} \tag{2.15}$$

Where the coefficient λ can be assumed as equal to $-\frac{2}{3}\mu$ under the so-called *Stokes' hypothesis*, and is therefore named *second viscosity coefficient*. In addition, by combining 2.14 and 2.15 with the definition of the stress tensor $\boldsymbol{\sigma}$, one can show the consequences of this last hypothesis, which consist of the definition given to the static pressure itself.

Subsequently, by writing the contribution of these surface forcing terms integrated over the whole surface Σ of the control volume, one obtains the following relations:

$$\int_\Sigma \boldsymbol{\sigma} \cdot \boldsymbol{n}\, d\Sigma = -\int_\Sigma p\boldsymbol{I} \cdot \boldsymbol{n}\, d\Sigma + \int_\Sigma \boldsymbol{\tau} \cdot \boldsymbol{n}\, d\Sigma$$

And, considering the contribute along each direction on a cartesian frame:

$$\int_\Sigma \boldsymbol{\sigma}_x \cdot \boldsymbol{n}\, d\Sigma = -\int_\Sigma p n_x\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_x \cdot \boldsymbol{n}\, d\Sigma$$
$$\int_\Sigma \boldsymbol{\sigma}_y \cdot \boldsymbol{n}\, d\Sigma = -\int_\Sigma p n_y\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_y \cdot \boldsymbol{n}\, d\Sigma$$
$$\int_\Sigma \boldsymbol{\sigma}_z \cdot \boldsymbol{n}\, d\Sigma = -\int_\Sigma p n_z\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_z \cdot \boldsymbol{n}\, d\Sigma$$

Combining all these terms with those previously written for the time derivative of the linear momentum and for its flow through the surface of the control volume, one can finally obtain the integral form of the momentum balance equation, which reads:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \rho\boldsymbol{V}\, d\Omega = -\int_\Sigma (\rho\boldsymbol{V})\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma - \int_\Sigma p\boldsymbol{I} \cdot \boldsymbol{n}\, d\Sigma + \int_\Sigma \boldsymbol{\tau} \cdot \boldsymbol{n}\, d\Sigma \tag{2.16}$$

Or, with respect to each cartesian direction:

$$\begin{cases} \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int_\Omega \rho u\, d\Omega = -\int_\Sigma \rho u\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma - \int_\Sigma p n_x\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_x \cdot \boldsymbol{n}\, d\Sigma \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int_\Omega \rho v\, d\Omega = -\int_\Sigma \rho v\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma - \int_\Sigma p n_y\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_y \cdot \boldsymbol{n}\, d\Sigma \\ \dfrac{\mathrm{d}}{\mathrm{d}t}\displaystyle\int_\Omega \rho w\, d\Omega = -\int_\Sigma \rho w\boldsymbol{V} \cdot \boldsymbol{n}\, d\Sigma - \int_\Sigma p n_z\, d\Sigma + \int_\Sigma \boldsymbol{\tau}_z \cdot \boldsymbol{n}\, d\Sigma \end{cases} \tag{2.17}$$

As we did for the mass balance equation, we now aim to find the differential form of 2.16 (or equivalently of each case of 2.18), again by applying the divergence theorem on all the surface integrals and gathering all the resulting terms under the same integral operator, then by excluding the integration over the domain Ω of the control volume, one has:

$$\frac{\partial(\rho\boldsymbol{V})}{\partial t} = -\nabla \cdot ((\rho\boldsymbol{V})\boldsymbol{V}) - \nabla p + \nabla \cdot \boldsymbol{\tau} \tag{2.18}$$

And with respect to each cartesian component:

$$\begin{cases} \dfrac{\partial(\rho u)}{\partial t} = -\nabla \cdot (\rho u\boldsymbol{V}) - \dfrac{\partial p}{\partial x} + \nabla \cdot \boldsymbol{\tau}_x \\ \dfrac{\partial(\rho v)}{\partial t} = -\nabla \cdot (\rho v\boldsymbol{V}) - \dfrac{\partial p}{\partial y} + \nabla \cdot \boldsymbol{\tau}_y \\ \dfrac{\partial(\rho w)}{\partial t} = -\nabla \cdot (\rho w\boldsymbol{V}) - \dfrac{\partial p}{\partial z} + \nabla \cdot \boldsymbol{\tau}_z \end{cases} \tag{2.19}$$

Finally, by developing all the divergence operators which appear in 2.19, one comes to the following last differential (and conservative) form of the momentum balance equation, split into its cartesian components:

$$\begin{cases} \dfrac{\partial(\rho u)}{\partial t} + \dfrac{\partial(\rho uu)}{\partial x} + \dfrac{\partial(\rho uv)}{\partial y} + \dfrac{\partial(\rho uw)}{\partial z} = -\dfrac{\partial p}{\partial x} + \dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{yx}}{\partial y} + \dfrac{\partial \tau_{zx}}{\partial z} \\ \dfrac{\partial(\rho v)}{\partial t} + \dfrac{\partial(\rho vu)}{\partial x} + \dfrac{\partial(\rho vv)}{\partial y} + \dfrac{\partial(\rho vw)}{\partial z} = -\dfrac{\partial p}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \tau_{yy}}{\partial y} + \dfrac{\partial \tau_{zy}}{\partial z} \\ \dfrac{\partial(\rho w)}{\partial t} + \dfrac{\partial(\rho wu)}{\partial x} + \dfrac{\partial(\rho wv)}{\partial y} + \dfrac{\partial(\rho ww)}{\partial z} = -\dfrac{\partial p}{\partial z} + \dfrac{\partial \tau_{xz}}{\partial x} + \dfrac{\partial \tau_{yz}}{\partial y} + \dfrac{\partial \tau_{zz}}{\partial z} \end{cases} \tag{2.20}$$

This time the conservative variables appearing in 2.18, 2.19 and 2.20 are the three components of the linear momentum per unit volume, that are ρu, ρv and ρw, together with their fluxes crossing the surface of the control volume.

2.2.3 Conservation of Energy

As last, we derive also the total energy balance equation, even though it will not be useful with respect to the current work, since it can be proven that the latter and the previous two balance equations are actually decoupled whenever one deals with incompressible cases. However, for completeness reasons, it is worth introducing also this last relationship in order to complete the presentation of the Navier-Stokes system of equations. First, it is necessary to specify which kind of energy we are concerned with: by indicating with e the so-called *internal energy* associated with the fluid under consideration, per unit mass, which could be approximated with the classical thermodynamic relations involving the temperature of the fluid and its specific heat at constant volume and per unit mass, thus we denote with:

$$\rho e + \frac{1}{2}\rho \boldsymbol{V}^2 = \rho\left(c_V T + \frac{1}{2}\boldsymbol{V}^2\right) = \rho E \tag{2.21}$$

In the latter we have defined the *total energy* of the fluid, indicated with E as it is usually done, by summing its internal energy with the kinetic one, per unit mass. Right this last introduced quantity, the total energy per unit volume ρE, is supposed to be the conservative variable which will appear in the energy balance equation derived here below, again together with its flux. As we did in the previous paragraphs in order to write down the mass and momentum balance equation, let us start with the time derivative of the total energy contained in the whole domain, i.e. the control volume Ω, which can be written as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \rho E \, d\Omega$$

Then, we write its flux through the control surface $\Sigma = \partial\Omega$ of the control volume, following the same structure used to write those relating to the mass and to the momentum flows, that is:

$$\int_\Sigma \rho E \boldsymbol{V} \cdot \boldsymbol{n} \, d\Sigma$$

As right-hand side of this last balance equation, we have to include all those terms which contribute someway to the production (or the consumption) of energy within the control

volume, including therefore all the kinds of sources whether they are internal (volume sources) or superficial. To begin with, we should consider the work produced by all those forcing terms previously introduced in the context of the momentum balance equation, thus referring to its integral form expressed by relation 2.16, one has to take into account the work due to the static pressure terms, that exerted by the viscous stress terms, and, if it was the case, also the work linked to the eventual volume forces contained in the domain Ω. Secondly, it is necessary to include the contribute of internal heating sources, if present, and also the conductive flow through the control surface Σ of the volume, associated to the temperature gradient. To this end, we start from the work-sources terms, and we thus write their contribute by considering the integrated power per unit volume produced by those forcing terms cited above, that are:

$$-\int_{\Sigma} p\boldsymbol{V}\cdot\boldsymbol{n}\,d\Sigma$$

$$\int_{\Sigma}(\boldsymbol{\tau}\cdot\boldsymbol{V})\cdot\boldsymbol{n}\,d\Sigma=\int_{\Sigma}(u\boldsymbol{\tau}_x+v\boldsymbol{\tau}_y+w\boldsymbol{\tau}_z)\cdot\boldsymbol{n}\,d\Sigma$$

As it regards the internal heating source terms and the conductive fluxes crossing the control surface, they can be easily written as follows, respectively:

$$\int_{\Omega}\rho Q_{\Omega}\,d\Omega$$

And:

$$-\int_{\Sigma}\boldsymbol{q}\cdot\boldsymbol{n}\,d\Sigma$$

Where the conductive flux $\boldsymbol{q}=(q_x,q_y,q_z)$ can be expressed referring to the well known *Fourier's law*, which states that $\boldsymbol{q}=-k\nabla T=-k(\frac{\partial T}{\partial x},\frac{\partial T}{\partial y},\frac{\partial T}{\partial z})$, and the minus sign is due again to the fact that when the conductive heating exits from the domain, that is when it is directed according to the convention adopted for the normal unit vector associated to the control surface, then its temperature T must decrease, and thus the time derivative of the internal energy (i.e. of the total energy) in turn has to be negative.

Gathering all those terms together, one finally obtains the integral and conservative form of the total energy balance equation, which reads:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho E\,d\Omega=&-\int_{\Sigma}\rho E\boldsymbol{V}\cdot\boldsymbol{n}\,d\Sigma-\int_{\Sigma}p\boldsymbol{V}\cdot\boldsymbol{n}\,d\Sigma+\int_{\Sigma}(\boldsymbol{\tau}\cdot\boldsymbol{V})\cdot\boldsymbol{n}\,d\Sigma+\\&+\int_{\Omega}\rho Q_{\Omega}\,d\Omega-\int_{\Sigma}\boldsymbol{q}\cdot\boldsymbol{n}\,d\Sigma\end{aligned}\tag{2.22}$$

Once again, in order to derive the corresponding differential form one applies the divergence theorem to all the surface integrals in 2.22, then gathering all the contributes within the same volume integral and thus reaching the final desired expression:

$$\frac{\partial(\rho E)}{\partial t}+\nabla\cdot(\rho E\boldsymbol{V})=-\nabla\cdot(p\boldsymbol{V})+\nabla\cdot(\boldsymbol{\tau}\cdot\boldsymbol{V})+\rho Q_{\Omega}-\nabla\cdot\boldsymbol{q}\tag{2.23}$$

The subsequent expanded form is obviously straightforward to obtain, and therefore here omitted for reasons of brevity.

2.3 Incompressible Navier-Stokes Equations

This section could have been placed together with the previous one as a sub-case of Navier-Stokes equations, instead it has been given the dignity of an autonomous section owing to its relevance to the current work, since it actually represents once and for all right the model which will be employed to carry out the hemodynamic numerical simulations on the subject of aortic dissection we are interested to reproduce. Those derived and presented in the last section 2.2 were the Navier-Stokes equation in their most general dress, written in the so-called conservative form, but we now aim to write down a more specialised expression for each of them,by adopting the incompressible hypothesis for the fluid which has to be modeled by the equations themselves. As already mentioned, we are no longer interested in the equation regarding the energy balance, the third one presented in section 2.2, thus from now on we will always refer to Navier-Stokes system of equations as it was composed only by the continuity and the Newton's law equations. Moreover, since the problem we will deal with is basically a 2-dimension case of study representing a hemodynamic phenomenon (i.e. a type B aortic dissection, for more details on the physiological background, see section 1.1) , we will disregard in the following the third cartesian component of the velocity vector field, that is we will deal with the problem of determining a solution in terms of $\boldsymbol{V}(\boldsymbol{x},t) = \big(u(\boldsymbol{x},t), v(\boldsymbol{x},t)\big)$ only.

2.3.1 The Continuity and Momentum Equations

Concerning the incompressibility assumption, it is very well-fitting in case of liquids, such as in our specific problem regarding blood, whereas it could be applied to gasses only when we deal with relatively low speed in comparison with the speed of sound, that is when the ratio between the flow speed and that of sound, i.e. the *Mach number* of the considered stream, is much less than 1, to say, up to $M = \frac{V}{c} \approxeq 0.3$, where c indicates the speed of sound in that specific condition, that is $c = \sqrt{\gamma \frac{p}{\rho}}$. When we assume incompressibility, then it means that:

$$\rho = const.$$

The last holds both with respect to time and to space, therefore any derivative (temporal or spacial) of the density will be null due to this condition, thus:

$$\frac{\partial \rho}{\partial t} = 0 \quad \text{and} \quad \frac{\partial \rho}{\partial x_i} = 0$$

That implies that several important considerations now can be done on the governing equations, and moving from their forms 2.12 and 2.20, first we can observe the fact that the former, the continuity equation, simplifies as:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \tag{2.24}$$

While as for the momentum equation, the second recalled one, let us start formulating our considerations from the following simplified expressions:

$$\begin{cases} \rho\left(\dfrac{\partial u}{\partial t} + \dfrac{\partial (uu)}{\partial x} + \dfrac{\partial (uv)}{\partial y}\right) = -\dfrac{\partial p}{\partial x} + \dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{yx}}{\partial y} \\ \rho\left(\dfrac{\partial v}{\partial t} + \dfrac{\partial (vu)}{\partial x} + \dfrac{\partial (vv)}{\partial y}\right) = -\dfrac{\partial p}{\partial y} + \dfrac{\partial \tau_{xy}}{\partial x} + \dfrac{\partial \tau_{yy}}{\partial y} \end{cases} \tag{2.25}$$

Coming back to the definition of the viscous stress terms, one can notice that it also simplifies as:

$$\tau_{xx} = \mu\left(\frac{\partial u}{\partial x} + \frac{\partial u}{\partial x}\right) = 2\mu\frac{\partial u}{\partial x}$$

$$\tau_{yy} = \mu\left(\frac{\partial v}{\partial y} + \frac{\partial v}{\partial y}\right) = 2\mu\frac{\partial v}{\partial y}$$

$$\tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) = \tau_{yx}$$

Thus inserting the latter in 2.25, and considering that:

$$\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} = \mu\left(2\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x \partial y}\right) = \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial}{\partial x}\left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\right)\right)$$

Which, by applying the continuity equation 2.24, becomes:

$$\frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} = \mu\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \mu\nabla^2 u$$

And, equivalently:

$$\frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} = \mu\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) = \mu\nabla^2 v$$

Moreover, the convective terms which appear in relations 2.25 could still be simplified again thanks to 2.24 as:

$$\frac{\partial (uu)}{\partial x} + \frac{\partial (uv)}{\partial y} = 2u\frac{\partial u}{\partial x} + u\frac{\partial v}{\partial y} + v\frac{\partial u}{\partial y} = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}$$

In the same way:

$$\frac{\partial (vu)}{\partial x} + \frac{\partial (vv)}{\partial y} = u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}$$

Thus, finally, the momentum balance equation under the assumption of incompressibility now becomes:

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} = -\dfrac{\partial p}{\partial x} + \nu\left(\dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial y^2}\right) \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} = -\dfrac{\partial p}{\partial y} + \nu\left(\dfrac{\partial^2 v}{\partial x^2} + \dfrac{\partial^2 v}{\partial y^2}\right) \end{cases} \tag{2.26}$$

Where the so-called *kinematic viscosity* $\nu = \frac{\mu}{\rho}$ has been introduced. One more step can be done with the purpose of writing a compact form of such equations 2.24 and 2.26,

that is a vectorial notation as they had been initially introduced, somehow heuristically, in the previous section 2.2. Grouping together the cartesian components of the velocity field $\boldsymbol{V} = (u, v)$ and the partial derivatives with respect to space in the *nabla* operator $\nabla = (\frac{\partial}{\partial x}, \frac{\partial}{\partial y})$, one comes to the couple of equations here below reported, which represent the so-called *incompressible 2D Navier-Stokes equations*.

$$\begin{cases} \nabla \cdot \boldsymbol{V} = 0 \\ \dfrac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \nu \nabla^2 \boldsymbol{V} \end{cases} \tag{2.27}$$

Where the non-linear term $\boldsymbol{V} \cdot \nabla \boldsymbol{V}$, also known as the *convection* or *advection* contribution, can be thought as:

$$\boldsymbol{V} \cdot \nabla \boldsymbol{V} = \begin{pmatrix} u & v \end{pmatrix} \cdot \begin{pmatrix} \dfrac{\partial u}{\partial x} & \dfrac{\partial v}{\partial x} \\ \dfrac{\partial u}{\partial y} & \dfrac{\partial v}{\partial y} \end{pmatrix}$$

While the viscous contribution expressed by $\frac{\mu}{\rho} \nabla^2 \boldsymbol{V}$ is composed of the so-called *diffusion* terms, responsible in particular for the arising of boundary layers in the wall's proximity, where gradients of the velocity field appear.

2.3.2 Boundary and Initial Conditions for Incompressible Problems

In the following we will refer in particular to the first chapter of the book by Quartapelle, reported in [65, p. 1-6], in order to attempt to provide a solid and complete introduction to what will be expressed as *boundary conditions* for incompressible cases, such as the one we are interested in, with the main purpose of enlightening the way for the setting-up of faithful and reliable numerical simulations, with respect to what has been presented in paragraph 2.1.2 about their expected properties. More details will be given later in the dedicated discussions, such as in chapter 3, sections from 3.4 to 3.3.3, where new and alternative will also be proposed beside the one adopted in this work, in order to deal with incompressible numerical problems, but for all of them special care needs to be reserved to the treatment of boundary conditions, since the posedness and the good behaviour of the simulation will depend right from this fundamental feature.

The 2D incompressible Navier-Stokes equations represent a *parabolic* problem (only one family of characteristics can thus be determined), because of their viscous contribute expressed throughout the elliptic diffusive terms, such as it happens in case of the well known *heat equation*. This means also that velocity and pressure have to be in good agreement one with respect to the other, especially as far as boundary conditions are concerned. In fact, as will be shown later, the pressure is no longer an evolutionary variable in such formulation of the governing equations, instead it works as a means by which the incompressibility condition can finally be applied on the velocity field, at any instant of time. When we aim to specify suitable boundary conditions for the problem we are about to face, one could for instance refer to the most spread conditions for the velocity field on the boundaries of the domain (hereafter indicated with D, and whose

frontier $\partial D = S$ is assumed to be the external boundary we are dealing with), which substantially consists of prescribing:

$$\boldsymbol{V}(\boldsymbol{x} \in S, t) = \boldsymbol{V}_S(\boldsymbol{x}_S, t) \tag{2.28}$$

When a condition named as *no-slip* is applied on the boundaries, then $\boldsymbol{V}_S$ is taken equal to the wall's speed, which means that the fluid has to exhibit solidarity with the wall itself in there. Under such conditions, no other assumption is required for the pressure value also present in the momentum equation 2.26, it will result from the computation. Conversely, if another kind of boundary condition was selected for the velocity field, such as in case of inflow or outflow sections, some Dirichlet or Neumann-like condition can also be applied to the pressure field on those regions. With respect to the continuity equation 2.24, by taking into account the condition 2.28, one obtains that, integrating over the whole domain D:

$$\int_D \nabla \cdot \boldsymbol{V}\, dD = \int_S \boldsymbol{V} \cdot \boldsymbol{n}\, dS = \int_S \boldsymbol{V}_S \cdot \boldsymbol{n}\, dS = 0$$

Furthermore, as it regards the initial condition also necessary in order to ensure the well-posedness of the problem, together with the boundary ones, it has to be, as it is usually said, *compatible* with the latter at the time instant $t = 0$. In other words, if we denote the initial condition for the velocity field over the whole domain with:

$$\boldsymbol{V}(\boldsymbol{x}, t = 0) = \boldsymbol{V}_0(\boldsymbol{x}) \tag{2.29}$$

Then, obviously it follows that this initial field has to be solenoidal too, that is, by referring again to the continuity equation 2.24:

$$\nabla \cdot \boldsymbol{V}_0 = 0$$

And finally, because of the compatibility constraint cited above, we have that:

$$\boldsymbol{V}_S(\boldsymbol{x}_S, t = 0) \cdot \boldsymbol{n} = \boldsymbol{V}_0(\boldsymbol{x} \in S) \cdot \boldsymbol{n} \tag{2.30}$$

This compatibility condition must hold in order for an incompressible problem to be well-posed, that is a problem in which the solution $\boldsymbol{V}(\boldsymbol{x}, t)$ is always a divergence-free field, but as it will be shown later in this work, this concerns only the normal component of it (as depicted in relation 2.30), and in addition such constraint is only due to that incompressible characteristic of the problem, with no dependence on its viscous or inviscid behaviour. This feature related to the compatibility condition only associated with the normal components of the solution also means that an incompressible problem can be effectively treated as a parabolic one just after having projected it onto a space of solenoidal vector fields, but we will focus more on this aspect in the next chapter, section 3.3. However, there exist problems for which condition 2.30 is not satisfied, for instance those involving the sudden initial motion of bodies, walls, or their impulsive appearance in the domain. For such situation, a different initial condition is established for the velocity field, given by a potential field which arises because of the incompressibility of the field. This correction is

directly linked to the jump of the prescribed values of the normal components involved in relation 2.30, so that whenever one has that $\boldsymbol{V}_S\big|_{t=0} \cdot \boldsymbol{n} \neq \boldsymbol{V}_0\big|_S \cdot \boldsymbol{n}$, then:

$$\begin{cases} \nabla^2 \Phi_0 = 0 \\ \nabla \Phi_0\big|_S \cdot \boldsymbol{n} = \boldsymbol{V}_S\big|_{t=0} \cdot \boldsymbol{n} - \boldsymbol{V}_0\big|_S \cdot \boldsymbol{n} \end{cases}$$

The latter is nothing but a Neumann problem for Φ_0, whose solution finally allows to determine the new initial velocity condition to apply in order to satisfy the compatibility expressed by 2.30, thus:

$$\boldsymbol{V}_0{}^* = \boldsymbol{V}_0 + \nabla \Phi_0$$

This, in turn, does not ensure the continuity of the tangential component of the velocity field as well, instead, it rather causes a jump of that quantity on the boundary of the domain, thus leading to:

$$\boldsymbol{V}_0{}^*\big|_S \times \boldsymbol{n} \neq \boldsymbol{V}_S\big|_{t=0} \times \boldsymbol{n}$$

2.3.3 The Pressure Poisson Equation

To conclude this section, we resume the key-factors of the discussion by referring to the problem 2.27, provided with proper initial (relation 2.29) and boundary (relation 2.28) conditions, and whose solution can be thus expressed in terms of a velocity vectorial field and a pressure scalar field, defined up to an arbitrary function of time, that is a pair $(\boldsymbol{V}(\boldsymbol{x}, t), p(\boldsymbol{x}, t))$. Right about the latter, some important considerations are worth doing. Whenever one deals with an incompressible problem, it is necessary to point out the fact that the pressure which appears in the momentum balance equations is no more the habitual thermodynamic pressure related to the other variables through the equation of state or different laws, on the contrary, it only plays the role of ensuring the incompressibility constraint to the velocity field, making it a divergence-free vectorial field at any instant of time. In other words, the term ∇p, which is included in equation 2.27, represents the degree of freedom needed by the velocity field in order to evolve in time respecting the incompressibility constraint depicted by the continuity equation, thus as a divergence-free field. This also means that there is no evolutionary equation for the pressure field, in contrast to what happens instead to the velocity field, handled by the momentum governing equation, since it has only to adjust itself continuously in time in order to ensure the incompressibility condition stated by the continuity equation, where the time dependence has been lost with the simplification introduced at the very beginning of this discussion, assuming a constant density for this specific case.

To be more precise and clear, it is possible to derive an equation which shows exactly the role played by the pressure in such incompressible cases, starting from the system represented in 2.27. By applying the divergence operator to the second equation, that is to the momentum equation, one comes to:

$$\frac{\partial}{\partial t}(\nabla \cdot \boldsymbol{V}) + \nabla \cdot (\boldsymbol{V} \cdot \nabla \boldsymbol{V}) + \nabla^2 p - \nu \nabla^2 (\nabla \cdot \boldsymbol{V}) = 0$$

And considering the continuity equations, all the $\nabla \cdot \boldsymbol{V}$ terms simplify, leading to the so-called *Pressure Poisson Equation* (PPE), which reads:

$$\nabla^2 p = -\nabla \cdot (\boldsymbol{V} \cdot \nabla \boldsymbol{V}) \tag{2.31}$$

Equation 2.31 defines an elliptic problem for the pressure field, to be solved at each time step in order to respect the divergence-free condition used right to derive it, and thus representing a non-evolutionary equation governing the pressure field the way described above. There exist some formulations whose aim is to face incompressible problems by devising methods based on this equation, and which we will discuss in the following (section 3.4), but they will not be our case, since we will be concerned with the much more widely spread *Fractional Step Methods*.

2.4 Hyperelastic Structural Models

In the context of the work here presented, i.e. a numerical simulation concerning blood flow within a human vessel, which is basically made up of biological tissues capable of deforming, moving and reacting when the pulsating stream flows through them, formulating a proper model to be associated with those structures, in order to allow for their behaviour's description, is an issue of primary importance. A specific structural model thus needs to be devised, and to this aim we present in this section one of the most popular among those usually addressed in the literature. Such structural models are mainly named *hyperelastic models*, since they can mimic the behaviour of those materials capable of deforming even with large scale displacements, in contrast with those models used to describe infinitesimal and linear strains (e.g. *De saint Venant* elastic theory). See figure 2.3 for some examples of different models commonly adopted. The model adopted in this particular work is the so-called *Mooney-Rivlin* hyperelastic model, whose main features will be presented soon in this section, but other different solutions are also possible, with their *pros* and *cons* each. First, before moving towards the introduction of that specific hyperelastic model, a brief introduction on finite strain or finite deformation theory (or large deformation) is worth exploring here in the following, whose simple description, as far as the current master's thesis work is concerned, is taken from the related Wikipedia's page, link in [3]. For what it follows, refer to figure 2.4. We can distinguish the three main contributes to the motion of a continuum body, here listed:

- rigid body motion: composed of rigid body translation and rotation;
- elastic (or hyperelastic) deformation.

In figure 2.4, with $k(\mathcal{B})$ it has been indicated the configuration of the represented body: $k_0(\mathcal{B})$ denotes the initial configuration, with the body still undeformed (and not moving as well), whereas $k_t(\mathcal{B})$ denotes the same body after the deformation occurred, with a change of its shape. If no change occurs in the shape of a moving body between situations 0 and t, then we deal with a solid-body displacement, conversely, if the first one is the case, then we have effectively an elastic body (the relative distance between each pair of its points changes).

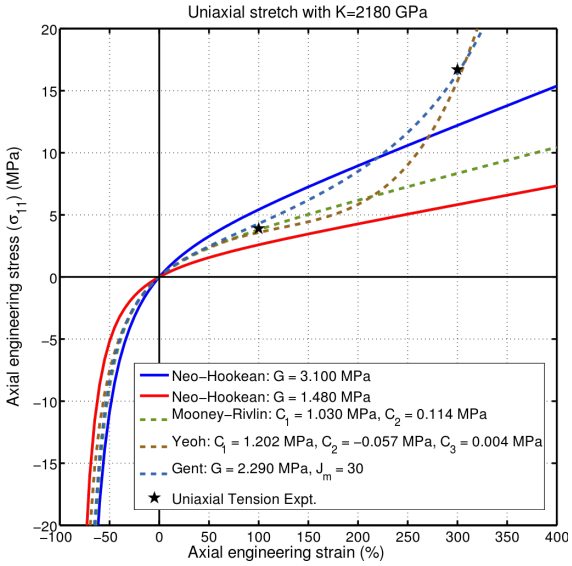


Figure 2.3: Examples of several hyperelastic models used to describe a non-linear stress-strain behaviour of such materials (taken from [2]).

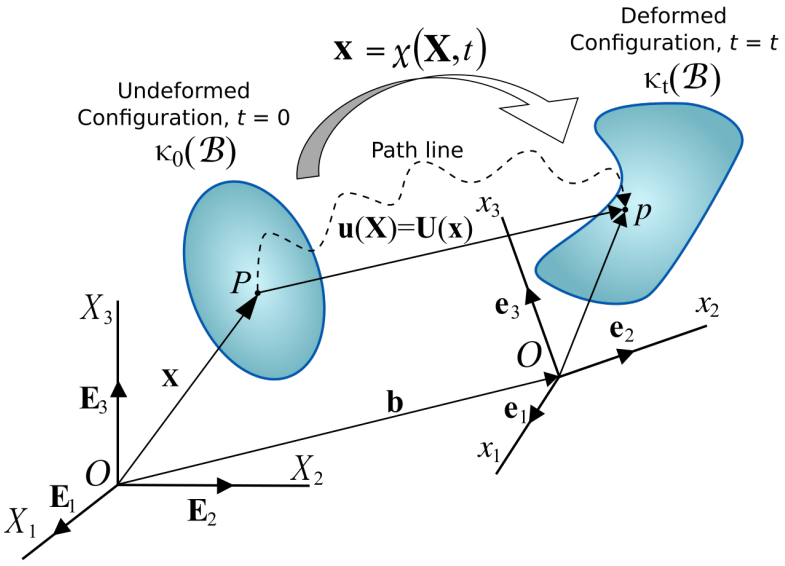


Figure 2.4: Motion of a continuum body, made up of translation, rotation and deformation.

From a lagrangian point of view, we can describe the displacement associated with the point P, belonging to the undeformed configuration $k_0(\mathcal{B})$, responsible for its motion up to the point p, of the deformed configuration $k_t(\mathcal{B})$, by introducing the following displacement vectors:

- $\boldsymbol{X}$: indicating the position of P;
- $\boldsymbol{x}$: indicating the position of p;

- $\boldsymbol{b}$: indicating the translation of the reference frame;
- $\boldsymbol{U}$: indicating the displacement vector (lagrangian view).

Then, it holds that:

$$\boldsymbol{U} = \boldsymbol{b} + \boldsymbol{x}(\boldsymbol{X}, t) - \boldsymbol{X} \tag{2.32}$$

At this point, introducing the partial derivatives of the displacement vector here obtained with respect to the $(\boldsymbol{E}_1, \boldsymbol{E}_2, \boldsymbol{E}_3)$ reference frame, that are $\nabla_{\boldsymbol{X}} = (\frac{\partial}{\partial X_1}, \frac{\partial}{\partial X_2}, \frac{\partial}{\partial X_3})$, one has that:

$$\nabla_{\boldsymbol{X}} \boldsymbol{U} = 0 + \nabla_{\boldsymbol{X}} \boldsymbol{x} - \nabla_{\boldsymbol{X}} \boldsymbol{X} = \boldsymbol{F} - \boldsymbol{I} \tag{2.33}$$

Where there appears the so-called *deformation gradient tensor* $\boldsymbol{F}$, defined as:

$$\boldsymbol{F} = \nabla_{\boldsymbol{X}} \boldsymbol{x}$$

On the other hand, from an eulerian point of view one could obtain a related result considering the fact that this time the position of point P in the undeformed configuration is reconstructed starting from its deformed state p, thus we have that, analogously:

$$\boldsymbol{u} = \boldsymbol{b} + \boldsymbol{x} - \boldsymbol{X}(\boldsymbol{x}, t) \tag{2.34}$$

And so it follows that, deriving this time with respect to the eulerian reference frame $(\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)$, with $\nabla_{\boldsymbol{x}} = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})$, we have:

$$\nabla_{\boldsymbol{x}} \boldsymbol{u} = 0 + \nabla_{\boldsymbol{x}} \boldsymbol{x} - \nabla_{\boldsymbol{x}} \boldsymbol{X} = \boldsymbol{I} - \boldsymbol{F}^{-1} \tag{2.35}$$

Subsequently, we can introduce two additional quantities that are the so-called *right* and *left Cauchy-Green's deformation tensors*, which read, respectively:

$$\boldsymbol{C} = \boldsymbol{F}^t \cdot \boldsymbol{F} = (\nabla_{\boldsymbol{X}} \boldsymbol{x})^t \cdot \nabla_{\boldsymbol{X}} \boldsymbol{x} \quad \text{and} \quad \boldsymbol{B} = \boldsymbol{F} \cdot \boldsymbol{F}^t = \nabla_{\boldsymbol{X}} \boldsymbol{x} \cdot (\nabla_{\boldsymbol{X}} \boldsymbol{x})^t$$

Whose invariants are formally the same for both of them two, and they read as follows (indicating with $\boldsymbol{A}$ the generic matrix representing either the right or the left Cauchy-Green's deformation tensor):

$$I_1 = tr(\boldsymbol{A}) = {\lambda_1}^2 + {\lambda_2}^2 + {\lambda_3}^2$$

$$I_2 = \frac{1}{2}(tr(\boldsymbol{A})^2 - tr(\boldsymbol{A^2})) = {\lambda_1}^2{\lambda_2}^2 + {\lambda_2}^2{\lambda_3}^2 + {\lambda_1}^2{\lambda_3}^2$$

$$I_3 = det(\boldsymbol{A}) = {\lambda_1}^2{\lambda_2}^2{\lambda_3}^2$$

Where obviously with λ_i there have been indicated the eigenvalues associated to the deformation gradient tensor $\boldsymbol{F}$. Now, with the aim of showing the main features of the Mooney-Rivlin hyperelastic model, the one adopted here, some references are given concerning the following discussion. Again, the suggested presentation is directly extracted from the related Wikipedia's pages, indicated in [4] end [5] respectively as it regards the energy function introduced soon in the following and the actual Mooney-Rivlin

theory explanation, but for further details on this subject one could directly refer to the original papers by Mooney, in [56] and Rivlin, in [70]; moreover, as far as the determination of this specific hyperelastic model is concerned, some references are available, for instance, in [33, 48], whereas for some examples of the applicability of such model, see [21]. Finally, two more works are advised in order to provide comparisons between the different hyperelastic models among the most spread ones that could be found in the literature: in [50] a general introduction to several models is carried out, while in [51] an applied case of study is approached disposing of some of them.

Resuming the discussion over the derivation of the Mooney-Rivlin Hyperlestatic model, we now introduce the following scalar quantity named *strain energy density function*, that is:

$$W = \widehat{W}(\boldsymbol{C}) = \overline{W}(\boldsymbol{B}) = \widetilde{W}(\boldsymbol{F})$$

The latter can therefore be expressed as a function of the eigenvalues of the Cauchy-Green's deformation tensors, and thus of their invariants presented above. Then, taking the partial derivatives of this quantity with respect to those invariants, which means to take the derivative with respect to strains, one finally obtains the *tension tensor*. For the Mooney-Rivlin model case, we have that the strain energy density function can be written as:

$$W = c_1(\bar{I}_1 - 3) + c_2(\bar{I}_2 - 3) \tag{2.36}$$

Where $\bar{I}_1$ and $\bar{I}_2$ are the first two invariants of $\bar{\boldsymbol{B}} = det(\boldsymbol{B})^{-\frac{1}{3}} \cdot \boldsymbol{B}$, thus:

$$\bar{I}_1 = J^{-\frac{2}{3}} I_1$$
$$\bar{I}_2 = J^{-\frac{4}{3}} I_2$$

And where, in turn, there appears the quantity $J = det\boldsymbol{F} = \lambda_1\lambda_2\lambda_3$. Actually, equation 2.36 is a particular case of the more general expression, due to Rivlin, of the generalized polynomial hyperelastic model, which reads:

$$W = \sum_{p,q=0}^{N} c_{p,q}(\bar{I}_1 - 3)^p(\bar{I}_2 - 3)^q - \sum_{m=1}^{M} D_m(J-1)^{2m}$$

In case of incompressible materials, one primarily has that $J = 1$, and thus it follows that both $\bar{I}_1 = I_1$ and $\bar{I}_2 = I_2$, and also $\bar{\boldsymbol{B}} = \boldsymbol{B}$. Furthermore, being:

$$c_1 = \frac{\partial W}{\partial \bar{I}_1} \quad c_2 = \frac{\partial W}{\partial \bar{I}_2}$$

$$2D_1(J-1) = \frac{\partial W}{\partial J}$$

Thus the tension tensor:

$$\boldsymbol{\sigma} = \frac{2}{J}\left(\frac{1}{J^{\frac{2}{3}}}\left(c_1 + \bar{I}_1 c_2\right)\boldsymbol{B} - \frac{1}{J^{\frac{4}{3}}}c_2\boldsymbol{B}\cdot\boldsymbol{B}\right) + \left(2D_1(J-1) - \frac{2}{3J}\left(c_1\bar{I}_1 + 2c_2\bar{I}_2\right)\right)\boldsymbol{I} \tag{2.37}$$

With the simplification introduced for the case of an incompressible material, expression 2.37 becomes:

$$\boldsymbol{\sigma} = 2(c_1 + I_1 c_2)\boldsymbol{B} - 2c_2\boldsymbol{B}\cdot\boldsymbol{B} - \frac{2}{3}(c_1 I_1 + 2c_2 I_2)\boldsymbol{I} \tag{2.38}$$

And, to conclude, by observing that $\boldsymbol{B}^{-1} = \boldsymbol{B}\cdot\boldsymbol{B} - I_1\boldsymbol{B} + I_2\boldsymbol{I}$, then expression 2.38 simplifies as:

$$\boldsymbol{\sigma} = 2c_1\boldsymbol{B} - c_2\boldsymbol{B}^{-1} - \frac{2}{3}(c_1 I_1 - c_2 I_2)\boldsymbol{I} \tag{2.39}$$

Right this last expression 2.39 is the one adopted by the code developed in order to perform the numerical simulation of our type B aortic dissection case of study.

2.5 Multiscale Models

Within this section we will address those models commonly adopted in order to provide numerical simulations with well suited boundary conditions, which could add some new information to the phenomenon represented and ensure the well-posedness of the problem at the same time. To this aim, we will first present the most spread multiscale configurations regarding computational domains linked together with lower-order models, that could be 0-dimensional or even 1 or 2-dimensional, depending on what kind of behaviour and with how much detail they are supposed to perform. Then, special care will be given to the popular lumped parameter models, that are 0-dimensional models in which some special contribute is provided by circuital elements such as resistances, capacitors, inductors, etc. Finally, we will focus on those lumped parameter models which are the most chosen by the authors who want to carry out numerical hemodynamic simulations, the already cited Windkessel models (see section 1.3, where the boundary condition issue is faced).

2.5.1 Multiscale Configurations for Hemodynamic Modeling

As presented in the introduction to this work, section 1.2, human's cardiovascular system is a very complex machine which performs several different functions and whose components collaborate all together in order to maintain the nominal condition allowing for the sustainment of life. This network of vessels can be thought as a hydraulic system powered by a pulsating pump (the heart), and right thanks to this analogy it will be possible to devise some mathematical simplified models capable of describing, depending on chosen the level of accuracy, the behaviour of such a complex machine. Evidently, it would be too expensive, computationally thinking, the idea of performing a computational fluid dynamic simulation over the whole network composing the human's cardiovascular system, due to its complexity, to its extension and therefore to the uncountable number of degrees of freedom that such a simulation should be supposed to handle. For this reason one chooses only some interesting regions where to focus his attention, and right this latter will be the selected domains which will host numerical simulations. But it is compulsory not to forget the importance of the influence that each part of the cardiovascular system exerts on the others, as they consist of a unique connected and complex network. Thus, one aims to take into account somehow the effects of the remaining part of the cardiovascular

system on the little portion represented in his affordable domain, a 3D reconstruction of a particular region where something has to be studied or observed. The neglected portion of the cardiovascular system needs therefore to be properly summarized, in order to allow for its effect to interact with the computational domain, giving its contribute to the solution (i.e. as its boundary condition). To make all this clear, let us consider what are commonly known as *multiscale models*, that are the conjunction of different-sized models, linked all together in order to provide a more faithful result regarding a particular investigated phenomenon. Such models could be constituted of a high-dimensional domain, for instance a 3D reconstructed region of the human cardiovascular system, connected to one or more lower-order models, such as 0D, 1D or even 2D ones, which provide the proper boundary conditions for the former (in some sense, the solution obtained from the lower-order models has its own relevance too, as it represents the behaviour of the involved variables in the remaining portion of the cardiovascular system, and thus it would be as if the 3D domain provided the needed boundary conditions to these ones). Hereafter, we will refer especially to works [76] and [47], for those parts regarding the introductory concepts about the choice among the different-sized model to be selected. Later, some additional applied examples will be given about this subject.

As we anticipated, the choice for building a computational model of the cardiovascular system basically consists of four possibilities:

- **3D modeling:** exploited to study those areas of most interest, necessary to compute the solution of complex flows, such as, for instance, within heart ventricles, through cardiac valves, around bifurcations and wherever vortices or separated flow shows up. They are based on the complete set of governing equations of fluid mechanics, with all advantages and drawbacks that this implies;

- **2D modeling:** applied to those vessels which exhibit particular symmetric features, in order to represent, to cite one, the radial variation of velocity in an axisymmetric domain, and together with the following remaining categories, they are more suitable for representing the vascular network, whereas the previous ones perform better in complex geometry cases. The equations are still the Navier-Stokes ones;

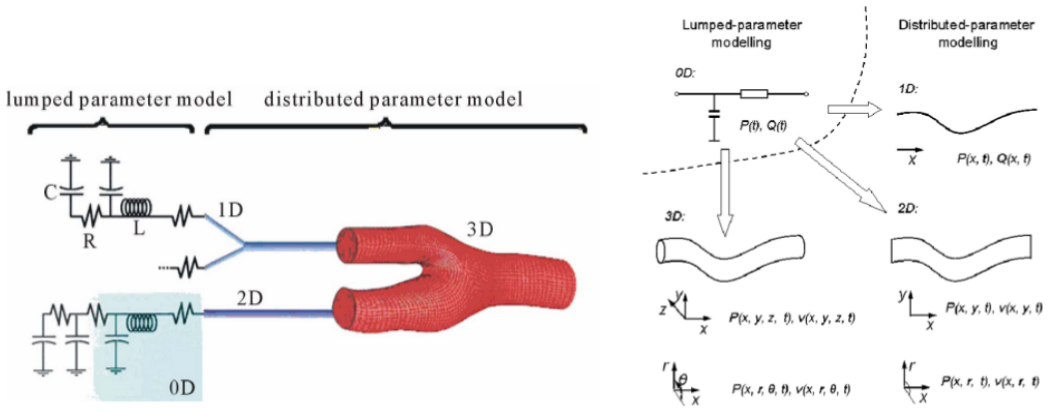
- **1D modeling:** presented more in detail in the following, they are based on a set of partial hyperbolic differential equations, and thus, considering also the spatial information concerning the solution, they are able to account for the wave transmission phenomena within human vessels, a problem which is becoming more and more interesting among researchers. In addition, they are reliable simplifications capable of providing realistic boundary conditions to the two cases mentioned above;

- **0D modeling:** This last category includes all the so-called *lumped parameter models*, constituted by a deep simplification of the represented network, which is supposed to be summarized by a group of lumped elements, fundamentally looking like circuital elements such as resistances, capacitors and inductors, but also diodes, etc., linked all together in such a way to provide the desired information, from each 0D compartment into which the global cardiovascular system is divided (from one up to several). The usual balance equations become for this simplified case

a set of Ordinary Differential Equations (ODEs), since the time remains as the only independent variable, and which can therefore describe global distributions of pressure, blood flow and volume within each compartment;

Another way we could look at those categories is the one regarding the mathematical approach applied to each of them. Facing such kind of modeling, in fact, offers basically three different strategies to researchers:

- **white boxes:** models described in detail, up to their internal structure which leads each variable from input to output sections throughout all needed relationships and iterations. This approach is certainly the most expensive and hard-coding, but also the most reliable and effective. This is the case for 3D and 2D modeling cited previously;
- **black boxes:** In contrast to the previous ones, these models aim to summarize and reproduce all transformations occurring to the involved variables throughout given compartments without explicitly defining them in detail, thus linking outputs and inputs by introducing some transition relationships properly devised depending on the single compartment represented. This is the case of 0D lumped parameter models;
- **grey boxes:** a combination of the two previously mentioned approaches, which exploits the knowledge of the internal structure of each compartment, without explicitly showing it, to define the simplified structure of the model;



(a) *Reference in [47]* (b) *Reference in [76]*

Figure 2.5: Examples of lumped parameter and distributed modeling.

In order to give a graphic idea of what has been discussed in this section, see figure 2.5 and table 2.1. From the multiscale modeling perspective, special care is then needed in handling the coupling between the different models: oftentimes this aspect could lead to ill-posed problems, which has to be detected and properly avoided or managed in advance, in order to design solid and reliable methods. These situations arise directly because of the fact that coupling two different-sized models means to impose boundary

Table 2.1: Sum up of multiscale modeling approaches.

Model	Application - pros & cons
3D	Study of the local flow through the solution of the governing equations in complex and reconstructed three-dimensional geometries; high detail but high computational effort requested.
2D	Study of the local flow through the solution of the governing equations in simplified geometries, under the assumption of some hypothesis concerning the domain's geometry such as axisymmetric cases; high detail and reduced cost, but limited applicability.
1D	Study of wave transmission phenomena such as wave reflections within the human vessels, thanks to the hyperbolic nature of the equations involving also the spatial coordinate. Simple and useful in providing boundary conditions for higher order models.
0D	Global description of entire portions of the cardiovascular system through a limited number of lumped elements, governed in time by a set of ODEs describing the evolution of the involved variables for each compartment; excellent to provide boundary conditions, easy to implement and reliable if a high level of detail is reached in representing the entire system.

conditions on both of them, which are one the solution of the other, respectively, and have therefore to be in good agreement (in case of 0D models, we rather deal with an initial condition). These conditions could refer to the direct imposition of the hemodynamic variables involved, or of their derivatives, or even combinations of them. Usually, when imposing the solution of a lower-order model as a boundary condition for a higher-order one, one has to apply such values, for instance, on the inflow section of the computational domain of the latter, assuming a proper distribution (uniform for the pressure, traditional profiles for velocity, etc.) which allows for the method to keep stable. Moreover, on the other hand, when the opposite situation occurs, one has to deal with integrations and averages of the outflowing variables, considering for instance the outflow section as connected to the entrance of a lower-order model, in order to provide the latter with its needed conditions. A proper and effective strategy needs thus to be devised.

Since 0D lumped parameter models will be introduced deeply in detail in the next section, and considering the limited applicability of 2D models (surely not suitable for our aortic dissection problem) and the fact that 3D models are nothing but a traditional CFD simulation, we now focus briefly on the presentation of some examples of applied 1D models, with the aim of underlining the main differences between them and 0D lumped parameter ones. Resuming what has been anticipated previously in this section, the study of propagating waves within the human vessels, that are pressure and flow waves, is becoming more and more an interesting issue approached by several researchers. The understanding of those waveforms is believed to be able to provide essential information concerning the cardiac functionality, the elastic properties of the vessels themselves, and also other physiological conditions related to the most important organs directly

communicating with the cardiovascular system. Many authors stated that a dense network of 0D lumped model compartments could be thought, in the limit of the approximation, as a representation of a 1D model (in this sense, we would have a first order approximation), as the information and the subsequent solution would travel through each compartment, evolving continuously in time and thus following somehow the spatial evolution that a 1D model would predict for that portion of the system. Actually, the main difference between 1D distributed models, governed by a set of hyperbolic partial derivative equations, and 0D lumped parameter ones, described by a set of time depending ordinary differential equations, instead, is the fact that the former can include the convective and non-linear contribute, whereas the latter are not concerned with such an issue. Due to their relatively low computational cost, if compared with higher-order 2D and 3D distributed models, 1D models are really fascinating when thought to compute the evolution of pressure and flow along the investigated vessel, especially considering the fact that, as it arose from some studies on the subject of 2D axisymmetric Navier-Stokes equations, provided that the vessel's radius is small enough when compared to a characteristic wavelength, the pressure is almost constant over any cross-section area of the channel and, meanwhile, the radial component of the velocity is negligible almost everywhere on those sections other than within very thin regions. For these reasons, the 1D formulation of the governing equations associated with a 1D distributed model of human vessels, regarding pressure, flow and cross-section area as main variables, is a valid solution commonly accepted within this research field.

To cite some applications of those models, one could find them in several works (listed in [76]) such as in the investigation of pulse wave dynamics in various vessel's segments of a complete human arterial network, in the study of pulse wave features in the pulmonary arterial network, or in arteries with stenosis, or in arterial vessels with bypass grafts, or again in the investigation of the pulse wave transmission in stenotic arteries with implanted stents, or in the pulse wave propagation in a detailed systemic arterial tree, or even within collapsible vessels. Another study exploiting 1D distributed models regards the so-called wave-intensity analysis,defined as the product of pressure and velocity.

2.5.2 Lumped Parameter Models

In this section we aim to focus more deeply on what have been defined as *0D lumped parameter models*, before going straight to the introduction of the well known Windkessel models, those adopted for the purposes of this work. This will therefore be only a brief transitional introduction, given in order to present the fundamentals of the analysis that constitutes the basis for the derivation of such models, as well as to present also more complicate and detailed examples of these very useful tools, as it will be shown soon.

As a matter of fact, simplifying a real system, such as the human cardiovascular system, throughout the adoption of a low-order model, could turn out to be incredibly helpful and meaningful in order to understand what the actual functionality of that complex system is. Lumped parameter models built with the purpose of imitating the behaviour of the human cardiovascular system are based on an electric-hydraulic analogy, assuming that the blood flow, the liquid flow within a pipe, would be comparable to the current intensity in an electrical circuit, while the blood pressure, or the pressure of the liquid flowing within the

pipe, could be represented by the electric voltage as well. Let us consider the pressure drop encountered in the human cardiovascular system: starting from the aortic channel connected to its heart chamber, the left ventricle, which could be thought as the pump responsible for the movement of the blood all over the systemic vasculature, and therefore as a tension generator in an electric circuit, then one goes down all the vascular system, towards the venous compartments, where the pressure level has reached very lower values, and then up again to the heart pump. All along this path, the blood pressure is relieved and its oscillating behaviour is dumped by the elasticity of large vessel's walls, which allow for the blood to be stored during cardiac systole, reducing the peak of pressure, and then ejecting it again during diastole, regularising that way the resulting pressure signal. Moreover, due to the friction exerted by all vessel's walls on the fluid itself, depending on its velocity, pressure is reduced again as we go down the systemic vasculature. All these behaviours could be mathematically represented, in terms of the electrical analogy we were carrying out, by introducing some circuital elements which are able to reproduce the same effect on an oscillating voltage signal (or, equivalently, on an oscillating current). These elements are: capacitors, which can reproduce the so-called *compliant* feature associated with the largest vessels, and thanks to which their deformation allows for the pressure signal to be properly regularised; resistances, which can mimic the frictional effect exerted by vessel's walls on the stream and which causes the subsequent pressure drop; inductances, which, in turn, can take into account the inertial properties of the blood while it flows along the largest vessels, when its oscillating behaviour has not been properly dumped yet. Beside these first elements here cited, we can find other devices, such as diodes, which could also take into consideration the functionality of cardiac valves, or even venous ones, but which will not be considered for the aim of this work. To sum up what has been said on the electric-hydraulic analogy up to this point of the discussion, we refer to table 2.2. Before concluding this brief and introductory part concerning lumped

Table 2.2: The electric-hydraulic analogy.

Hydraulic	**Physiologic**	**Electric**
liquid pressure $p\,[Pa]$	blood pressure $p\,[mmHg]$	voltage $V\,[V]$
liquid flow rate $Q\,[m^3\,s^{-1}]$	blood flow rate $Q\,[ml\,s^{-1}]$	current $i\,[A]$
liquid volume $\mathcal{V}\,[m^3]$	blood volume $\mathcal{V}\,[ml]$	charge $q\,[C]$
wall's friction	vessels' friction	electric resistance $R\,[\Omega]$
air reservoir chamber	vessels' compliant walls	capacitor's capacity $C\,[F]$
liquid's inertia	blood's inertia	inductor's inertance $L\,[H]$
Poiseuille's law	Poiseuille's law	Ohm's law

parameter models in general, we highlight the main distinction criteria usually adopted in order to classify them. Depending on the number of elements introduced to reproduce the entire cardiovascular system, or only a portion of it, and thus on the degree of detail which we aim to reach, one can distinguish substantially between two families of lumped models:

- **single-compartment models:** those where the entire cardiovascular system, or a

portion of it, is supposed to be summarized by only one compartment of lumped elements, that is a single group composed by a resistor, a capacitor and, if necessary, an inductor. It is therefore clear enough that the depth of detail, as well as the ability of such a simple model of representing an entire complex system, the human's one, will be strictly limited, and the possibility of extracting somehow information regarding the wave transmission phenomena is, obviously, excluded. To this division belong all the first and simplest Windkessel models that will be introduced later in the next section, appositely dedicated. Even though the great simplicity of these models could seem excessively constraining in terms of reliability ad faithfulness of their results, they still represent a widely adopted solution, especially concerning their application as boundary conditions for higher order distributed models in order to provide them with realistic pressure or flow information, such as the case of our work (examples reported in figure 2.6);

- **multi-compartment models:** they still consist of lumped parameter models, but organised in many more different compartments, that is multiple combinations of resistor-capacitor-inductor elements, in order to mimic and represent in a more realistic manner the functionality of the different elements composing the human cardiovascular system. Depending on the degree of detail one wants to reach, a number of circuital elements are chosen and, subsequently, a corresponding number of different compartments can thus be defined. As we discussed previously, 0D models cannot take into consideration wave transmission phenomena, although a multi-compartment model with a higher number of elements describing its internal structure can somehow represent also the spatial evolution of the involved variables, and could thus be helpful in determining some initial information about signals transport. A common way of organising this kind of models could be setting up a different compartment for each portion into which the cardiovascular system can be distinguished, starting from the heart chambers, then proceeding with the aortic compartment, the large arteries one, the arteries' and arterioles', then there follow the capillary compartment and the veins one, and finally the large veins'. See figures 2.7 and 2.8 as examples (the last is taken from professor S. Scarsoglio's work in [71]).

2.5.3 Windkessel Models

Among the above-mentioned lumped parameter models, those which have acquired the most popularity and relevance are the so-called Windkessel ones. As anticipated, they consist of a circuital representation of the human vasculature according to an electric-hydraulic analogy, based on the purpose of taking into account the resistance exerted by vessels' walls on the blood flowing through them, and the compliant behaviour of large arteries, responsible for the regularisation of the pressure signal arising from the left ventricle stream ejection and relaxation during systolic and diastolic phases of the cardiac cycle. According to the literature, the first who suggested the adoption of such simplified models, in their most essential conception, were S. Hales (1733) and W. E. Weber (1827), to whom is due the first attempt to compare the functionality of the human cardiovascular

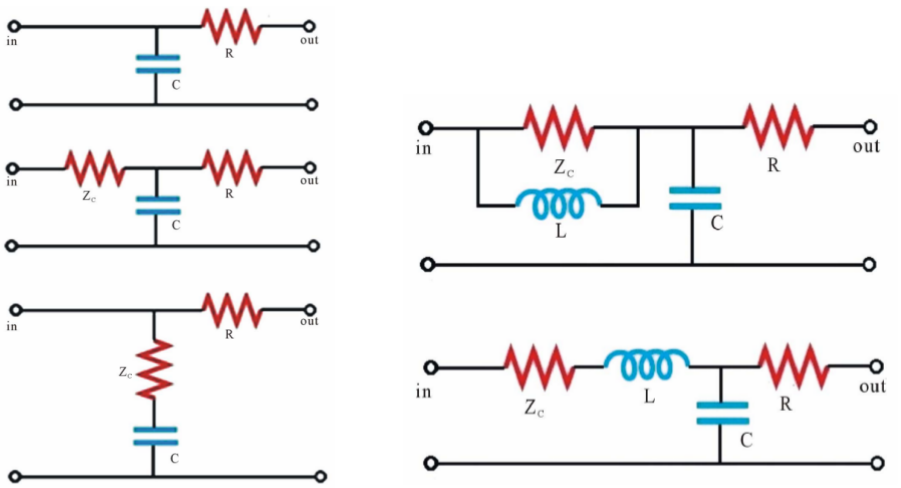


Figure 2.6: Single-compartment 0D modeling examples: 2, 3 and 4-elements Windkessel models.

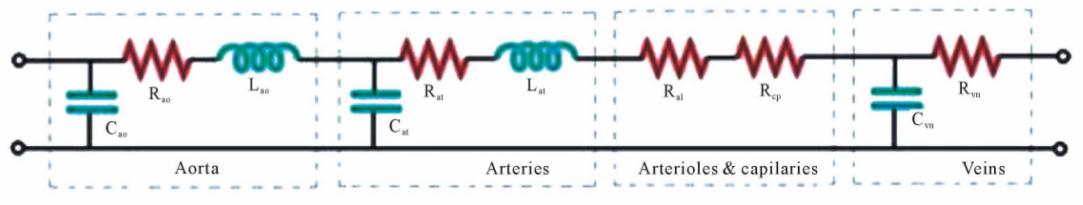


Figure 2.7: Multi-compartment 0D modeling example.

system to that of a *windkessel* appearing in fire engines, as depicted in figure 2.9. This analogy suggested that pressure variations in large arterial compartments were related to their own elasticity, i.e. their compliant features. What follows could be found, other than in the previously mentioned works [47] and [76], also in [82] and in the master's thesis work [60]. Other references to research thesis in order to provide different applied examples of these widely adopted models could be found in [58], where the author adopted the 3-element Windkessel model as boundary condition for abdominal aortic aneurysms simulations (see figure 2.10), and in [34], where those models have been applied to the investigation of cardiac output and arterial resistance by the computation of arterial blood pressure, again exploiting a modified version of the 3-elements Windkessel model. A graphic anticipation of the following three Windkessel models presented, together with their hydraulic analogous, is then reported is figure 2.11.

Two-elements Windkessel model:

Then, in 1899, Otto Frank finally proposed the first mathematical formulations of these lumped parameter models,setting up and quantitatively formulating the well known *2-elements* Windkessel model, simply constituted of a resistor and a capacitor element linked

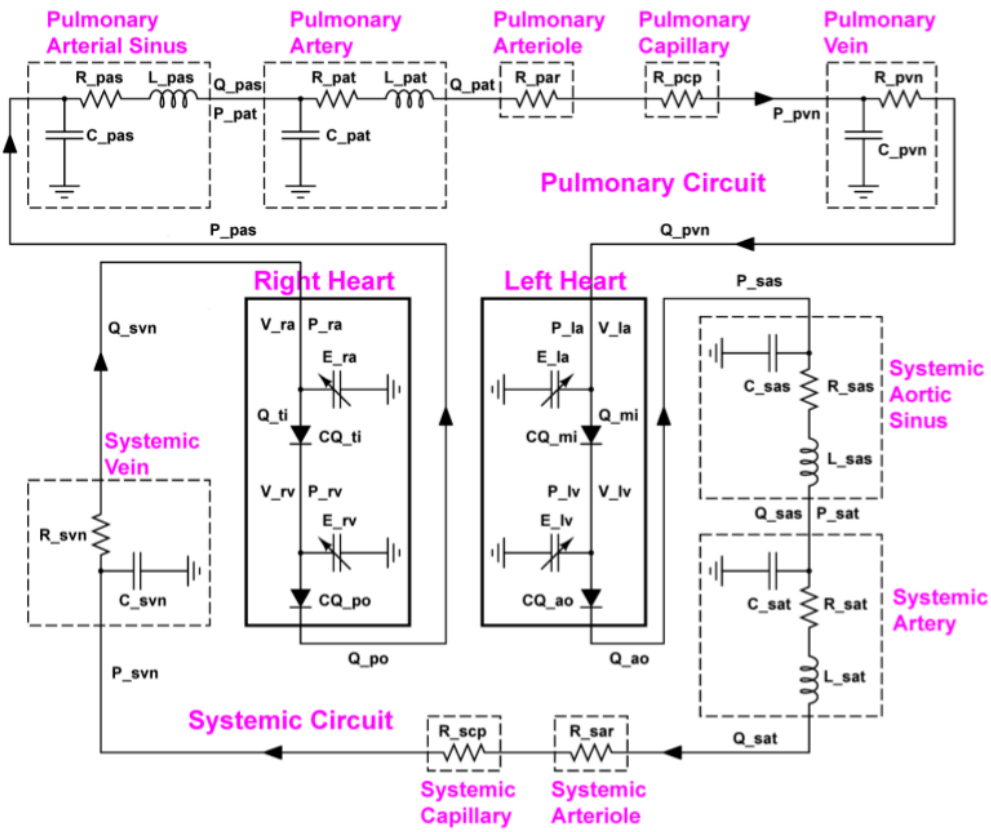


Figure 2.8: Multi-compartment 0D modeling of the entire human cardiovascular system.

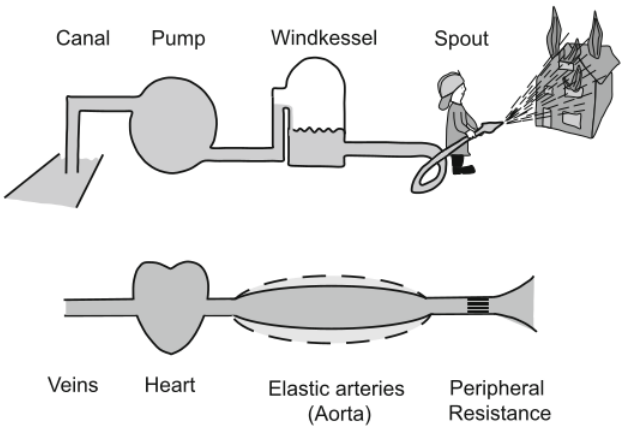


Figure 2.9: Analogy between the functionality of the human cardiovascular system and a fire engine scheme.

in parallel, as reported in figure 2.12. The first of them, the resistance, was introduced in order to consider all the resistance contributes of the human vasculature, from the

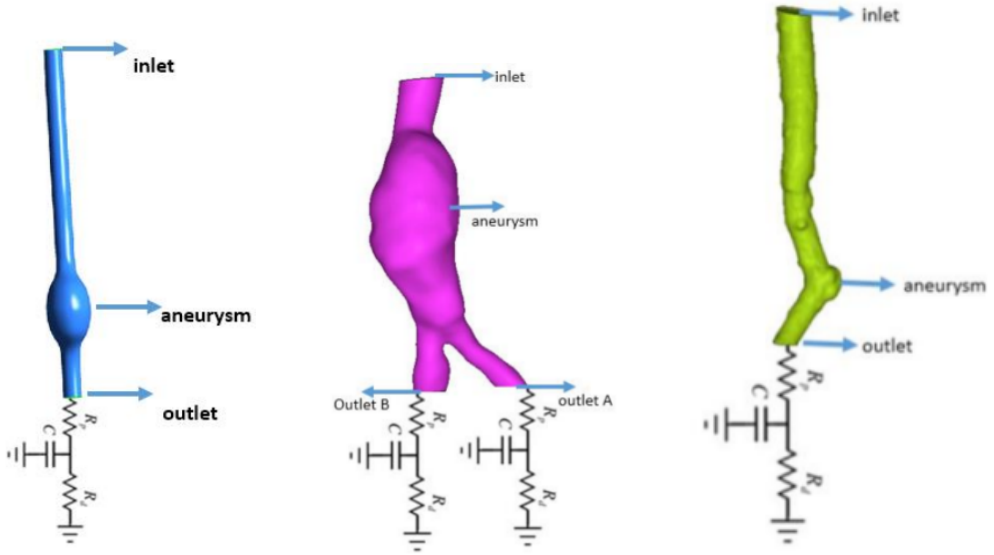


Figure 2.10: Applied examples of WK3 models as boundary conditions for an abdominal aortic aneurysm simulation.

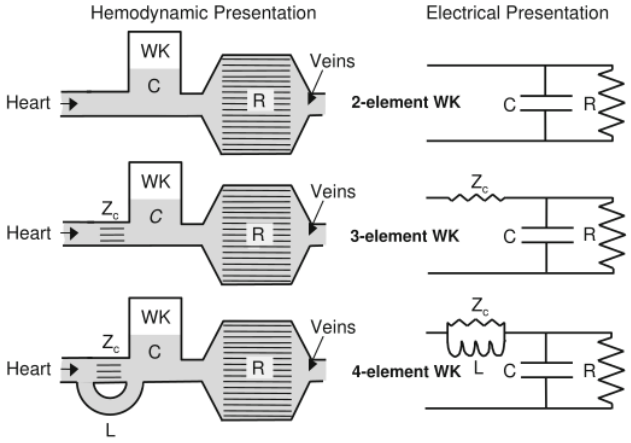


Figure 2.11: Analogy between different Windkessel models and their respective hydraulic representation.

largest vessels to the smallest ones, according to Poiseuille's law which states that such feature is proportional to the inverse of the fourth power of the channel's radius, and thus primarily referred to the smaller arterioles and capillary vessels. This global resistance R was therefore named *total peripheral resistance*, as it referred to all the contributes given by the peripheral vasculature summarized by the model, and could be calculated as:

$$R = \frac{p_{aortic,mean} - p_{vein,mean}}{CO} \approx \frac{p_{aortic,mean}}{CO}$$

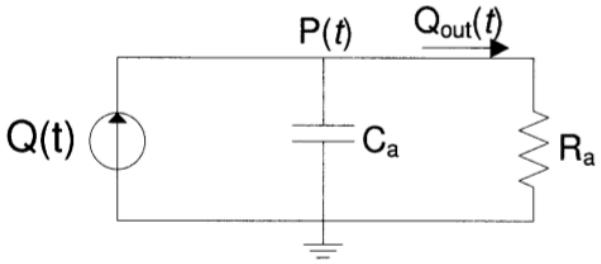


Figure 2.12: Circuital scheme of a 2-elements Windkessel model.

Where with CO has been denoted the *cardiac output* ejected from the left heart ventricle, that is the product between the *heart rate* and the *stroke volume* (difference between the volume of the ventricle in end-diastolic condition and in the end-systolic condition). As one can notice from the relationship above, the venous system was assumed to be a 0-pressure far field condition, for simplicity. The second element, instead, the capacity, was supposed to replace the compliant feature of the largest arteries, caused by their elastic behaviour in response to pulsating flow, such as the blood ejected through the aortic valve by the left ventricle during systole, and its subsequent closure during diastole. This capacity estimation could be obtained by summing all the compliances related to all human vessels, as they were connected in parallel one another, thus leading to the so-called *total arterial compliance*, C, which could be estimated as:

$$C = \frac{\Delta \mathcal{V}}{\Delta p}$$

The previous relation corresponds to the variation of pressure due to the related blood volume one, even though such expression turns out to be quite useless, since it would be almost impossible to perform experiments pointing at the determination of a pressure variation, following a volume injection of blood, as the latter would be very difficult to track along the human vasculature. Therefore, an alternative way of estimating that parameter needed to be devised, such as by studying the pulse wave velocity in the aorta or by considering the exponential pressure decay characterising the diastolic cardiac phase, depending on the characteristic time $\tau = RC$.

By referring to figure 2.12, we now aim to derive the mathematical relationship which governs this 2-elements essential model. The heart is represented as a pumping source of current $Q(t)$, while the circuital elements associated respectively to the total arterial compliance, the capacity C_a, and to the total peripheral resistance, the resistor R_a, are linked in parallel, and whose exit pressure signal, the one we are mostly interested in, is indicated with $P(t)$. Furthermore, recalling the constitutive relations which govern those circuital elements, that are the analog Ohm's law for the resistor:

$$Q = \frac{\Delta p}{R}$$

And, by deriving with respect to time the definition of capacity given above, the capacitor

constitutive's one:

$$Q = C\frac{\mathrm{d}p}{\mathrm{d}t}$$

Then, by applying the circuital Kirchhoff's current law, one could write the following relationship:

$$Q(t) = C_a\frac{\mathrm{d}P(t)}{\mathrm{d}t} + \frac{P(t)}{R_a} \tag{2.40}$$

That represented by equation 2.40 is a very simple and easy-handling model, also quite straightforward to calibrate and whose results are a first useful set of data if one needs to provide a higher-order model with meaningful boundary conditions. However, among its weaknesses, this model cannot represent any effect related to wave transmission, propagation and reflection phenomena, assuming that the pressure computed in time is the same all over the summerized vascular bed. Moreover, while the pressure decay corresponding to the cardiac diastole is quite well predicted by this simple scheme, the same does not hold for the systolic pressure rise (see figures 2.15 and 2.16).

Three-elements Windkessel model:

To the end of improving the faithfulness with respect to the realistic aortic pressure signal coming from these models, the 3-elements Winkessel one has been devised. This new and revisited version of the old Frank's 2-elements one is composed by the introduction of an extra impedance element in front of the parallel combination of the peripheral resistance and the aortic compliance. Actually, this last element is not a mere resistance, and should therefore be interpreted from an oscillating point of view. Anyway, for the sake of simplicity and since its modulus represents only about 6-7% of the total peripheral resistance, it will be here considered as a pure resistance. This innovative model, first thought by G. Landes and N. Westerhof, and for this reason also known as *Westkessel model*, has gathered more and more attention and popularity among researchers, becoming the most adopted of the developed Windkessel models, because of its reliable and easy-implementing features and for his faithfulness with respect to realistic data. The circuital scheme of this kind of model is depicted in figure 2.13. The primary goal of such scheme is to take into account

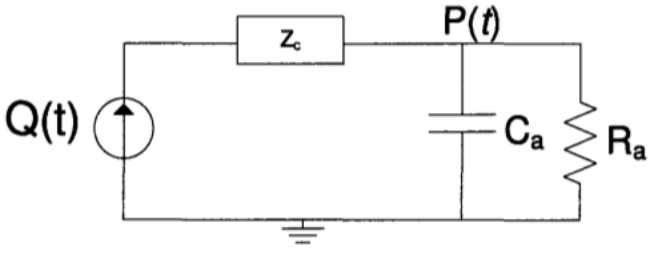


Figure 2.13: Circuital scheme of a 3-elements Windkessel model.

and to represent, throughout the definition of this new impedance, also the travelling waves phenomena and effects on the outcomes of the lumped model, since it is defined as:

$$Z_c = \frac{c\rho}{A}$$

Where ρ and A refer, respectively, to the blood density and to the aortic cross-section area, whereas with c the travelling wave speed is indicated. Despite of its still very simple structure, this new model constituted by the introduction of this additional element, hereafter called *characteristic impedance*, it has been proved to greatly improve the performance of the lumped model itself, especially when higher working frequencies are considered. In that case, it has been shown that the modulus of this input impedance added to the model, at high frequencies, equals the characteristic impedance of the proximal aorta, and for this reason we will occasionally refer to it as the *proximal resistance*, simply.

Following the same procedure adopted above for the 2-elements model, we now point at deriving the pressure-flow relationship which holds in the case of this new and improved model. Referring to the same notations, and by indicating with Q_R and Q_C respectively the flow rate crossing the two branches of the circuit hosting the aortic capacity C_a and the peripheral resistance R_a, with P the pressure before the parallel composed by the last two, and finally with P_{wk3} the final upstream pressure in front of the input proximal impedance Z_c, one has that, thanks to the Kirchhoff's current law:

$$Q(t) = Q_R(t) + Q_C(t) = C_a \frac{\mathrm{d}P(t)}{\mathrm{d}t} + \frac{P(t)}{R_a}$$

And, since (application of Kirkhhoff's tension law):

$$P_{wk3}(t) = P(t) + Q(t)Z_c$$

Thus, it follows that, after some rearrangements:

$$P_{wk3}(t) = Q(t)(R_a + Z_c) - C_a R_a \frac{\mathrm{d}P_{wk3}(t)}{\mathrm{d}t} + C_a R_a Z_c \frac{\mathrm{d}Q(t)}{\mathrm{d}t} \tag{2.41}$$

This last expression 2.41 will be resumed and further developed in the following sections, since it is the one which will be adopted as boundary condition for the considered aortic dissection problem, and therefore it will be discretized and implemented in order to provide the 2D model with a proper outflow condition, ensuring its well-posedness as well as high level of faithfulness with respect to reality. The main differences between this version and the other Windkessel schemes considered in this section could be found in figures 2.15 and 2.16.

Beside this one presented here, other different version of 3-elements Windkessel models do exist in the literature (we recall, for instance, the one reported in figure 2.6, down on the left). They will not be reported here for brevity, but one could easily find some examples in those works previously cited, e.g. in [76].

Four-elements Windkessel model:

Another improvement can be found among all proposed Windkessel schemes in the literature, first introduced by N. Stergiopulos and again G. Landes, consisting in the addition of a fourth element to the previous 3-element ones, that is an inductor capable of taking into consideration also blood's inertia, and so its inertial properties and behaviour when a pulsating (i.e. oscillating) flow is imposed through the circuit. The improvement

especially concerns the low and middle frequency range of work, where the introduction of the characteristic input impedance lacked in accuracy, recovering a good agreement with experimental data, as shown in figure 2.14. Several versions of this last 4-elements scheme have been proposed by authors, and all of them agree on the fact that introducing such additional element is never worth the additional complexity of the model itself, in particular, as far as determining and tuning circuital element's values is concerned. This is also why we chose to adopt a simpler 3-element model for this work. The total arterial

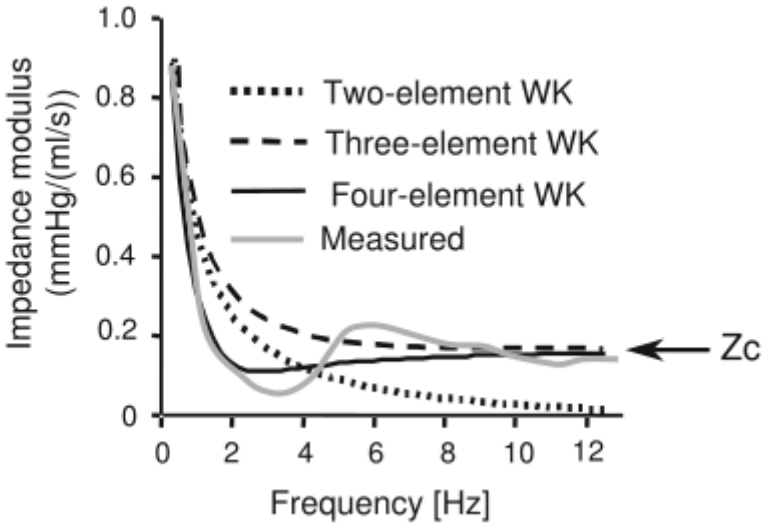


Figure 2.14: Comparison of the different introduced Windkessel models over the estimation of the input impedance with respect to experimental data, in the frequency domain.

inertance depends only on the geometrical properties of the considered vessels, such as their length, their cross-section area and, of course, on blood density. Thus, by indicating with L the vessel's inertance, one has the following definition of such property:

$$L = \frac{l\rho}{A}$$

Where again with A is indicated the cross-section area of the vessel, with l its length, and with ρ the blood density. For reasons of brevity, the derivation of the relationship linking blood pressure and blood flow will not be reported here, but we address to the thesis work [60, p. 35-37] in order to find its development procedure. Finally, to provide a comparison between the different estimation of the aortic pressure signal during the cardiac cycle performed by the different models here proposed, we refer again to figures 2.15 and 2.16, and in table 2.3 are listed and summarized properties, advantaged and disadvantages of each of the considered models.

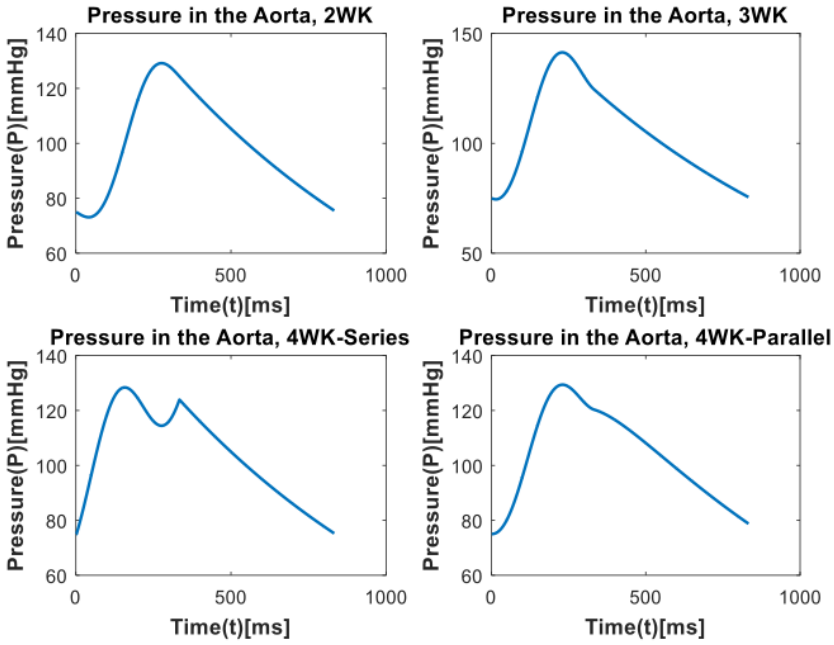


Figure 2.15: Comparison between the pressure signal obtained adopting different Winkessel models (1), reference in [60].

Table 2.3: Windkessel different models comparison.

Feature	**2-elements WK**	**3-elements WK**	**4-elements WK**
diastole waveshape	RC decay	RC decay	RC decay
systole waveshape	poor	good	good
input impedance approx.	poor at high freq.	small error at low freq.	good at all freq.
compliance estimation	good	overestimation	good
inertance estimation	not involved	not involved	tough to estimate

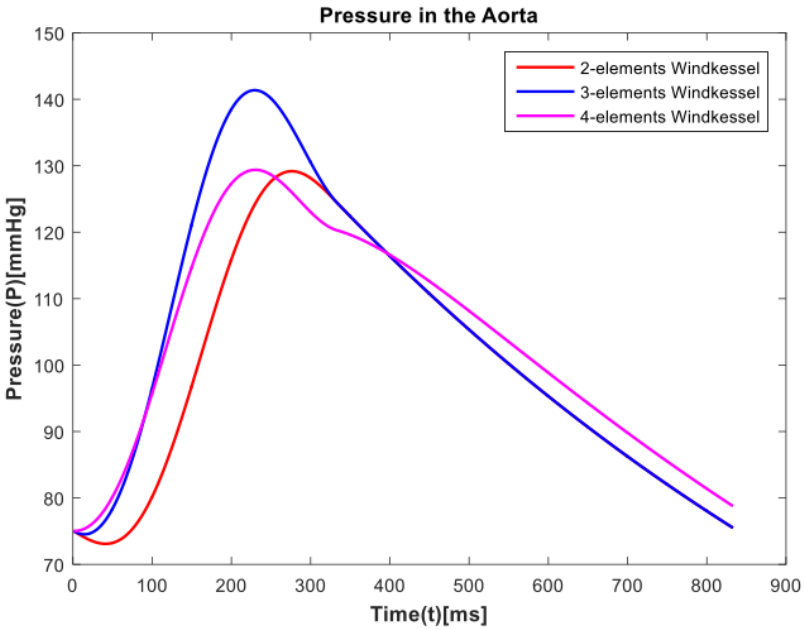


Figure 2.16: Comparison between the pressure signal obtained adopting different Winkessel models (2), reference in [60].

Chapter 3

Numerical Methods

In this chapter, we will recall the most spread numerical methods developed so far with the aim of solving numerical PDEs problems. Among them, one can typically find Finite Differences Methods, Finite Volume Methods and Finite Element Methods, which will be briefly presented in the first section of the chapter. Then, in particular, numerical approaches developed in order to face incompressible problems will be introduced, with special focus on Fractional Step Methods. For the sake of completeness, an additional section is also included regarding again the matter of posedness of such numerical problems and the proper treatment and definition of their boundary conditions, depending on the specific method selected from those previously listed.

3.1 Numerical Methods for PDEs

As anticipated, within this section the most popular numerical methods for solving PDEs will be resumed, and some fundamental schemes will then be provided since directly involved in this master's thesis work. For further details on the subject, as well as for a complete and rigorous description of the problem itself and of its treatment here simply summarized and reported in terms of its cardinal steps, we address to the two works by A. Quarteroni in [67] and [66], but an introduction of these fundamental methods could be found in any Numerical Analysis or Applied Mathematics text; furthermore, we will also refer to some slides presented by prof. S. Pieraccini and prof. S. Berrone (Polytechnic of Turin) during Numerical Methods and Scientific Computing classes (reference in [63]), and again to the slides by prof. D. D'Ambrosio on Computational Fluid Dynamics (in [28]). Refer to section 2.1 for an introduction to differential problems and PDEs classification.

3.1.1 Finite Difference Method

When one has to deal with the numerical solution of a PDEs problem, such as those presented in section 2.1 also recalled above, those introduced properties must be taken into consideration. This means, to resume what was stated in that section, that the method needs to be *consistent* and *stable*, so that it can be said to be *convergent* too. In addition, a numerical method has to be *accurate* in order to provide better and reliable

results, and right to this end we will also apply the definition of *order of accuracy* of a numerical scheme. All these issues apply for each of the considered methods regarding PDEs resolutions, but we will particularly focus on Finite Difference Methods since they are those mainly considered in the present work.

Spatial Schemes:

Let us consider a function ϕ of the independent variable x, defined on a domain D, so that:

$$\phi : D \subseteq \mathbb{R} \longrightarrow \mathbb{R}, \quad x \longmapsto \phi(x)$$

Whose graph is depicted in figure 3.2. Then, the first derivative of this function, with respect to its only variable, is an object defined as:

$$\phi'(x) = \frac{\mathrm{d}\phi(x)}{\mathrm{d}x}$$

Or, in case ϕ was a multi-variable depending function, for instance $\phi(x,t)$, we should speak about its partial derivatives:

$$\phi_{,x}(x,t) = \frac{\partial \phi}{\partial x} \qquad ; \qquad \phi_{,t}(x,t) = \frac{\partial \phi}{\partial t}$$

These, together with higher order derivatives of ϕ, are exactly the quantities appearing in PDEs example problems reported in section 2.1, such as in Poisson's equation, transport equation, etc. Obviously, in order to allow for a calculator to compute a numerical solution of such differential problems, one needs to approximate somehow those differential operators. For this reason *finite differences* have been devised, with the aim of approximating numerically the derivatives of functions involved in differential problems of any sort, thus giving a representation of those derivatives up to a numerical error which has, of course, to be properly controlled, and to finally allow for their resolution, and thus leading to an approximate result of the considered problem. In a 1D domain problem, let us consider

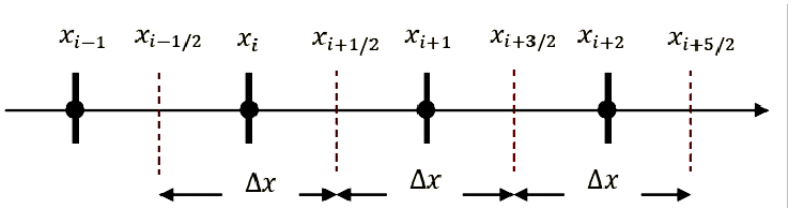


Figure 3.1: 1D spatial domain uniform discretization.

a space discretization of the oriented horizontal axes as reported in figure 3.1, which consists of a uniform repartition of the space using $N+1$ points, thus dividing the whole domain in N uniform intervals of length $\Delta x = \Delta x_{i+1} = x_{i+1} - x_i$, $\forall i \in [0, N-1]$. We can now introduce an approximation of the spatial partial derivative of ϕ, that is $\phi_{,x}$, by approximating it with the differential quotient computed using its values evaluated at two different points of the considered 1D grid. For example, if one wants to find an

approximation of the derivative $\phi_{,x}(x_i = i\Delta x, t)$, this can be done by computing the differential quotient between the point x_i and the following one x_{i+1}, then:

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi(x_{i+1}) - \phi(x_i)}{x_{i+1} - x_i} = \frac{\phi(x_{i+1}) - \phi(x_i)}{\Delta x} \tag{3.1}$$

The latter is the so-called *1st order forward difference approximation* of the first spatial derivative of ϕ, and, in other words, it can be represented through the slope of the line linking points $\phi(x_{i+1})$ and $\phi(x_i)$, as shown in figure 3.2. A similar result can also be obtained by choosing points $\phi(x_i)$ and $\phi(x_{i-1})$ instead, thus leading to the so-called *1st order backward difference approximation*, that is:

$$\frac{\partial \phi}{\partial x} \approx \frac{\phi(x_i) - \phi(x_{i-1})}{x_i - x_{i-1}} = \frac{\phi(x_i) - \phi(x_{i-1})}{\Delta x} \tag{3.2}$$

Hereafter, we will refer to those numerical derivatives by indicating them with the notation $\frac{\partial \phi}{\partial x} = \left(\frac{\partial \phi}{\partial x}\right)_i = (\phi_{,x})_i$, and the function evaluations with $\phi(x_i) = x_i$, for brevity. Another scheme which is possible to be easily introduced regarding the approximation of the first spatial derivative of a function is the so-called *2st order central difference* one, involving points in front and back the considered x_i, that are the points x_{i+1} and x_{i-1} previously encountered. The scheme thus looks like:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{\phi_{i+1} - \phi_{i-1}}{x_{i+1} - x_{i-1}} = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \tag{3.3}$$

The here presented three schemes refer all to the approximation of a first order spatial derivative, but the same results hold in case of a time derivative, of course (next paragraph). Furthermore, their graphic interpretation can be found again in figure 3.2. We anticipated that the central scheme 3.3 is second order accurate, while the previous two 3.1 and 3.2 are only first order accurate, and this can be proved by writing the Taylor approximation of the function ϕ centered in x_i, and evaluated at the surrounding points x_{x+1} or x_{i-1}, which read:

$$\phi_{i+1} = \phi_i + \left(\frac{\partial \phi}{\partial x}\right)_i \Delta x + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i \Delta x^2 + \mathcal{O}(\Delta x^3) \tag{3.4}$$

$$\phi_{i+1} = \phi_i - \left(\frac{\partial \phi}{\partial x}\right)_i \Delta x + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i \Delta x^2 + \mathcal{O}(\Delta x^3) \tag{3.5}$$

Therefore, by taking the first derivative term either from relation 3.4 or 3.5, one gets, respectively, the schemes 3.1 and 3.2 previously presented, together with the order of accuracy related to the *truncation error* that they introduce by disregarding higher order terms. Then, they read:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_i}{\Delta x} + \mathcal{O}(\Delta x)$$

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_i - \phi_{i-1}}{\Delta x} + \mathcal{O}(\Delta x)$$

To conclude, in order to prove the stated degree of accuracy of the central difference scheme 3.3, by taking the difference between 3.4 and 3.5 and again by putting in evidence the first derivative term, one has that:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

Note that these results hold as long as the grid step Δx is uniform, which will be our specific case, since this work deals with regular cartesian grids. For this reason the issue regarding non-uniform grids will not be taken into consideration within this discussion, for the sake of brevity. Also higher order derivatives have to be considered for the particular

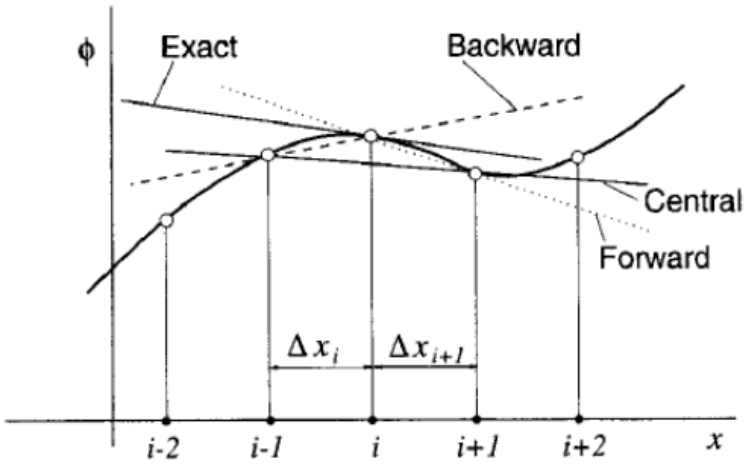


Figure 3.2: Approximation of functionϕ's spatial derivatives with backward, central or forward finite differences.

aim of this word, and a proper scheme for their approximation therefore needs to be devised. Following the Taylor expansion strategy, it is now pretty straightforward to obtain an approximation for the second spatial derivative of ϕ in x_i, that is $(\phi_{,xx})_i$, by taking this time the sum between the two relations 3.4 and 3.5, now expanded up to the third order of derivation, and then putting in evidence the second order derivative, and thus coming to the following second order accurate central scheme:

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i = \frac{\phi_{i+1} - 2\phi_i + \phi_{i-1}}{\Delta x^2} + \mathcal{O}(\Delta x^2) \tag{3.6}$$

Other than Taylor expansions, which exhibit the advantage of showing directly the order of approximation of the constructed schemes by highlighting their truncation error as a function of the grid step, we can also derive alternative (or equivalent) schemes by changing our point of view, that is, by adopting the *polynomial fitting* technique. This approach basically consists of the idea of writing an approximation of the function ϕ by recurring to a polynomial representation p_n of degree n, so:

$$p_n(x) = a_0 + a_1 x + \cdots + a_n x^n$$

This object will obviously depend on $n+1$ coefficients $(a_0, \ldots, a_n)$, and thus it will need $n+1$ independent conditions to be uniquely defined. These information come from the interpolating data of the function values of ϕ evaluated at a specific number of grid points. This approach allows to write schemes even of higher order of accuracy, but as primary drawback one has to consider the larger *stencil* extension, which could turn out to be a problem especially in terms of boundaries handling. As an example, we will now derive a second order accurate approximation for the first spatial derivative of the function ϕ, by adopting a 2nd-order polynomial such as:

$$p_2(x) = ax^2 + bx + c$$

In fact, by considering a stencil composed of three grid points $[i, i+1, i+2]$, and centering then the reference frame in $x_i = 0$, so that now $p_2(x_i = 0) = \phi_i = 0$, $p_2(x_{i+1} = \Delta x) = \phi_{i+1} = c + b\Delta x$ and $p_2(x_{i+2} = 2\Delta x) = \phi_{i+2} = 2\Delta x$, and then noting in addition the fact that $\left(\frac{\partial \phi}{\partial x}\right)_i = p_2'(x_i = 0) = b$, one has that:

$$b = \left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{4\phi_{i+1} - \phi_{i+2} - 3\phi_i}{2\Delta x} \tag{3.7}$$

The same could have been obtained by adopting again the Taylor's expansions approach, this time by considering also the approximation for the farthest point of the considered stencil, that is:

$$\phi_{i+2} = \phi_i + \left(\frac{\partial \phi}{\partial x}\right)_i 2\Delta x + \frac{1}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i 4\Delta x^2 + \frac{1}{6}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_i 8\Delta x^3 + \mathcal{O}(\Delta x^4) \tag{3.8}$$

And then taking the difference between four times expression 3.4 for ϕ_{i+1} and the latter 3.8, one has again:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{-\phi_{i+2} + 4\phi_{i+1} - 3\phi_i}{2\Delta x} + \mathcal{O}(\Delta x^2)$$

Moreover, a similar formulation can also be found for a sort of backward second order difference, which exploits this time the stencil $[i-2, i-1, i]$ still to approximate the first order derivative $(\phi_{,x})_i$. By following a straightforward and analogous procedure (this time extending the Taylor's backward expansion up to the point x_{i-1}), one gets that:

$$\left(\frac{\partial \phi}{\partial x}\right)_i = \frac{3\phi_i - 4\phi_{i-1} + \phi_{i-2}}{2\Delta x} + \mathcal{O}(\Delta x^2) \tag{3.9}$$

Resuming the second order spatial derivative, an unbalanced formulation exploiting either a forward scheme with stencil $[i, i+1, i+2]$, or a backward one with stencil $[i-2, i-1, i]$, can be written also for that object. Thus, by resorting again to polynomial fitting, for instance, one has that:

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i \approx \frac{\phi_i - 2\phi_{i+1} + \phi_{i+2}}{\Delta x^2} \tag{3.10}$$

$$\left(\frac{\partial^2 \phi}{\partial x^2}\right)_i \approx \frac{\phi_i - 2\phi_{i-1} + \phi_{i-2}}{\Delta x^2} \tag{3.11}$$

The last two schemes presented here below will be the most extensive ones, involving four grid points each, and thus reaching a third order of accuracy in space for the first spatial derivative. Because of their width, they could be difficult to handle in the proximity of the boundaries, but their exploitation all over the internal domain will be the chosen one for our case of study. Starting from the forward one, the involved stencil is composed as $[i-1, i, i+1, i+2]$, and the interpolating polynomial has the structure $p_3(x) = ax^3 + bx^2 + cx + d$, and consequently the resulting numerical scheme reads:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{-\phi_{i+2} + 6\phi_{i+1} - 3\phi_i - 2\phi_{i-1}}{6\Delta x} \tag{3.12}$$

And, in case of a backward approximation whose stencil is instead $[i-2, i-1, i, i+1]$, one has that, equivalently:

$$\left(\frac{\partial \phi}{\partial x}\right)_i \approx \frac{2\phi_{i+1} + 3\phi_i - 6\phi_{i-1} + \phi_{i-2}}{6\Delta x} \tag{3.13}$$

The latter are schemes developed in order to achieve the stability properties of the method they are based on. Indeed, it is well known the fact that evolutionary problems, when numerically faced by resorting to those approaches here presented, tend to exhibit instabilities which could lead the solution to diverge in time. This is due to the fact that the adopted scheme itself could cause the introduction of such instabilities, and of their growth in time above all, since they introduce a numerical error which tends to amplify those oscillations instead of smoothing them. This problem can be tackled by selecting proper numerical schemes, and by satisfying determinate conditions in order to introduce the same sort of error, which is actually related to the second spatial derivative of the solution and is therefore commonly named as *artificial viscosity* or *numerical viscosity*, but that this time allows for those oscillations to be dumped in time, such that the solution remains stable. Obviously; these schemes lead to less accurate solutions, since these numerical diffusive terms tend to smooth the entire solution over the whole domain as the simulation runs, exactly as if it was caused by the actual physical viscosity. Among these kind of schemes, one can find the so-called *upwind* schemes, which are based on the idea of following the signal propagation along the characteristic lines of the problem, and thus reconstructing the solution step by step in time, by studying how it is physically propagated over the whole domain. Resuming schemes 3.12 and 3.13, but also the other unbalanced ones previously derived, they are examples of upwind schemes, when applied to numerically approximate a differential problem by taking into account the direction of propagation of signals over the solution domain.

As far as it regards the extension to multi-dimensional problems, it is actually pretty straightforward to develop in case of regular and structured grids. The main drawback of finite difference methods indeed, despite of their simplicity, is the fact that they cannot be used on unstructured grids, since they need some sort of regularity in order to be properly defined all over the discretization points of the domain (see figure 3.3 for an example of unstructured grid). For problems characterised by unstructured grids, we address to the next subsection 3.1.2 for a brief introduction on the subject. By the way, as already mentioned, we are mainly concerned with finite differences since we will deal only

with regular and uniform cartesian grids, as will be shown more clearly within the next chapter 4. To conclude this first subsection on finite difference introduction, we provide a simple example of discretization of a 2D elliptic problem such as the one presented in section 2.1, that is the Poison's equation 2.3:

$$-\nabla^2\phi(x,y) = -\frac{\partial^2\phi}{\partial x^2} - \frac{\partial^2\phi}{\partial y^2} = g(x,y)$$

With boundary condition on a squared domain Ω of Dirichlet type, such as:

$$\phi(x,y) = \phi_D \quad \text{when} \quad (x,y) \in \partial\Omega$$

Then, by discretizing the second order spatial derivatives with the central scheme 3.6 everywhere in Ω other than on the boundary's points of $\partial\Omega$, for which special attention has to be paid, one obtains the following numerical scheme:

$$\begin{cases} -\dfrac{\phi_{i-1,j} - 2\phi_{i,j} + \phi_{i+1,j}}{\Delta x^2} - \dfrac{\phi_{i,j-1} - 2\phi_{i,j} + \phi_{i,j+1}}{\Delta y^2} = g_{i,j} \\ i \in (1, N_x - 1),\ j \in (1, N_y - 1) \end{cases} \tag{3.14}$$

Where i, j are the pair of indexes used to indicate each grid point, and which are therefore defined in a domain such as:

$$\Omega = \Delta x(0, N_x) \times \Delta y(0, N_y)$$

Supposing to have again a uniform equally spaced grid. It is now clear enough that from this scheme a linear system of equations arises, whose solution finally represents the spatial distribution of the function $\phi(x,y)$ over the whole domain Ω for this elliptic problem.

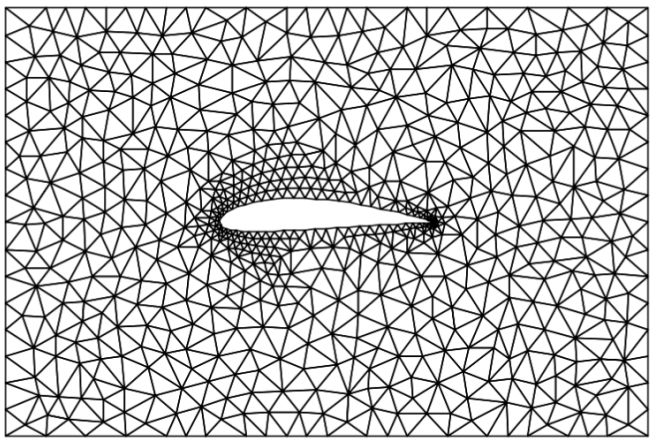


Figure 3.3: Example of 2D non-structured grid constituted by a triangulation surrounding an airfoil profile.

ENO & WENO Schemes:

Actually, a much more intriguing upwind scheme is adopted in order to devise approximations up to the fifth order of accuracy, based on what in the literature are commonly known as *ENO* or *WENO* schemes. About these innovative methods, a lot of references are available in the literature, and for what it follows we address primarily to the two works by the same author reported in [78] and [77], whereas, in order to give some examples of their implementation and application in a couple of research fields, consider the works [44], on their exploitation in the resolution of the so-called Hamilton-Jacobi equation, linked to some sort of problems that will be addressed also in the following chapters of this thesis, and [36], where they have been employed in the study of the biochemical process of proteins polymerization.

Essentially, they consist of a polynomial approximation characterized by a high level of accuracy (always at least greater than 2nd order of accuracy), useful in those situations where one needs to find the value of some variables on the interfaces between each pair of grid points, say, $\phi_{i+\frac{1}{2}}$, starting from their discrete distribution over all the grid nodes, ϕ_i. The main idea of *essentially non-oscillatory* schemes is that of exploiting a reconstruction of the interface value starting from its representation obtained through several lower-order polynomials, grouped together with respect to the result given by a higher-order one, built on a longer stencil which would imply, as a consequence, a higher regularity for the solution, a condition not always guaranteed due to the presence of discontinuities between adjacent cells. One thus looks for a representation such as:

$$\phi_{i+\frac{1}{2}} = \gamma_1 \phi^{p_1}_{i+\frac{1}{2}} + \gamma_2 \phi^{p_2}_{i+\frac{1}{2}} + \gamma_3 \phi^{p_3}_{i+\frac{1}{2}}$$

Where, if one considers as the largest stencil that composed by $[i-2, i-1, i, i+1, i+2]$, then the introduced quantities indicated with $\phi^{p_1}_{i+\frac{1}{2}}$, $\phi^{p_2}_{i+\frac{1}{2}}$ and $\phi^{p_3}_{i+\frac{1}{2}}$ are the lower-order approximations of the interface values of the solution obtained throughout the evaluation of the three respective polynomials composed with the three shorter stencils derived from the splitting of the largest one, that is:

$$\phi^{p_1}_{i+\frac{1}{2}} = p_1(x_{i+\frac{1}{2}}), \quad \text{on the stencil:} \quad [i-2, i-1, i]$$

$$\phi^{p_2}_{i+\frac{1}{2}} = p_2(x_{i+\frac{1}{2}}), \quad \text{on the stencil:} \quad [i-1, i, i+1]$$

$$\phi^{p_3}_{i+\frac{1}{2}} = p_3(x_{i+\frac{1}{2}}), \quad \text{on the stencil:} \quad [i, i+1, i+2]$$

While the parameters γ_1, γ_2 and γ_3 are those necessary to make that combination identical, term by term, to that associated with the higher order polynomial built on the longer stencil, thus:

$$\phi^{p_4}_{i+\frac{1}{2}} = p_4(x_{i+\frac{1}{2}}), \quad \text{on the stencil:} \quad [i-2, i-1, i, i+1, i+2]$$

Then, a *weighted essentially non-oscillatory* schemes modifies that structure by adopting specific weight functions ω_j (from now on j = 1,2,3) depending on both the values of γ_j

already known and some *smoothness indicator* β_j defined as:

$$\beta_j = \sum_{l=1}^{k} \Delta x^{2l-1} \int_{x_{i-\frac{1}{2}}}^{i+\frac{1}{2}} \left(\frac{\mathrm{d}^l p_j(x)}{\mathrm{d}x^l} \right)^2 dx$$

Where k is the polynomial degree of each p_j. This way, the presence of discontinuities will no longer represent an issue affecting the accuracy of the scheme, since the choice to assume non-linear weights ω_j instead of the classical linear ones γ_j connected to the ENO schemes ensures that only those polynomials built upon the stencil not containing such discontinuity give the most of the contribution, while according to the classical ENO methods, one has to select appropriately one of the three polynomial representation in order to avoid that discontinuity (using, for instance, some *limiter* function). Therefore, a WENO scheme is thought such that either $\omega_j \approx \gamma_j$, if the larger stencil does not contain discontinuities, or $\omega_j \approx 0$ if the polynomial p_j contains a discontinuity into his respective stencil. However, both these approaches lead to stable and non fluctuating solutions, as suggested by their own names. Those weight functions are then defined as:

$$\omega_j = \frac{\widetilde{\omega}_j}{\widetilde{\omega}_1 + \widetilde{\omega}_2 + \widetilde{\omega}_3}, \quad \text{and} \quad \widetilde{\omega}_j = \frac{\gamma_j}{(\varepsilon + \beta_j)^2}$$

Temporal Schemes:

As mentioned within the previous paragraph, the same schemes developed in order to approximate spatial derivative can be applied also in case of time derivatives, both of first and second order. When we consider, as a simple example, an ordinary differential equation such as the following one:

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = u'(t) = f(t, u(t)) \tag{3.15}$$

One can resort to several methods in order to numerically solve it, by properly discretizing the time derivative and thus obtaining a numerical solution which hopefully should be, depending on the adopted scheme, stable in time. Among the most widely adopted numerical schemes used with the aim of solving ODEs, one can find basically two families of them: *one step methods*, and *multi-steps* ones. The first case gathers some very popular schemes such as *explicit Euler*, *implicit Euler*, *Heun*, *trapezes* and the higher order *Runge-Kutta* schemes; within the second group one can find, for instance, *Adam-Bashforth* and *Adam-Moulton schemes*. In the following, we will introduce and show how to derive, in particular, the simplest Euler explicit and implicit schemes, the trapezes' (or *Crank-Nicolson's*) scheme and the two-steps Adam-Bashforth's.

Resuming the ordinary differential equation 3.15 introduced before, one could think to approximate its time derivative by adopting one of the numerical schemes presented in the previous paragraph, say, a forward finite difference scheme in time, considering a time step $\Delta t = t^{k+1} - t^k$, with $t \in (0, T)$, such that:

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} \approx \frac{u^{k+1} - u^k}{\Delta t} = f(t^k, u^k) = f^k$$

Thus, one obtains the so-called explicit Euler scheme, which is first order accurate in time due to the approximation adopted with respect to the time derivative:

$$u^{k+1} = u^k + \Delta t\, f(t^k, u^k) \tag{3.16}$$

Analogously, by adopting a backward finite difference scheme to discretize the time derivative in 3.15, thus evaluating the right-hand side term at the $(k+1)$-th time step, one obtains instead the implicit Euler scheme, which exhibits the advantage of being unconditionally stable with respect to the time step choice:

$$u^{k+1} = u^k + \Delta t\, f(t^{k+1}, u^{k+1}) \tag{3.17}$$

In order to derive the trapezes' scheme, it is necessary to proceed by integrating in time both the left and the right-hand side of 3.15, that is:

$$\int_{t^k}^{t^{k+1}} \frac{u^{k+1} - u^k}{\Delta t}\, dt = \int_{t^k}^{t^{k+1}} f(t^k, u^k)\, dt$$

And then, exploiting the trapezes' quadrature formula on the right-hand side term in order to approximate also that integral operator, one finally obtains the desired scheme:

$$u^{k+1} = u^k + \frac{\Delta t}{2}(f^k + f^{k+1}) \tag{3.18}$$

Which is, in addition, second order accurate in time. Just to recall them, but without explicitly focusing on the derivation of their structure, here below also the higher order Runge-Kutta schemes are presented in their essential formulation:

$$u^{k+1} = u^k + \Delta t \sum_{i=1}^{s} a_i k_i \tag{3.19}$$

With:

$$k_i = f\big(t^k + b_i \Delta t, u^k + \Delta t \sum_{j=1}^{i-1} c_{i,j} k_j\big)$$

Where s denotes the number of *stages* of the scheme, linked to its consequent order of accuracy, whereas the parameters a_i, b_i and $c_{i,j}$ are chosen depending on the particular scheme one would like to build (some combination of them can lead to the already mentioned Eulero's and Heun's scheme, among the others), according to the so-called *Butcher's tableau*, and keeping in mind that:

$$\sum_{i=1}^{s} a_i = 1 \quad \text{and} \quad b_i = \sum_{j=1}^{s} c_{i,j}, \forall i = 1, \ldots, s$$

To conclude, the two-steps Adam-Bashforth scheme is now obtained: similarly to what has been done with the Crank-Nicolson's scheme, here we again integrate both sides of 3.15 over the time step (t^k, t^{k+1}), coming to the same expression where the quantity

$\int_{t^k}^{t^{k+1}} f(t,u)\,dt$ appears as right-hand side. At this point, we introduce a first order polynomial approximation for $f(t,u)$, exploiting this time the pair (t^{k-1}, t^k), thus:

$$f \approx p(t) = f^k + \frac{f^k - f^{k-1}}{t^k - t^{k-1}}(t - t^k)$$

Considering that $p(t^k) = f^k$, $p(t^{k-1}) = f^{k-1}$ and $p(t^{k+1}) = 2f^k - f^{k-1}$, by substituting this new expression for $f(t,u)$ in the previous one, we have:

$$u^{k+1} = u^k + \int_{t^k}^{t^k+1} \left(f^k + \frac{f^k - f^{k-1}}{t^k - t^{k-1}}(t - t^k) \right) dt$$

And then, the final expression comes from solving the integral:

$$u^{k+1} = u^k + \Delta t \left(\frac{3}{2} f^k - \frac{1}{2} f^{k-1} \right) \tag{3.20}$$

3.1.2 Finite Volume and Finite Element Methods

Within this additional section we will address very briefly the Finite Volumes and the Finite Element methods as numerical tools used to discretize and solve approximately different kind of differential problems. These methods are not of our interest, since they will not be applied in this work, in particular, but a quick introduction is worth making, considering their importance in the current state of the art of the numerical community. More details on these approaches can be found quite easily in the literature, since a huge amount of books, works and publications are available nowadays. As a reminder, we refer to the works by Quarteroni in [66] and [67] for a deep and exhaustive analysis, or to the slides by professors Pieraccini and D'Ambrosio (PoliTo) concerning their classes on Numerical Methods and Scientific Computing, and Computational Fluid Dynamics, respectively, in [63] and [28].

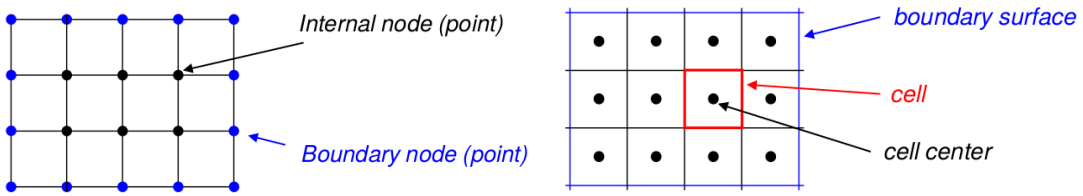


Figure 3.4: Examples of 2D grids, on the right a cell-centered Finite Volume-like grid (also vertex-centered grids are possible).

Finite Volume Method:

This approach, when applied to Fluid Dynamics problems, exploits the integral form of the conservation laws introduced in section 2.2. Taking as simple example a 2D linear advection equation written as:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \nabla \cdot \boldsymbol{F} = 0 \tag{3.21}$$

Then, by integrating over the whole domain Ω and by applying the divergence theorem to the second terms, one obtains the following conservative law for the previous expression (denoting with $\Sigma = \partial\Omega$ the external frontier of the domain):

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \boldsymbol{W}\, d\Omega + \int_{\Sigma} \boldsymbol{F}\cdot\boldsymbol{n}\, d\Sigma = 0 \tag{3.22}$$

The relation above is exactly the one which will be now discretized on the computational grid, by means of the Finite Volume Method. Let us consider a 2-dimensional grid such as the one depicted in figure 3.4, but which could also be not-regular and not-structured, then, for each cell, an integral conservation law such as 3.22 has to be written. Thus, by denoting with $\Omega_{i,j}$ the (i,j)-th cell's volume, and with $\Sigma_{i,j}$ its surface, one has that, for each cell of the mesh:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega_{i,j}} \boldsymbol{W}\, d\Omega_{i,j} + \int_{\Sigma_{i,j}} \boldsymbol{F}\cdot\boldsymbol{n}_{i,j}\, d\Sigma_{i,j} = 0$$

Now, those integrals are numerically approximated by a proper quadrature technique, such as rectangular or trapeze scheme, or more in general by referring to the *integral mean value theorem*. Calling $\boldsymbol{W}_{i,j}$ the value of the conservative variables corresponding to the cell center i,j, and denoting with indexes $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$ the respective cell's surfaces, whose associated fluxes would therefore be $\boldsymbol{F}_{i\pm\frac{1}{2},j}$ and $\boldsymbol{F}_{i,j\pm\frac{1}{2}}$, then we have:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{W}_{i,j}\Omega_{i,j} + (&\boldsymbol{F}_{i+\frac{1}{2},j}\cdot\boldsymbol{n}_{i+\frac{1}{2},j}\Sigma_{i+\frac{1}{2},j} + \boldsymbol{F}_{i-\frac{1}{2},j}\cdot\boldsymbol{n}_{i-\frac{1}{2},j}\Sigma_{i-\frac{1}{2},j} + \\ &+ \boldsymbol{F}_{i,j+\frac{1}{2}}\cdot\boldsymbol{n}_{i,j+\frac{1}{2}}\Sigma_{i,j+\frac{1}{2}} + \boldsymbol{F}_{i,j-\frac{1}{2}}\cdot\boldsymbol{n}_{i,j-\frac{1}{2}}\Sigma_{i,j-\frac{1}{2}}) = 0\end{aligned}$$

Which could also be written in the compact form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{W}_{i,j}\Omega_{i,j} + \sum_{f=1}^{4} \boldsymbol{F}_f\cdot\boldsymbol{n}_f\Sigma_f = 0 \tag{3.23}$$

What equation 3.23 represents is a set of ordinary differential equations for each cell of the mesh, and by adopting a discretization technique among those presented in the previous section 3.1.1, such as a forward 1st-order accurate finite difference scheme, we can finally approximate even the time derivative, thus coming to the following definitive scheme:

$$\boldsymbol{W}_{i,j}^{k+1} = \boldsymbol{W}_{i,j}^{k} - \frac{\Delta t}{\Omega_{i,j}}\sum_{f=1}^{4} \boldsymbol{F}_f^k\cdot\boldsymbol{n}_f\Sigma_f \tag{3.24}$$

Where with the superscript k has been indicated the current time evaluation of the corresponding quantity, that is $\boldsymbol{W}_{i,j}^{k} = \boldsymbol{W}_{i,j}(t_k)$, and thus with $k+1$ its subsequent one, having therefore defined a time step $\Delta t = t_{k+1} - t_k$. This is the main idea of Finite Volume schemes then: to allow for the resolution of a conservative approximated equation for each cell of the grid, where a proper way of writing integral quantities such as the mean value over the whole cell and on each of its interface for the fluxes have to be devised, and whose primary advantage is the extraordinary property of *conservativity*, due to the fact

that for adjacent cells, with a common interface, the flux $\boldsymbol{F}$ is imposed to be exactly the same, then:

$$\boldsymbol{F}_{i+\frac{1}{2}^-,j} \cdot \boldsymbol{n}_{i+\frac{1}{2}^-,j} = \boldsymbol{F}_{i+\frac{1}{2}^+,j} \cdot \boldsymbol{n}_{i+\frac{1}{2}^+,j}$$

Where $\boldsymbol{n}$ refers always to the outward normal vector with respect to each cell's surface, $i+\frac{1}{2}^-$ refers to the East surface of the i-th. cell, and $i+\frac{1}{2}^+$ to the West surface of the $(i+1)$-th cell.

Finite Element Method:

The following approach, as well as the previous one, lends itself pretty enthusiastically good to non-structured grid cases, and in addition provide a solid mathematical tool which allows for the performance evaluation of the implemented method, in terms of error estimates, control, etc. Let us consider as an example the following 2D parabolic problem:

$$\frac{\partial u}{\partial t} - \nu \nabla^2 u = f \tag{3.25}$$

Defined in the domain Ω, and whose boundary conditions, again, are of Dirichlet type (homogeneous for simplicity) on the whole frontier $\partial\Omega$, that is:

$$u = u_D = 0 \quad \text{on} \quad \partial\Omega$$

Equation 3.25, together with its boundary (and initial) condition, represent what is commonly known as the *strong formulation* of the problem, while Finite Element methods aim to derive a so-called *weak formulation*, or *variational*, which is an integral formulation of the same problem whose solution, intended in a weak sense, is the same as for the initial differential problem. The main advantage of considering such a procedure comes from the fact of allowing for more general problems to make sense, requiring a lower regularity for the solution u since we substantially low the order of derivation of our differential problems. Now, thanks to this aspect, even punctual and concentrate forcing terms could be considered, as well as piecewise defined ones, or piecewise definitions for the involved parameters. In order to derive the weak formulation of this parabolic problem, let us multiply equation 3.25 by a test function $v(x,y)$ and then let us integrate all the members over the domain Ω.

$$\int_\Omega \frac{\partial u}{\partial t} v \, d\Omega - \int_\Omega \nu \nabla^2 u \, v \, d\Omega = \int_\Omega f v \, d\Omega \tag{3.26}$$

The test function v must belong to the space of admissible functions $\mathbb{V}$, defined simply as:

$$\mathbb{V} = \mathbb{H}^1_0(\Omega) = \{v \in \mathbb{H}^1(\Omega) \to \mathbb{R} : v = 0 \text{ on } \partial\Omega\}$$

Where the space $\mathbb{H}^1$ is the so-called Hilbert's space (Sobolev's) of square integrable functions, that is:

$$\mathbb{H}^1(\Omega) = \{v \in \mathbb{L}^2(\Omega) : \nabla v \in \mathbb{L}^2(\Omega)\}$$

And, in turn, the space $\mathbb{L}^2$ of square integrable functions is defined as:

$$\mathbb{L}^2(\Omega) = \{v \in \Omega \to \mathbb{R} : \sqrt{\int_\Omega |v|^2 \, d\Omega} < +\infty\}$$

Thus, resuming equation 3.26 and proceeding with integration by parts for the laplacian term, and due to the fact that the test function v nullifies on the boundary of the domain, one gets the following weak formulation of the problem 3.25:

$$\int_{\Omega} \frac{\partial u}{\partial t} v \, d\Omega - \int_{\Omega} \nu \nabla u \nabla v \, d\Omega = \int_{\Omega} f v \, d\Omega \tag{3.27}$$

Introducing the discretization by considering a proper partitioning of the domain Ω_h, and a new subspace of discrete admissible test functions $\mathbb{V}_h$ whose basis could be chosen in a suitable manner (for instance, one could consider the lagrangian basis $\{\phi_i\}_i^N$), such that now all admissible discrete test functions $v_h \in \mathbb{V}_h$, as well as $u_h \in \mathbb{V}_h$, the new discretized solution in Ω_h. This approach therefore leads to the following discrete variational formulation of the problem 3.25:

$$\int_{\Omega_h} \frac{\partial u_h}{\partial t} v \, d\Omega_h - \int_{\Omega_h} \nu \nabla u_h \nabla v_h \, d\Omega_h = \int_{\Omega_h} f v_h \, d\Omega_h \tag{3.28}$$

Having considered for this example a discretization based on N degrees of freedom, one has now the following representation for the test functions v_h and for the discretized solution u_h:

$$v_h = \phi_j \quad \text{for} \quad j \in (1, N)$$

$$u_h = \sum_{i=1}^{N} u_i(t) \phi_i(\boldsymbol{x})$$

Thus, it follows that:

$$\sum_{i=1}^{N} \frac{\mathrm{d}u_i}{\mathrm{d}t} \int_{\Omega_h} \phi_i \phi_j \, d\Omega_h - \sum_{i=1}^{N} \int_{\Omega_h} \nu \nabla \phi_i \nabla \phi_j \, d\Omega_h = \int_{\Omega_h} f \phi_j \, d\Omega_h \quad j \in (1, N) \tag{3.29}$$

Which represents a linear system of ODEs, that is:

$$\boldsymbol{B}\dot{\boldsymbol{u}} + \boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$$

Where:

$$\boldsymbol{B} \in \mathbb{R}^{N \times N}, \quad \boldsymbol{A} \in \mathbb{R}^{N \times N} \quad \text{and} \quad \boldsymbol{f} \in \mathbb{R}^{N}$$

3.2 Adopted Spatial and Temporal Schemes

In this section we will finally present the chosen spatial and temporal schemes in order to discretize the governing equations for incompressible flows already presented in section 2.2. With respect to the 2-dimensional grid defined as computational domain of our problem, which consists basically of an extremely simple orthogonal cartesian grid, with uniform spacing both in horizontal and vertical direction, we will refer to each grid point by indicating its position with the pair of indexes (i, j), where:

$$i \in (0, N_x - 1) \quad \text{and} \quad j \in (0, N_y - 1)$$

Special care must be reserved to those grid points close to the domain boundary, and different schemes will be adopted in such cases. In the following, we will first focus on the discretization of the convective terms appearing in Navier-Stokes equations (refer to equations 2.26 in section 2.3), and then on the diffusive ones. For the moment, the issue regarding the pressure terms will be left apart, as it will be more deeply investigated in the next sections, and only a first idea of its discretization will be given. Subsequently, the temporal discretization will also be introduced, and finally the numerical treatment of the 0-dimensional Windkessel models will be addressed, as well as their coupling with the 2-dimensional domain.

3.2.1 Convective and Pressure terms

Convective terms:

First, the convective terms appearing in the 2D incompressible Navier-Stokes equations 2.26 are here recalled:

$$\boldsymbol{V} \cdot \nabla \boldsymbol{u} = u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y}$$

$$\boldsymbol{V} \cdot \nabla \boldsymbol{v} = u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y}$$

respectively associated with the x and y-wise component of the momentum balance vectorial equation. Depending on the position over the grid nodes, a discretization of the following type will be introduced:

$$\boldsymbol{V} \cdot \nabla \boldsymbol{u} \approx u_{i,j}\left(\frac{\delta u}{\delta x}\right)_{i,j} + v_{i,j}\left(\frac{\delta u}{\delta y}\right)_{i,j}$$

$$\boldsymbol{V} \cdot \nabla \boldsymbol{v} \approx u_{i,j}\left(\frac{\delta v}{\delta x}\right)_{i,j} + v_{i,j}\left(\frac{\delta v}{\delta y}\right)_{i,j}$$

Where with $u_{i,j} = u(x_i, y_j, t) = u(x_0 + i\Delta x, y_0 + j\Delta y, t)$ has been indicated the i,j-th grid point's horizontal velocity component, as well as for $v_{i,j}$, whereas with the symbols $\left(\frac{\delta}{\delta x}\right)_{i,j}$ and $\left(\frac{\delta}{\delta y}\right)_{i,j}$ have been denoted some proper discretization for the velocity gradients still corresponding to the i,j-th grid point. With regard to these quantities, in order to introduce the selected discretization concerning each grid point, it is first necessary to distinguish their position with respect to the external boundaries:

- $i = 0$, $j = 0$, $i = Nx - 1$ or $j = Ny - 2$: points belonging to the boundaries, for which a boundary condition is applied on both the velocity components u and v, refer to section 3.3.3;
- $i = 1$ or $i = N_x - 1$ (with $j \neq 0$ and $j \neq N_y - 1$): points immediately next to the left and right boundaries, a central scheme (relation 3.3) is adopted for the determination of the velocity gradients in the x direction, that is:

$$\left(\frac{\delta u}{\delta x}\right)_{1,j} = \frac{u_{2,j} - u_{0,j}}{2\Delta x} \quad \text{and} \quad \left(\frac{\delta u}{\delta x}\right)_{N_x-2,j} = \frac{u_{N_x-1,j} - u_{N_x-3,j}}{2\Delta x}$$

And the same holds for the y-wise velocity component:

$$\left(\frac{\delta v}{\delta x}\right)_{1,j} = \frac{v_{2,j} - v_{0,j}}{2\Delta x} \quad \text{and} \quad \left(\frac{\delta v}{\delta x}\right)_{N_x-2,j} = \frac{v_{N_x-1,j} - v_{N_x-3,j}}{2\Delta x}$$

- $j = 1$ or $j = N_y - 1$ (with $i \neq 0$ and $i \neq N_x - 1$): as for the previous case, it involves the points next to the lower and to the upper boundaries, where again a central scheme is employed for both the velocity components in the y direction, thus:

$$\left(\frac{\delta u}{\delta y}\right)_{i,1} = \frac{u_{i,2} - u_{i,0}}{2\Delta y} \quad \text{and} \quad \left(\frac{\delta u}{\delta y}\right)_{i,N_y-2} = \frac{u_{i,N_y-1} - u_{i,N_y-3}}{2\Delta y}$$

$$\left(\frac{\delta v}{\delta y}\right)_{i,1} = \frac{v_{i,2} - v_{i,0}}{2\Delta y} \quad \text{and} \quad \left(\frac{\delta v}{\delta y}\right)_{i,N_y-2} = \frac{v_{i,N_y-1} - v_{i,N_y-3}}{2\Delta y}$$

- $1 < i < N_x - 1$ (with $j \neq 0$ and $j \neq N_y - 1$): an upwind 4-points scheme (relations 3.12 and 3.13) is applied to all the internal grid nodes, which is third order accurate and stable with respect to signals' propagation direction, depending on the sign of the local x-wise velocity component, that is:

$$\left(\frac{\delta u}{\delta x}\right)_{i,j} = \frac{2u_{i+1,j} + 3u_{i,j} - 6u_{i-1,j} + u_{i-2,j}}{6\Delta x}, \quad \text{if} \quad u_{i,j} > 0$$

$$\left(\frac{\delta u}{\delta x}\right)_{i,j} = \frac{-u_{i+2,j} + 6u_{i+1,j} - 3u_{i,j} - 2u_{i-1,j}}{6\Delta x}, \quad \text{if} \quad u_{i,j} < 0$$

The same holds for the y-wise component of the velocity v:

$$\left(\frac{\delta v}{\delta x}\right)_{i,j} = \frac{2v_{i+1,j} + 3v_{i,j} - 6v_{i-1,j} + v_{i-2,j}}{6\Delta x}, \quad \text{if} \quad u_{i,j} > 0$$

$$\left(\frac{\delta v}{\delta x}\right)_{i,j} = \frac{-v_{i+2,j} + 6v_{i+1,j} - 3v_{i,j} - 2v_{i-1,j}}{6\Delta x}, \quad \text{if} \quad u_{i,j} < 0$$

- $1 < j < N_y - 1$ (with $i \neq 0$ and $i \neq N_x - 1$): analogously along the vertical direction, where the proper upwind scheme is selected depending on the sign of the local y-wise velocity component:

$$\left(\frac{\delta u}{\delta y}\right)_{i,j} = \frac{2u_{i,j+1} + 3u_{i,j} - 6u_{i,j-1} + u_{i,j-2}}{6\Delta y}, \quad \text{if} \quad v_{i,j} > 0$$

$$\left(\frac{\delta u}{\delta y}\right)_{i,j} = \frac{-u_{i,j+2} + 6u_{i,j+1} - 3u_{i,j} - 2u_{i,j-1}}{6\Delta y}, \quad \text{if} \quad v_{i,j} < 0$$

The same holds for the y-wise component of the velocity v:

$$\left(\frac{\delta v}{\delta y}\right)_{i,j} = \frac{2v_{i,j+1} + 3v_{i,j} - 6v_{i,j-1} + v_{i,j-2}}{6\Delta y}, \quad \text{if} \quad v_{i,j} > 0$$

$$\left(\frac{\delta v}{\delta y}\right)_{i,j} = \frac{-v_{i,j+2} + 6v_{i,j+1} - 3v_{i,j} - 2v_{i,j-1}}{6\Delta y}, \quad \text{if} \quad v_{i,j} < 0$$

$$\left(\frac{\delta v}{\delta y}\right)_{i,j} = \frac{-v_{i,j+2} + 6v_{i,j+1} - 3v_{i,j} - 2v_{i,j-1}}{6\Delta y}, \quad \text{if} \quad v_{i,j} < 0$$

Furthermore, in order to reach up to the fifth order of accuracy in the determination of these velocity gradients, WENO schemes could also be adopted for all points inside the domain, that is, in case $i \in (3, N_x - 4)$ and $j \in (3, N_y - 4)$. In this case, one looks at a stencil composed of 6 grid points, depending again on the sign of the local velocity component corresponding to the derivation direction:

$$[i-3, i-2, i-1, i, i+1, i+2] \quad \text{if} \quad u_{i,j} > 0$$

$$[i-2, i-1, i, i+1, i+2, i+3] \quad \text{if} \quad u_{i,j} < 0$$

$$[j-3, j-2, j-1, j, j+1, j+2] \quad \text{if} \quad v_{i,j} > 0$$

$$[j-2, j-1, j, j+1, j+2, j+3] \quad \text{if} \quad v_{i,j} < 0$$

Then, one can now compute all the spatial derivatives on the interfaces between each pair of points belonging to those stencils, employing the centered finite difference scheme in such a way that (with $k = 1, \ldots, 5$):

$$\left(\frac{\partial u}{\partial x}\right)_{i+k-3-\frac{1}{2},j} = \nu_k = \frac{u_{i+k-3,j} - u_{i+k-4,j}}{\Delta x}, \quad \text{if} \quad u_{i,j} > 0$$

$$\left(\frac{\partial u}{\partial x}\right)_{i+k-3-\frac{1}{2},j} = \nu_k = \frac{u_{i+k-2,j} - u_{i+k-3,j}}{\Delta x}, \quad \text{if} \quad u_{i,j} < 0$$

$$\left(\frac{\partial u}{\partial y}\right)_{i,j+k-3-\frac{1}{2}} = \nu_k = \frac{u_{i,j+k-3} - u_{i,j+k-4}}{\Delta y}, \quad \text{if} \quad v_{i,j} > 0$$

$$\left(\frac{\partial u}{\partial y}\right)_{i,j+k-3-\frac{1}{2}} = \nu_k = \frac{u_{i,j+k-2} - u_{i,j+k-3}}{\Delta y}, \quad \text{if} \quad v_{i,j} < 0$$

And the same holds, once again, also for the y-wise velocity component v (always with $k = 1, \ldots, 5$):

$$\left(\frac{\partial v}{\partial x}\right)_{i+k-3-\frac{1}{2},j} = \nu_k = \frac{v_{i+k-3,j} - v_{i+k-4,j}}{\Delta x}, \quad \text{if} \quad u_{i,j} > 0$$

$$\left(\frac{\partial v}{\partial x}\right)_{i+k-3-\frac{1}{2},j} = \nu_k = \frac{v_{i+k-2,j} - v_{i+k-3,j}}{\Delta x}, \quad \text{if} \quad u_{i,j} < 0$$

$$\left(\frac{\partial v}{\partial y}\right)_{i,j+k-3-\frac{1}{2}} = \nu_k = \frac{v_{i,j+k-3} - v_{i,j+k-4}}{\Delta y}, \quad \text{if} \quad v_{i,j} > 0$$

$$\left(\frac{\partial v}{\partial y}\right)_{i,j+k-3-\frac{1}{2}} = \nu_k = \frac{v_{i,j+k-2} - v_{i,j+k-3}}{\Delta y}, \quad \text{if} \quad v_{i,j} < 0$$

Now, since with the index k it has been denoted a sequential ratio corresponding to an interface laying between two grid points, the application of a WENO reconstruction based on these velocity gradients can lead to the desired approximation of the spatial derivatives $\left(\frac{\delta u}{\delta x}\right)_{i,j}$, $\left(\frac{\delta v}{\delta x}\right)_{i,j}$, $\left(\frac{\delta v}{\delta x}\right)_{i,j}$ and $\left(\frac{\delta v}{\delta y}\right)_{i,j}$ as belonging to the interface between the points $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$, corresponding to specific values of k. Thus, following the procedure presented in the previous section 3.1 and what is described in works [77, 44], one has that:

$$\beta_1 = \frac{13}{12}(\nu_1 - 2\nu_2 + \nu_3)^2 + \frac{1}{4}(\nu_1 - 4\nu_2 + 3\nu_3)^2$$

$$\beta_2 = \frac{13}{12}(\nu_2 - 2\nu_3 + \nu_4)^2 + \frac{1}{2}(\nu_2 - \nu_4)^2$$

$$\beta_3 = \frac{13}{12}(\nu_3 - 2\nu_4 + \nu_5)^2 + \frac{1}{4}(3\nu_3 - 4\nu_4 + \nu_5)^2$$

In addition, following the procedure explained in section 3.1 regarding the computation of linear weights, one obtains:

$$\gamma_1 = \frac{3}{10}, \quad \gamma_2 = \frac{6}{10} \quad \text{and} \quad \gamma_2 = \frac{1}{10}$$

So that, we can now define:

$$\omega_j = \frac{\widetilde{\omega}_j}{\widetilde{\omega}_1 + \widetilde{\omega}_2 + \widetilde{\omega}_3}, \quad \text{and} \quad \widetilde{\omega}_j = \frac{\gamma_j}{(\varepsilon + \beta_j)^2}, \quad j = 1,2,3$$

And, finally, the desired scheme for the generic spatial gradient:

$$\frac{\delta(.)}{\delta x} = \omega_1\left(\frac{\nu_1}{3} - \frac{7\nu_2}{6} + \frac{11\nu_3}{6}\right) + \omega_2\left(-\frac{\nu_2}{6} + \frac{5\nu_3}{6} + \frac{\nu_4}{3}\right) + \omega_3\left(\frac{\nu_3}{3} + \frac{5\nu_4}{6} - \frac{\nu_5}{6}\right) \quad (3.30)$$

$$\frac{\delta(.)}{\delta y} = \omega_1\left(\frac{\nu_1}{3} - \frac{7\nu_2}{6} + \frac{11\nu_3}{6}\right) + \omega_2\left(-\frac{\nu_2}{6} + \frac{5\nu_3}{6} + \frac{\nu_4}{3}\right) + \omega_3\left(\frac{\nu_3}{3} + \frac{5\nu_4}{6} - \frac{\nu_5}{6}\right) \quad (3.31)$$

Pressure terms:

As mentioned, only a firs idea of pressure treatment will be given in this section, as it will much more deeply discussed in the following one regarding the global numerical method usually involved with incompressible flows, which gives to the pressure a special role in the play (see for instance [65, p.1-6]). The pressure contribute appears into equations 2.26 through its spatial gradient, and for this reason its absolute value is not that important for the solution of the problem, since only spatial pressure differences are taken into account. Actually, as will be specified and largely explained later, the pressure is no longer a thermodynamic variable when incompressibility occurs, indeed it serves only as a constraint for the velocity field, in order to keep it incompressible, enforcing somehow the condition represented by the continuity equation 2.24 over the whole domain, at each instant of time (see also section 2.3.3). Therefore, in order to write a first discretization of these contributions, one could simply resort to centered difference schemes 3.3 as follows:

$$\frac{\partial p}{\partial x} \approx \left(\frac{\delta p}{\delta x}\right)_{i,j} = \frac{p_{i+1,j} - p_{i-1,j}}{2\Delta x} \quad (3.32)$$

$$\frac{\partial p}{\partial y} \approx \left(\frac{\delta p}{\delta y}\right)_{i,j} = \frac{p_{i,j+1} - p_{i,j-1}}{2\Delta y} \tag{3.33}$$

Where, obviously, only internal nodes will be considered, thus the ones with $i \in (1, Nx-2)$ and $j \in (1, Ny-2)$, since for those belonging to boundaries proper conditions will be imposed (section 3.3.3).

3.2.2 Diffusive terms

Concerning the dissipative (or diffusive) terms appearing in the viscous formulation of the governing equations, those showed in the incompressible Navier-Stokes equations 2.26, we recall them here in order to introduce their discretization strategy:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$$

$$\nabla^2 v = \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}$$

We therefore aim to look for a numerical representation such as:

$$\nabla^2 u \approx \left(\frac{\delta^2 u}{\delta x^2}\right)_{i,j} + \left(\frac{\delta^2 u}{\delta y^2}\right)_{i,j}$$

$$\nabla^2 v \approx \left(\frac{\delta^2 v}{\delta x^2}\right)_{i,j} + \left(\frac{\delta^2 v}{\delta y^2}\right)_{i,j}$$

Where, again, all the approximated derivatives $\left(\frac{\delta^2(.)}{\delta x^2}\right)_{i,j}$ and $\left(\frac{\delta^2(.)}{\delta y^2}\right)_{i,j}$ are evaluated in the (i,j)-th grid point, and thus on the (i,j)-th cell's center of an equivalent cell's center mesh representation, as that depicted in figure 3.5. Considering the laplacian operator involved in the incompressible formulation of these equations, one could resort simply to the centered scheme 3.6 devised in order to approximate second order derivatives, thus writing expressions such as:

$$\left(\frac{\delta^2 u}{\delta x^2}\right)_{i,j} + \left(\frac{\delta^2 u}{\delta y^2}\right)_{i,j} = \frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2}$$

$$\left(\frac{\delta^2 v}{\delta x^2}\right)_{i,j} + \left(\frac{\delta^2 v}{\delta y^2}\right)_{i,j} = \frac{v_{i-1,j} - 2v_{i,j} + v_{i+1,j}}{\Delta x^2} + \frac{v_{i,j-1} - 2v_{i,j} + v_{i,j+1}}{\Delta y^2}$$

Conversely, due to stability reasons, it could be more convenient taking into account the interface values of the velocity components surrounding each (i,j)-th cell, referring in addition to the more general formulation of the governing equations 2.20, where the divergence of the viscous stress tensor appears, and where the incompressibility condition (i.e. the continuity equation 2.24) has not been applied yet. Thus, by rewriting those diffusive contributions as:

$$\nabla \cdot \boldsymbol{\tau}_x = \frac{\partial \tau_{xx}}{\partial x} + \frac{\partial \tau_{xy}}{\partial y} = \approx \left(\frac{\delta \tau_{xx}}{\delta x}\right)_{i,j} + \left(\frac{\delta \tau_{xy}}{\delta y}\right)_{i,j}$$

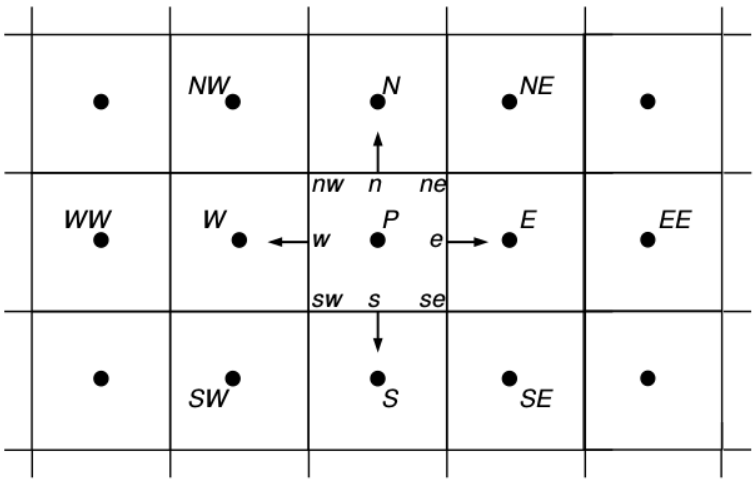


Figure 3.5: Scheme of the numerical grid with emphasis on cell's interfaces and their notation.

$$\nabla \cdot \boldsymbol{\tau}_y = \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \tau_{yy}}{\partial y} =\approx \left(\frac{\delta \tau_{xy}}{\delta x}\right)_{i,j} + \left(\frac{\delta \tau_{yy}}{\delta y}\right)_{i,j}$$

They could be discretized as follows, by employing a centered scheme (relation 3.3) which is consistent with the fundamental isotropic behaviour of these diffusive effects:

$$\left(\frac{\delta \tau_{xx}}{\delta x}\right)_{i,j} + \left(\frac{\delta \tau_{xy}}{\delta y}\right)_{i,j} = \frac{(\tau_{xx})_e - (\tau_{xx})_w}{\Delta x} + \frac{(\tau_{xy})_n - (\tau_{xy})_s}{\Delta y} \tag{3.34}$$

$$\left(\frac{\delta \tau_{xy}}{\delta x}\right)_{i,j} + \left(\frac{\delta \tau_{yy}}{\delta y}\right)_{i,j} = \frac{(\tau_{xy})_e - (\tau_{xy})_w}{\Delta x} + \frac{(\tau_{yy})_n - (\tau_{yy})_s}{\Delta y} \tag{3.35}$$

Obviously, these schemes will regard only the internal points of the computational domain, therefore those which satisfy $i \in (1, N_x - 2)$ and $j \in (1, N_y - 2)$. To clarify the adopted notation, referring to the scheme showed in figure 3.5, with letters n, s, e, w have been indicated the four interfaces of the (i, j)-th cell, thus corresponding to the grid points $i \pm \frac{1}{2}$ and $j \pm \frac{1}{2}$, following the idea of a geographic cardinal points frame. A proper strategy to approximate the components of the shear stress tensor has also to be devised at this point. With regard to this, and recalling in addition the general definition 2.15 of those terms (under the assumption of incompressibility, such that $\sum_{k=1}^{3} \frac{\partial u_k}{\partial x_k} = 0$), which reads:

$$\tau_{xx} = 2\mu \frac{\partial u}{\partial x}, \quad \tau_{xy} = \mu\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right) \quad \text{and} \quad \tau_{yy} = 2\mu \frac{\partial v}{\partial y}$$

Therefore, it is now possible to associate each interface shear stress which appears in the relations above to the corresponding velocity gradient properly computed on those intermediate points, i.e. $\left(\frac{\partial(.)}{\partial x}\right)_e$, $\left(\frac{\partial(.)}{\partial x}\right)_w$, $\left(\frac{\partial(.)}{\partial x}\right)_n$, $\left(\frac{\partial(.)}{\partial x}\right)_s$, and so on. The components of the stress tensor are thus written as follows:

$$(\tau_{xx})_e = 2\mu\left(\frac{\partial u}{\partial x}\right)_e, \quad (\tau_{xy})_e = \mu\left(\left(\frac{\partial u}{\partial y}\right)_e + \left(\frac{\partial v}{\partial x}\right)_e\right), \quad (\tau_{yy})_e = 2\mu\left(\frac{\partial v}{\partial y}\right)_e$$

$$(\tau_{xx})_w = 2\mu\left(\frac{\partial u}{\partial x}\right)_w, \quad (\tau_{xy})_w = \mu\left(\left(\frac{\partial u}{\partial y}\right)_w + \left(\frac{\partial v}{\partial x}\right)_w\right), \quad (\tau_{yy})_w = 2\mu\left(\frac{\partial v}{\partial y}\right)_w$$

$$(\tau_{xx})_n = 2\mu\left(\frac{\partial u}{\partial x}\right)_n, \quad (\tau_{xy})_n = \mu\left(\left(\frac{\partial u}{\partial y}\right)_n + \left(\frac{\partial v}{\partial x}\right)_n\right), \quad (\tau_{yy})_n = 2\mu\left(\frac{\partial v}{\partial y}\right)_n$$

$$(\tau_{xx})_s = 2\mu\left(\frac{\partial u}{\partial x}\right)_s, \quad (\tau_{xy})_s = \mu\left(\left(\frac{\partial u}{\partial y}\right)_s + \left(\frac{\partial v}{\partial x}\right)_s\right), \quad (\tau_{yy})_s = 2\mu\left(\frac{\partial v}{\partial y}\right)_s$$

Then, we introduce the following discretization technique for the latter, exploiting the interface grid points cited above and denoted again through the corresponding geographical cardinal points:

$$\left(\frac{\partial u}{\partial x}\right)_e = \frac{u_E - u_P}{\Delta x} = \frac{u_{i+1,j} - u_{i,j}}{\Delta x}, \quad \left(\frac{\partial u}{\partial x}\right)_w = \frac{u_P - u_E}{\Delta x} = \frac{u_{i,j} - u_{i-1,j}}{\Delta x}$$

$$\left(\frac{\partial u}{\partial x}\right)_n = \frac{1}{2\Delta x}\left(\frac{u_{NE} + u_E}{2} - \frac{u_{NW} + u_W}{2}\right) = \frac{1}{2\Delta x}\left(\frac{u_{i+1,j+1} + u_{i+1,j}}{2} - \frac{u_{i-1,j+1} + u_{i-1,j}}{2}\right)$$

$$\left(\frac{\partial u}{\partial x}\right)_s = \frac{1}{2\Delta x}\left(\frac{u_{SE} + u_E}{2} - \frac{u_{SW} + u_W}{2}\right) = \frac{1}{2\Delta x}\left(\frac{u_{i+1,j-1} + u_{i+1,j}}{2} - \frac{u_{i-1,j-1} + u_{i-1,j}}{2}\right)$$

$$\left(\frac{\partial u}{\partial y}\right)_e = \frac{1}{2\Delta y}\left(\frac{u_{NE} + u_N}{2} - \frac{u_{SE} + u_S}{2}\right) = \frac{1}{2\Delta y}\left(\frac{u_{i+1,j+1} + u_{i,j+1}}{2} - \frac{u_{i+1,j-1} + u_{i,j-1}}{2}\right)$$

$$\left(\frac{\partial u}{\partial y}\right)_w = \frac{1}{2\Delta y}\left(\frac{u_{NW} + u_N}{2} - \frac{u_{SW} + u_S}{2}\right) = \frac{1}{2\Delta y}\left(\frac{u_{i-1,j+1} + u_{i,j+1}}{2} - \frac{u_{i-1,j-1} + u_{i,j-1}}{2}\right)$$

$$\left(\frac{\partial u}{\partial y}\right)_n = \frac{u_N - u_P}{\Delta y} = \frac{u_{i,j+1} - u_{i,j}}{\Delta y}, \quad \left(\frac{\partial u}{\partial y}\right)_s = \frac{u_P - u_S}{\Delta x} = \frac{u_{i,j} - u_{i,j-1}}{\Delta y}$$

$$\left(\frac{\partial v}{\partial x}\right)_e = \frac{v_E - v_P}{\Delta x} = \frac{v_{i+1,j} - v_{i,j}}{\Delta x}, \quad \left(\frac{\partial v}{\partial x}\right)_w = \frac{v_P - v_E}{\Delta x} = \frac{v_{i,j} - v_{i-1,j}}{\Delta x}$$

$$\left(\frac{\partial v}{\partial x}\right)_n = \frac{1}{2\Delta x}\left(\frac{v_{NE} + v_E}{2} - \frac{v_{NW} + u_W}{2}\right) = \frac{1}{2\Delta x}\left(\frac{v_{i+1,j+1} + v_{i+1,j}}{2} - \frac{v_{i-1,j+1} + v_{i-1,j}}{2}\right)$$

$$\left(\frac{\partial v}{\partial x}\right)_s = \frac{1}{2\Delta x}\left(\frac{v_{SE} + v_E}{2} - \frac{v_{SW} + v_W}{2}\right) = \frac{1}{2\Delta x}\left(\frac{v_{i+1,j-1} + v_{i+1,j}}{2} - \frac{v_{i-1,j-1} + v_{i-1,j}}{2}\right)$$

$$\left(\frac{\partial v}{\partial y}\right)_e = \frac{1}{2\Delta y}\left(\frac{v_{NE} + v_N}{2} - \frac{v_{SE} + v_S}{2}\right) = \frac{1}{2\Delta y}\left(\frac{v_{i+1,j+1} + v_{i,j+1}}{2} - \frac{v_{i+1,j-1} + v_{i,j-1}}{2}\right)$$

$$\left(\frac{\partial v}{\partial y}\right)_w = \frac{1}{2\Delta y}\left(\frac{v_{NW} + v_N}{2} - \frac{v_{SW} + v_S}{2}\right) = \frac{1}{2\Delta y}\left(\frac{v_{i-1,j+1} + v_{i,j+1}}{2} - \frac{v_{i-1,j-1} + v_{i,j-1}}{2}\right)$$

$$\left(\frac{\partial v}{\partial y}\right)_n = \frac{v_N - v_P}{\Delta y} = \frac{v_{i,j+1} - v_{i,j}}{\Delta y}, \quad \left(\frac{\partial v}{\partial y}\right)_s = \frac{v_P - v_S}{\Delta x} = \frac{v_{i,j} - v_{i,j-1}}{\Delta y}$$

To conclude this section, by developing expressions 3.34 and 3.35 applying the appropriate schemes reported in an extremely clear manner, one can finally write the discretized form of the viscous terms included in the Navier-Stokes equations adopted as the mathematical model describing our problem.

3.2.3 Temporal Schemes and Multi-D Models

In this section both temporal schemes concerning the evolutionary behaviour of governing equations and those associated with 0-dimensional models employed as coupled boundary conditions (see section 2.5) will be presented, since in both cases we are supposed to deal with time dependent ODEs or a system of such equations.

Time Discretization:

Starting from the time discretization chosen to compute the temporal evolution of the solution, a simple and easy-coding explicit Euler scheme (relation 3.16) is employed in order to predict the new velocity field at each time step. Note that the expression *prediction* is a key-word in this sense, as will be better clarified in the following, once having introduced the Fractional Step Method in section 3.3 appositely dedicated, which consists of the actual time handling of the equations. For the moment, let us suppose to deal with a time-space discrete scheme which looks basically like the following set of expressions:

$$\begin{cases} \frac{u_{i,j}^{k+1}-u_{i,j}^{k}}{\Delta t} = -u_{i,j}^{k}\left(\frac{\delta u}{\delta x}\right)_{i,j}^{k} - v_{i,j}^{k}\left(\frac{\delta u}{\delta y}\right)_{i,j}^{k} + \frac{1}{\rho}\left(\frac{\delta p}{\delta x}\right)_{i,j}^{k} + \nu\left(\left(\frac{\delta \tau_{xx}}{\delta x}\right)_{i,j}^{k} + \left(\frac{\delta \tau_{xy}}{\delta v}\right)_{i,j}^{k}\right) \\ \frac{v_{i,j}^{k+1}-v_{i,j}^{k}}{\Delta t} = -u_{i,j}^{k}\left(\frac{\delta v}{\delta x}\right)_{i,j}^{k} - v_{i,j}^{k}\left(\frac{\delta v}{\delta y}\right)_{i,j}^{k} + \frac{1}{\rho}\left(\frac{\delta p}{\delta y}\right)_{i,j}^{k} + \nu\left(\left(\frac{\delta \tau_{xy}}{\delta x}\right)_{i,j}^{k} + \left(\frac{\delta \tau_{yy}}{\delta v}\right)_{i,j}^{k}\right) \end{cases} \tag{3.36}$$

Where, obviously, the kinematic viscosity $\nu = \frac{\mu}{\rho}$ has been introduced. This scheme applies to all grid points $(i, j) \in \Omega = (0, N_x - 1) \times (0, N_y - 1)$, and to all the temporal time steps $k \in (0, N_t - 1)$, being $T = (N_t - 1)\Delta t$, starting from the initial data previously assigned in order to mathematically pose the problem (section 2.3.2).

Finally, a couple of discretization approaches are given also in order to approximate and couple 0-dimensional models such as Windkessel's with the multi-dimensional one (hereafter multi-D). First, it is worth noticing that many different models, and thus many different schemes are possible to be devised in this context. In section 1.3.4 some examples are given, and it is fairly easy to understand that one could reach any level of detail simply by adding new elements to his models, tuning them and consequently coupling them to his multi-D ones. The fundamentals of Winkessel models have already been widely presented in section 2.5, and here we will resume the final expression 2.41 derived from the 3-elements Windkessel model, reported below (refer to figure 2.13 for element's notation):

$$P_{wk3}(t) = Q(t)(R_a + Z_c) - C_a R_a \frac{\mathrm{d}P_{wk3}(t)}{\mathrm{d}t} + C_a R_a Z_c \frac{\mathrm{d}Q(t)}{\mathrm{d}t}$$

As already stated, it consists of a simple 1st order ODE in time, which could be properly discretized by adopting one of the several schemes proposed in section 3.1.1. Here, we will derive a 1st semi-implicit Euler scheme, and we will also provide a 4th stages Runge-Kutta one, even if its results, as will be shown in the dedicated chapter, are exactly the same as those obtained with the former, thus leading to that more straightforward choice.

- **Semi-implicit Euler:** by assuming the same notation as before, for the sake of clarity, and by denoting the following averaged quantities related to the multi-D

domain as:

$$P_{wk3}(t) = \overline{p_{out}}(t) = \frac{1}{L_y}\int_{\Gamma_{out}} p(x = L_x, y, t)\, dy$$

$$Q(t) = \overline{Q_{out}}(t) = \int_{\Gamma_{out}} u(x = L_x, y, t)\, dy$$

Then, one can proceed with the following time approximation:

$$P_{wk3}^{k+1} = Q^{k+1}(R_a + Z_c) - C_a R_a \frac{P_{wk3}^{k+1} - P_{wk3}^{k}}{\Delta t} + C_a R_a Z_c \frac{Q^{k+1} - Q^{k}}{\Delta t}$$

And finally, by evidencing the outflow mean pressure $P_{wk3}(t)$:

$$P_{wk3}^{k+1} = \frac{Q^{k+1}(R_a + Z_c + \frac{C_a R_i R_a}{\Delta t}) + \frac{C_a R_a}{\Delta t} P_{wk3}^{k} - \frac{C_a R_a Z_c}{\Delta t} Q^{k}}{1 + \frac{C_a R_a}{\Delta t}} \tag{3.37}$$

This relation, linking the outflowing mass rate and the respective mean pressure, will be applied as boundary condition on the outlet boundary of the domain Γ_{out}, in a fashion deeply discussed in the following.

- **RK-4:** what it follows is directly taken from [6], so that we resort to a scheme such as (assuming that $\frac{\mathrm{d}\phi}{\mathrm{d}t} = f(t, \phi)$):

$$\begin{aligned}
k_1 &= \Delta t f(t^k, \phi^k) \\
k_2 &= \Delta t f(t^k + \frac{\Delta t}{2}, \phi^k + \frac{k_1}{2}) \\
k_3 &= \Delta t f(t^k + \frac{\Delta t}{2}, \phi^k + \frac{k_2}{2}) \\
k_4 &= \Delta t f(t^k + \Delta t, \phi^k + \frac{k_3}{2}) \\
\phi^{k+1} &= \phi^k + \frac{k_1}{6} + \frac{k_2}{3} + \frac{k_3}{3} + \frac{k_4}{6} + \mathcal{O}(\Delta t^5)
\end{aligned}$$

Then, by employing the latter at the time depending ODE 2.41 of our interest, one obtains the following algorithm:

$$\begin{aligned}
k_1 &= \Delta t\left(Q^{k+1}\frac{R_a + Z_c}{C_a R_a} + Z_c\frac{Q^{k+1} - Q^k}{\Delta t} - P_{wk3}^k\frac{1}{C_a R_a}\right) \\
k_2 &= \Delta t\left(Q^{k+1}\frac{R_a + Z_c}{C_a R_a} + Z_c\frac{Q^{k+1} - Q^k}{\Delta t} - P_{wk3}^k\frac{1}{C_a R_a} + \frac{k_1}{2}\right) \\
k_3 &= \Delta t\left(Q^{k+1}\frac{R_a + Z_c}{C_a R_a} + Z_c\frac{Q^{k+1} - Q^k}{\Delta t} - P_{wk3}^k\frac{1}{C_a R_a} + \frac{k_2}{2}\right) \\
k_4 &= \Delta t\left(Q^{k+1}\frac{R_a + Z_c}{C_a R_a} + Z_c\frac{Q^{k+1} - Q^k}{\Delta t} - P_{wk3}^k\frac{1}{C_a R_a} + k_3\right) \\
P_{wk3}^{k+1} &= P_{wk3}^k + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned} \tag{3.38}$$

Multi-D Models Discretization and Coupling:

As it regards the coupling between the 0-dimensional and the multi-D models, again several solutions are possible, depending on the expected interaction between them two and also on the characteristics of the scheme we aim to devise (explicit, implicit, etc.). A couple of enlightening examples can be found in [16] and [54]. In particular, the former describes the coupling procedure step by step in a very clear and easy-coding manner, both in case of explicit and implicit coupling, associating the 0D discretized model we derived for the 3-elements Windkessel, and represented by 3.37, to a Fractional-Step solver for the incompressible Navier-Stokes equations. Since we have not introduced this method yet, here we only aim to provide a first idea of the coupling strategy, assuming to have already decoupled the treatment of pressure from that of the velocity field in the equations, a procedure which will be clarified in the subsequent section but which essentially exploits the new role played by the pressure in incompressible flows. refer to figure 3.6 to have an example of the coupling main idea. Following in particular the approach suggested by the

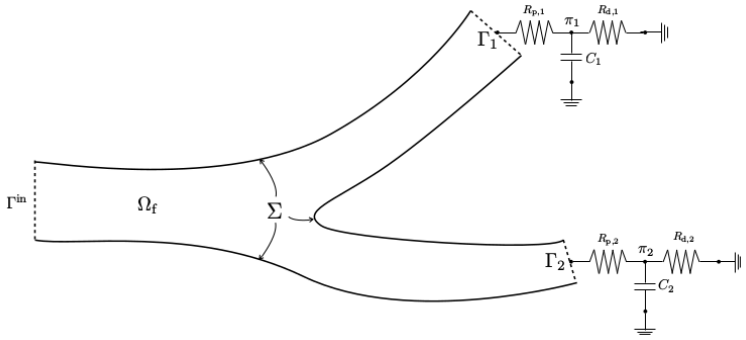


Figure 3.6: Representation of double coupling between a higher-dimension computational domain and 2 different WK3 models, through its 2 outflow sections.

authors in [16], with symbols π and P have been indicated respectively the *distal* and the *proximal* pressures, corresponding to the pressure at the parallel elements in the 0D model and to the one at the outflow section of the multi-D domain (entrance of the 0D one). Thus, they base the whole discussion on the following model system:

$$\begin{cases} C\frac{\mathrm{d}\pi}{\mathrm{d}t} + \frac{\pi}{R_d} = Q \\ P = R_p Q + \pi \end{cases} \tag{3.39}$$

Then, employing a backward Euler time discretization, they get:

$$\begin{cases} \pi^k = \alpha\pi^{k-1} + \beta Q^k \\ P^k = \gamma Q^k + \alpha\pi^{k-1} \end{cases} \tag{3.40}$$

Where there have been introduced the quantities:

$$\alpha = \frac{R_d C}{R_d C + \Delta t}, \quad \beta = R_d(1-\alpha) \quad \text{and} \quad \gamma = R_p + \beta$$

Actually, this is nothing but the same scheme we presented in 2.41. At this point, among the wide range of strategies reported in the literature on this subject, the authors here cited suggest to couple the two different scale models after the *predictor* step represented by the already introduced equation 3.36, in such a way to build a *corrective pressure field*, which accounts for the effect exerted by the 0D model on the multi-D one throughout its boundary conditions, meant to ensure the incompressibility constraint to the solution. To briefly summarize the mentioned procedure:

$$\begin{aligned}
\text{Multi-D:} \quad & \boldsymbol{V}^k \quad \longrightarrow \quad \boldsymbol{V}^{k+\frac{1}{2}} \quad \longrightarrow \quad Q^k_{out} = Q^k_{wk3} \\
\text{WK3:} \quad & Q^k_{wk3} \quad \longrightarrow \quad P^k_{wk3} = P^k_{out} \\
\text{Multi-D:} \quad & P^k_{out} \quad \longrightarrow \quad p^k_{out} \quad \longrightarrow \quad \boldsymbol{V}^{k+1}
\end{aligned}$$

3.3 Fractional Step Projection Method

Once having introduced all the numerical tools needed in order to handle differential problems of any sort, including coupling techniques between different order models, we can now focus on the actual numerical method commonly employed, and adopted also for this work, for the resolution of incompressible viscous flow problems. Within the next two subsections, the so-called *Fractional-Step Projection Method* (in the following FSM) will be introduced and deeply investigated, moving from its theoretical background and mathematical basis, up to its final time and space discrete formulation. Then, an essential part will be dedicated to the treatment of boundary conditions, since it does not consist of a trivial issue, conversely, it represents the most important aspect to be considered in order for the problem to be said as *well posed*. Concerning bibliographic references, we cannot omit to cite the first one who proposed such method as a successful scheme devised for incompressible problems, Alexandre Chorin, with his first publication [25] in 1967-1968 (see also [24] for a more recent version). Since its introduction, several different versions of this method have been proposed later by the authors, and a well constructed overview of these can be found in [41], in which they have been classified basically into three distinct categories:

- pressure-correction methods;
- velocity-correction methods;
- consistent splitting methods.

To our aims, only the first group of methods will be investigated, concerning those schemes based on a decoupling between velocity and pressure, which are therefore treated in a separate manner such that a different equation (or a system of them in case of 2D and 3D problems) is associated to each of their fields. Essentially, what is proposed by pressure-correction FSMs is to look for a corrective pressure field, or a sort of, exploited subsequently to impose the incompressibility condition on the velocity field, which meanwhile has been advanced in time by solving its evolutionary discrete system of parabolic equations. Some

of those schemes tend to perform this splitting either partially, by including still a pressure contribution in the velocity evolutionary equation, or completely, by omitting such terms and considering them later only in the corrective step of the method. Here we will deal with the former, always including also the pressure contribution in the evolutionary prediction step, by employing a scheme which basically looks like the one derived in 3.36. In addition to what it is stated by the aforementioned authors in their cited works, the following contents are inspired to the introduction given in [65, p.178-188], again to the slides by Professor D'Ambrosio [28] (PoliTo) or, for a fairly clear and easy introduction, to the Wiki's webpage [7].

3.3.1 Helmholtz-Hodge Decomposition

What follows is a brief recall of the so-called *Ladyzhenskaya's theorem*, a useful orthogonal decomposition which is meant to be a special case of the more general *Helmholtz-Hodge orthogonal decomposition*. Deeper details, as well as stronger mathematical background concerning this topic, can be found in [26, p. 31-45] and [10, p. 95-101].

Let us resume the incompressible vectorial formulation of the continuous Navier-Stokes equations 2.27:

$$\begin{cases} \nabla \cdot \boldsymbol{V} = 0 \\ \dfrac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \nu \nabla^2 \boldsymbol{V} \end{cases}$$

A problem defined in a spatial-temporal domain $\Omega \times (0, T)$, and associated with the following boundary and initial conditions:

$$\begin{cases} \boldsymbol{V}(\boldsymbol{x}, t) = \boldsymbol{V}_\Gamma \quad \text{on} \quad \partial\Omega = \Gamma, \forall t \\ \boldsymbol{V}(\boldsymbol{x}, t = 0) = \boldsymbol{V}_0(\boldsymbol{x}) \quad \text{in} \quad \Omega \end{cases}$$

Whose time discrete formulation, according to expressions 3.36, could look like:

$$\frac{\boldsymbol{V}^* - \boldsymbol{V}^k}{\Delta t} = -\boldsymbol{V}^k \cdot \nabla \boldsymbol{V}^k + \frac{1}{\rho} \nabla p^k + \nu \nabla^2 \boldsymbol{V}^k \tag{3.41}$$

At this point, one introduces the above mentioned theorem formulated by Ladyzhenskaya, which states:

Ladyzhenskaya decomposition theorem (Helmholtz-Hodge orthogonal decomposition). *Any vector field $\boldsymbol{V}$, defined in a proper set Ω, admits the unique orthogonal decomposition:*

$$\boldsymbol{V} = \nabla \Phi + \nabla \times \Psi \tag{3.42}$$

Where $\nabla \times \Psi$ is a solenoidal field with zero normal component on the boundary Γ of Ω, whereas $\nabla \Phi$ is an irrotational field with non-zero component on Γ.

Thanks to the previous property, we can now decompose our velocity field into a solenoidal and an irrotational part, so that $\boldsymbol{V} = \boldsymbol{V}_{irr} + \boldsymbol{V}_{sol}$. Now, let us suppose to be interested only in the solenoidal part of that decomposition, then in order to obtain it

from the original field, we could proceed by applying the divergence operator to it, so that, considering the splitting deriving from 3.42, we have that:

$$\nabla \cdot \boldsymbol{V} = \nabla \cdot \boldsymbol{V}_{irr} + \nabla \cdot \boldsymbol{V}_{sol} = \nabla \cdot \nabla\Phi + \nabla \cdot \nabla \times \Psi = \nabla^2\Phi + 0$$

Because of the properties of the nabla operator. This way, one obtains a Poisson's problem for the scalar field Φ, whose solution would in turn lead to the determination of the solenoidal component $\boldsymbol{V}_{sol}$ of $\boldsymbol{V}$, by resorting again to 3.42, so that:

$$\boldsymbol{V}_{sol} = \boldsymbol{V} - \nabla\Phi$$

Effectively, the idea exploited by this decomposition, depicted in figure 3.7, consists basically of projecting the original field onto a space of divergence-free fields, denoted with $\mathcal{J}$ formally through a projective operator that we could indicate with $\mathcal{P}_{\mathcal{J}}$, such that:

$$\mathcal{J}_\Omega = \{\boldsymbol{V} \in \Omega : \nabla \cdot \boldsymbol{V} = 0\}$$

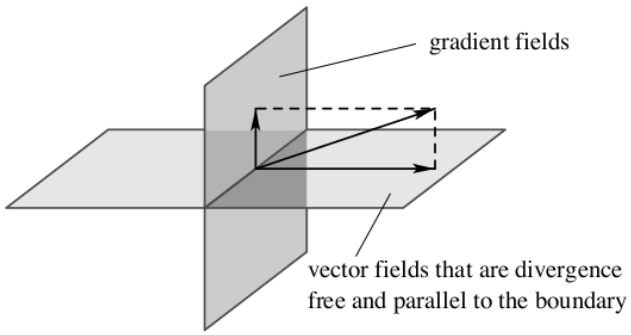


Figure 3.7: Helmholtz-Hodge orthogonal decomposition illustration.

3.3.2 FSM Algorithm

We now aim to apply this fundamental result to our incompressible Navier-Stokes problem. In other words, we will try to extract the divergence-free component of the solution to the Navier-Stokes equations by applying the previously introduced projector operator $\mathcal{P}_{\mathcal{J}}$, then:

$$\frac{\partial \boldsymbol{V}}{\partial t} = \mathcal{P}_{\mathcal{J}}\Big(- \boldsymbol{V} \cdot \nabla \boldsymbol{V} - \nabla p + \nu \nabla^2 \boldsymbol{V}\Big)$$

If we suppose to compute the solution to equation 3.41, thus thinking of it as a traditional evolutionary parabolic equation which is let advance in time, then we would get a new velocity field $\boldsymbol{V}^*$ not respecting the incompressibility constraint, since the related continuity equation would not have been applied yet. This first time integration is therefore called *Prediction Step*, and it consists of the first main step of the homonymous FSM method. For this reason, we will think from now on about that step as it was the first half-step of the complete time evolutionary step which leads from $\boldsymbol{V}^k$ to $\boldsymbol{V}^{k+1}$, providing thus an

intermediate and not divergence-free field $\boldsymbol{V}^{k+\frac{1}{2}}$. Therefore, we formally have for the prediction step that (by introducing also the spatial discretization already discussed in section 3.2):

$$\begin{cases} \boldsymbol{V}_{i,j}^{k+\frac{1}{2}} = \boldsymbol{V}_{i,j}^{k} + \Delta t \Big(- \boldsymbol{V}_{i,j}^{k} \cdot \left(\nabla_\delta \boldsymbol{V}^k\right)_{i,j} + \frac{1}{\rho} \left(\nabla_\delta p^k\right)_{i,j} + \nu \left(\nabla_\delta^2 \boldsymbol{V}^k\right)_{i,j} \Big) \\ \boldsymbol{V}_{i,j}^{k+\frac{1}{2}} = \left(\boldsymbol{V}_\Gamma^{k+\frac{1}{2}}\right)_{i,j} \quad \text{on} \quad \partial\Omega = \Gamma, \, \forall k \\ \boldsymbol{V}_{i,j}^{k=0} = \left(\boldsymbol{V}_0\right)_{i,j} \quad \text{in} \quad \Omega \end{cases} \tag{3.43}$$

Where, obviously, the discrete nabla operator defined as $\nabla_\delta(.) = \left(\frac{\delta(.)}{\delta x}, \frac{\delta(.)}{\delta y}\right)$ resumes the discrete schemes previously devised for these equations.

In order to impose the incompressibility condition on the resulting velocity field $\boldsymbol{V}^{k+\frac{1}{2}}$, we now resort again to the previously introduced Helmholtz decomposition, aiming then at projecting the predicted field onto a divergence-free space, by splitting it into the sum of a solenoidal velocity field $\boldsymbol{V}^{k+1}$ and the gradient of a scalar function $\nabla\Pi$, or $\nabla\Phi$, among the most popular notations, which is in turn an irrotational field:

$$\boldsymbol{V}^{k+\frac{1}{2}} = \boldsymbol{V}^{k+1} + \frac{1}{\rho} \nabla \Phi^{k+1}$$

Then, as mentioned before, by applying the divergence operator to that decomposition, one obtains the following Poisson's equation:

$$\nabla \cdot \boldsymbol{V}^{k+\frac{1}{2}} = \frac{1}{\rho} \nabla^2 \Phi^{k+1} \tag{3.44}$$

Where the solenoidal component of the decomposed field is nothing but the continuity equation which states the incompressibility condition $\nabla \cdot \boldsymbol{V}^{k+1} = 0$ we wanted to restore in the predicted field. The scalar field Φ^{k+1} actually represents the corrective field which is able to ensure such incompressibility constraint on the predicted field $\boldsymbol{V}^{k+\frac{1}{2}}$, thus the resolution of the related Poisson's problem is what is more commonly known as *Projection Step* of the complete FSM, the second one after the prediction step. By its solution, one can now compute the new divergence-free field $\boldsymbol{V}^{k+1}$ by resorting again to Helmholtz decomposision's definition, so that:

$$\boldsymbol{V}^{k+1} = \boldsymbol{V}^{k+\frac{1}{2}} - \frac{1}{\rho} \nabla \Phi^{k+1} \tag{3.45}$$

In other words, if we denote with $\Phi^{k+1} = \Delta t \, (p')^{k+1} = \Delta t (p^{k+1} - p^k)$, it is pretty straightforward to understand that what has been done here is nothing but introducing a corrective pressure field p' capable of enforcing the divergence-free condition, and thus representing somehow what is the actual role played by the pressure in incompressible flows: it is not the usual thermodynamic variable, but it is rather a forcing player which ensures the incompressibility of the field all over the temporal evolution of the solution, step by step, as expressed by the Pressure Poisson equation derived at the beginning of this work, in section 2.3.3. For this reason, expression 3.45 represents the final so-called

Correction Step of the FSM, the third one, which could be also thought, together with the second step, as the application of the projective operator introduced in the previous section:

$$\boldsymbol{V}^{k+1} = \mathcal{P}_{\mathcal{J}}(\boldsymbol{V}^{k+\frac{1}{2}})$$

In fact, by rearranging terms in 3.45, one obtains simply the second half-step which completes the single time integration started with the prediction one in 3.43, that is:

$$\begin{cases} \dfrac{\boldsymbol{V}_{i,j}^{k+1} - \boldsymbol{V}_{i,j}^{k+\frac{1}{2}}}{\Delta t} = -\dfrac{1}{\rho}\nabla_{\delta}\Phi_{i,j}^{k+1} \\ \nabla \cdot \boldsymbol{V}_{i,j}^{k+1} = 0 \\ \boldsymbol{V}_{i,j}^{k+1} \cdot \boldsymbol{n} = (\boldsymbol{V}_{\Gamma}^{k+1})_{i,j} \cdot \boldsymbol{n} \quad \text{on} \quad \partial\Omega = \Gamma, \forall k \end{cases} \tag{3.46}$$

In the problem above there appears clearly the new boundary condition applied to the velocity field, which consists now essentially of a simple inviscid problem, meaning that we have to deal with an *inviscid substep*. This is why the first half-step of the method, represented by 3.43, is sometimes referred to as the *viscous substep*. As it concerns the nature of that boundary condition, it is a direct consequence of the projection operation on the space $\mathcal{J}$, through Ladyzhenskaya's decomposition, which gives rise to a configuration such as the one depicted in figure 3.7. Since we substantially lose the control on the boundary condition associated with the tangential velocity component, we thus have that:

$$\boldsymbol{V}_{i,j}^{k+1} \times \boldsymbol{n} \neq (\boldsymbol{V}_{\Gamma}^{k+1})_{i,j} \times \boldsymbol{n} \quad \text{on} \quad \partial\Omega = \Gamma, \forall k$$

Anyway, in general, the condition on the tangential component has already been imposed within the first half-step 3.43 on $\boldsymbol{V}^{k+\frac{1}{2}}$, therefore making the error small enough to be considered negligible (see [65, p. 188]).

A very wide range of implementation examples for such numerical method exist and are strongly present in the literature, whenever incompressible problems are encountered. To cite some, we address the works by the supervisors of this master's thesis internship at Inria (Bordeaux), such as [15] and [14]. In the latter, the author suggests also to adopt the idea of a staggered grid (figure 3.8), by computing the face centered velocities right after the first half-step of the method, that is finding the interface components:

$$\mathcal{U}_{i,j}^{k+\frac{1}{2}} = u_{(i,j)-\frac{1}{2}}^{k+\frac{1}{2}} \quad \text{and} \quad \mathcal{V}_{i,j}^{k+\frac{1}{2}} = v_{(i,j)-\frac{1}{2}}^{k+\frac{1}{2}}$$

Starting from their node values, through some approximation technique, in order to improve the corrective pressure field smoothness resulting from the following solution of the related Poisson's problem. This approach has also been adopted for the present work, so that we are able to compute the left-hand side of 3.44 starting from the interface centered velocity components to build the finite divergence operator. A clear scheme summarizing the here presented algorithm concerning the implementation of the FSM is provided in table 3.1.

$$\left(\frac{\delta u}{\delta x}\right)_{i,j}^{k+\frac{1}{2}} + \left(\frac{\delta v}{\delta y}\right)_{i,j}^{k+\frac{1}{2}} = \frac{\mathcal{U}_{i+1,j}^{k+\frac{1}{2}} - \mathcal{U}_{i,j}^{k+\frac{1}{2}}}{\Delta x} + \frac{\mathcal{V}_{i,j+1}^{k+\frac{1}{2}} - \mathcal{V}_{i,j}^{k+\frac{1}{2}}}{\Delta y}$$

In addition, in the same work, different time integration schemes are also proposed instead of the simple explicit Euler here adopted, such as the trapezes' 3.18 for the viscous terms, and the Adam Bashforth's 3.20 for the convective portion. To conclude, with regards to

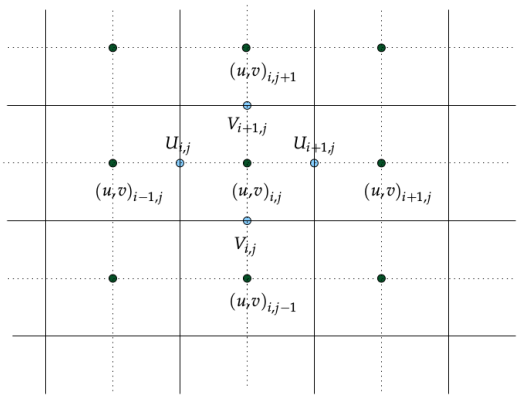


Figure 3.8: Staggered grid with interface velocity components representation.

the role assigned to the pressure within the first and second half-steps of the method, as mentioned it is supposed to be a corrective field capable of ensuring the incompressibility of the solution, step by step in time. However, since its introduction by Chorin and then by Teman, the FSM has been misunderstood by the different authors who exploited its structure in order to face incompressible problems, especially with respect to the actual identity of the corrective scalar field Φ, which is commonly given the name of *corrective pressure field*, as said. To this aim, another short article was published by Teman in 1991, reported in [79], who therefore attempted to clarify once and for all the originally thought meaning of that scalar quantity, giving also a stronger knowledge on its derivation in order to allow for the determination of the most proper boundary conditions the related Poisson's equation must be associated with. In that context, the author reminds of the original formulation of the FSM proposed by Chorin, which should be interpreted as an approximation of an evolutionary equation for the velocity field (thus not for the whole momentum balance equation), followed by an orthogonal projection, according to the Helmholtz's decomposition, finally leading to a new divergence-free velocity field. This procedure involves the introduction of a «technical quantity (Φ), a mathematical auxiliary that must be introduced for the construction of $\mathcal{P}_{\mathcal{J}}$» (Teman in [79]). According to the author then, that quantity must not be confused with the pressure , unless we include the latter in the discretized 1st half-step evolutionary equation through some proper approximation (as shown in equation 3.43). That way, the scalar quantity Φ do approximates the pressure, but, still according to Teman, «in a very weak sense», meaning that a boundary layer of the normal derivative of p will appear next to the domain's frontiers. The issue regarding such boundary conditions and their proper treatment is deeply investigated in the following section.

Table 3.1: Summary of the Fractional Step Projection Method algorithm.

Step	Start	End
Prediction:	$u_{i,j}^{k}$, $v_{i,j}^{k}$	$\mathcal{U}_{i,j}^{k+\frac{1}{2}}$, $\mathcal{V}_{i,j}^{k+\frac{1}{2}}$
Projection:	$(\frac{\delta u}{\delta x})_{i,j}^{k+\frac{1}{2}} + (\frac{\delta v}{\delta y})_{i,j}^{k+\frac{1}{2}}$	$\Phi_{i,j}^{k+1} = \Delta t(p_{i,j}^{k+1} - p_{i,j}^{k})$
Correction:	$u_{i,j}^{k+\frac{1}{2}}$, $v_{i,j}^{k+\frac{1}{2}}$, $\Phi_{i,j}^{k+1}$	$u_{i,j}^{k+1}$, $v_{i,j}^{k+1}$

3.3.3 Boundary Conditions and Posedness

As anticipated, this section is completely dedicated to the definition of the most suitable boundary conditions for the examined problem, that is an incompressible flow simulation of an aortic dissection. Considering the time integration algorithm developed and widely introduced within the previous section 3.3.2, one can easily notice that a large number of boundary and initial conditions are necessary in order to properly define the mathematical problem, and some of those could even be not that obvious and trivial to determinate. We thus organise the present section in two parts: the first one will focus on the boundary condition for the velocity field, whereas the second will be totally dedicated to those applying to the pressure field or, more precisely, to well-define the corrective pressure Poisson's equation problem.

Velocity Boundary Conditions:

For the sake of completeness, this paragraph dedicated to the definition of velocity field's boundary condition has also been inserted, even though they are fairly easy to devise, to be adapted to the particular tackled problem, and also to be implemented, thus offering deep ductility and a large range of different opportunities. Basically, one could think of the 2D domain Ω as bounded by 4 different portions of the frontier, so that $\partial\Omega = \Gamma = \Gamma_{in} \cup \Gamma_{up} \cup \Gamma_{down} \cup \Gamma_{out}$, as shown in figure 3.9. This way, 4 distinct conditions need to be imposed, one for each portion of the entire boundary. Usually, proceeding with respect to what has been found out thanks to the literature contributions, exposed in section 1.3, one tends to provide Dirichlet-like conditions at least on the three boundaries denoted as: Γ_{in}, Γ_{up} and Γ_{down}, imposing a traditional velocity profile on the former, while a traditional no-slip condition is applied on the last two. As it concerns the outflow section, Γ_{out}, a no-stress type condition seems to be the best possible choice, since it represents an easy-coding and consistent condition with respect to what will be stated for the pressure on that boundary. Among the different spatial distributions which are possible to be imposed as inflow condition, one could resort to the most traditional Poiseuille profile, to the more complicated and spreadly adopted in hemodynamics Womersley profile, or even to a simple polynomial profile (usually a 10th degree polynomial is selected in this case). We then summarize the last statements in the attached table 3.2. Concerning the mentioned inflow velocity profiles, indicated briefly with $u_{in}(y)$, we could exploit

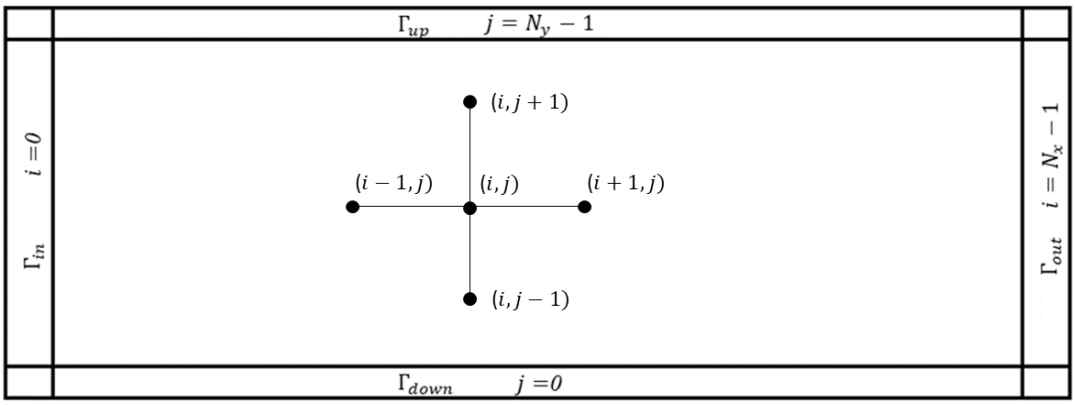


Figure 3.9: Illustration of the computational grid with emphasis on its frontiers Γ.

Table 3.2: Velocity boundary conditions.

Boundary	Type	Expression	Coding
Γ_{in}	Dirichlet	$\begin{cases} u(0,y,t) = u_{in}(y) \\ v(0,y,t) = 0 \end{cases}$	$\begin{cases} u_{0,j}^{k+\frac{1}{2}} = u_{in}(y_{0,j}) \\ v_{0,j}^{k+\frac{1}{2}} = 0 \end{cases}$
Γ_{up}	Dirichlet	$\begin{cases} u(x,Ly) = 0 \\ v(x,L_y) = 0 \end{cases}$	$\begin{cases} u_{i,N_y-1}^{k+\frac{1}{2}} = 0 \\ v_{i,N_y-1}^{k+\frac{1}{2}} = 0 \end{cases}$
Γ_{down}	Dirichlet	$\begin{cases} u(x,0) = 0 \\ v(x,0) = 0 \end{cases}$	$\begin{cases} u_{i,0}^{k+\frac{1}{2}} = 0 \\ v_{i,0}^{k+\frac{1}{2}} = 0 \end{cases}$
Γ_{out}	Neumann	$\begin{cases} \nabla^2 u(L_x,y) = 0 \\ \nabla^2 v(L_x,y) = 0 \end{cases}$	$\begin{cases} u_{N_x-1,j}^{k+\frac{1}{2}} = 2u_{N_x-2,j}^{k+\frac{1}{2}} - u_{N_x-3,j}^{k+\frac{1}{2}} \\ v_{N_x-1,j}^{k+\frac{1}{2}} = 2v_{N_x-2,j}^{k+\frac{1}{2}} - v_{N_x-3,j}^{k+\frac{1}{2}} \end{cases}$

substantially the following formulations (where $R = \frac{L_y}{2}$):

- Poiseuille profile:

$$u_{Pois}(y) = u_{max}\left(1 - \left(\frac{y-R}{R}\right)^2\right) \tag{3.47}$$

- Womersley profile (refer to [69] and [80, p. 17-22]):

$$u_{Wom}(y) = \frac{i}{\rho\omega}\left|\frac{\partial\hat{p}}{\partial x}\right|\left(1 - \frac{J_0\left(i^{\frac{3}{2}}\frac{\alpha(y-R)}{R}\right)}{J_0\left(i^{\frac{3}{2}}\alpha\right)}\right) \tag{3.48}$$

Where $\alpha = \sqrt{\frac{\omega R^2}{\nu}}$ is the *Womersley number*, ω is the pulsating frequency of the flow, ν as usual is the kinematic viscosity, and J_0 is the so-called *first-type Bessel's function*:

$$J_0(s) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!k!} \left(\frac{s}{2}\right)^{2k}$$

- 10th degree polynomial profile:

$$u_{p10th}(y) = u_{max}\left(1 - \left(\frac{y-R}{R}\right)^{10}\right) \tag{3.49}$$

Furthermore, a time-varying profile could be imposed by adding a temporal law based on the physiological values related to a heart beating dynamics. Denoting with RR the duration of an entire heart beat ($RR \approx 0.8\,s$), and supposing to separate it into its cardiac phases, namely, systole and diastole, respectively equal to $T_{syst} \approx \frac{2}{5}$ and $T_{dias} \approx \frac{3}{5}$, one has that (where $\bar{t}$ denotes the time reduced to the first beat):

$$q(t) = \begin{cases} \sin\left(\dfrac{2\pi}{2T_{syst}}\bar{t}\right) & \text{if} \quad \bar{t} < T_{syst} \\ 0 & \text{if} \quad T_{syst} < \bar{t} < T_{dias} \end{cases} \tag{3.50}$$

So that now a time-dependent inflow condition can be imposed instead of a stationary one, by defining:

$$u(0, y, t) = u_{in}(y)\, q(t) \tag{3.51}$$

Occasionally, a sine-squared profile could also be adopted.

Pressure Boundary Conditions:

The issue regarding the determination of proper boundary conditions for the pressure field associated with the solution of an incompressible problem has always been given the primary importance, and oftentimes tackled by the authors who proposed several different possibilities, and who published many articles and papers concerning such challenging aspect. Briefly, it is a matter of providing the corrective pressure Poisson's problem 3.44 representing the projection step of the FSM with suitable boundary conditions for the variable Φ, or $p' = p^{k+1} - p^k$. A well constructed work is advised in [68], where a wide overview of different possible interpretations of an incompressible problem are presented (see next section 3.4), together with the treatment of their best boundary conditions in order to set up well-posed boundary value problems. This work, in turn, owes its discussion on boundary condition determination to the already mentioned Quartapelle's one [65, p. 187-188], and here, finally, a deep investigation on this topic is carried out in the following.

Let us consider, then, again the pressure Poisson equation for the corrective scalar field $\Phi^{k+1} = \Delta t(p^{k+1} - p^k)$:

$$\frac{1}{\rho}\nabla^2\Phi^{k+1} = \nabla \cdot \boldsymbol{V}^{k+\frac{1}{2}}$$

Actually, by considering equation 3.46, the problem associated with the 2nd half-step of the FSM, together with its boundary conditions, and also those attached to problem 3.43, that is the 1st half-step, one sees that by projecting the former onto the boundaries of the domain we have:

$$\frac{\boldsymbol{V}^{k+1} \cdot \boldsymbol{n} - \boldsymbol{V}^{k+\frac{1}{2}} \cdot \boldsymbol{n}}{\Delta t} = -\frac{1}{\rho} \nabla \Phi^{k+1} \cdot \boldsymbol{n} \tag{3.52}$$

At this point, considering velocity boundary conditions listed in table 3.2, we have that on the upper and lower frontiers of the domain the left-hand side of 3.52 nullifies, thus leading to the following homogeneous Neumann-like boundary condition for the corrective field Φ on such boundaries:

$$\nabla \Phi \cdot \boldsymbol{n} = \frac{\partial \Phi}{\partial \boldsymbol{n}} = 0, \quad \text{on} \quad \Gamma_{up}, \Gamma_{down} \tag{3.53}$$

The same holds also for the inflow section where, by following the procedure adopted above, now on Γ_{in} the left-hand side of 3.52 nullifies again because of the fact that $\boldsymbol{V}^{k+1} \cdot \boldsymbol{n}$ and $\boldsymbol{V}^{k+\frac{1}{2}} \cdot \boldsymbol{n}$ will be, in general, identical, since the velocity boundary condition is imposed exactly on the normal component of the field by 3.49, which in turn will be equal to that imposed on the complete velocity vector $\boldsymbol{V}^{k+\frac{1}{2}}$ on the same section by 3.43, being the latter normal to the inflow frontier, by definition of $u_{in}(y)$ in table 3.2. Moreover, the imposed inflow profile should already be divergence-free, as can be easily proved, and thus it should not need any sort of correction by means of an additional scalar field. Therefore, again a homogeneous Neumann boundary condition will always be applied also to the inflow section, for the scalar field Φ, such that:

$$\nabla \Phi \cdot \boldsymbol{n} = \frac{\partial \Phi}{\partial \boldsymbol{n}} = 0, \quad \text{on} \quad \Gamma_{in} \tag{3.54}$$

More interesting and tricky is the last case, concerning the outflow section of the domain. Generally, in the literature even on that boundary a homogeneous Neumann-like boundary condition is usually applied, as suggested by all the authors previously considered. Nevertheless, one of the main goals of this work was exactly to establish a proper and suitable boundary condition for the outflow section of an aortic channel, independently from the dissection included upstream, but taking into account the fact that a pulsating and thus non-stationary flow will be imposed as inflow condition. When such a situation occurs, a homogeneous Neumann-like boundary condition is no longer valid for such boundary, and another solution need to be devised in order to well-pose the problem. This statement will therefore be proven in the following. As long as we deal with stationary flows, i.e. stationary Poiseuille, Womersley or polynomial velocity profile set as inflow conditions without any temporal behaviour, we are still allowed to impose homogeneous Neumann-like boundary condition on the opposite section, that is:

$$\nabla \Phi \cdot \boldsymbol{n} = \frac{\partial \Phi}{\partial \boldsymbol{n}} = 0 \quad \text{on} \quad \Gamma_{out}, \quad \text{if} \quad u_{in} \neq u_{in}(y,t) \tag{3.55}$$

This is true since, by integrating equation 3.44 over the whole domain Ω (the predicted field is now denoted by $\boldsymbol{V}^*$ for brevity, in order to let alone the current considered time

step k):

$$\frac{1}{\rho}\int_{\Omega}\nabla^2\Phi\,d\Omega=\int_{\Omega}\nabla\cdot\boldsymbol{V}^*\,d\Omega \tag{3.56}$$

And then by applying the divergence theorem:

$$\frac{1}{\rho}\int_{\Gamma}\nabla\Phi\cdot\boldsymbol{n}\,d\Gamma=\int_{\Gamma}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma=$$

$$=\int_{\Gamma_{in}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{in}+\int_{\Gamma_{up}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{up}+\int_{\Gamma_{down}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{down}+\int_{\Gamma_{out}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{out}$$

Clearly, assuming homogeneous Neumann boundary condition on all the four frontiers of the domain, the integral $\int_{\Gamma}\nabla\Phi\cdot\boldsymbol{n}\,d\Gamma$ nullifies, and in order to have a consistent problem, also all the other terms on the right-hand side must undergo the same end. Namely, considering that $\boldsymbol{V}^*$ has already been imposed as null on the upper and lower walls, we have that:

$$\int_{\Gamma_{in}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{in}=-\int_{\Gamma_{out}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{out}$$

But the latter can only be satisfied when a stationary flow is imposed as inflow condition, otherwise those two integral will always be different, and the problem would therefore be said as ill-posed. Resuming equation 3.44, now we know that, in case of stationary inflow condition, the integral $\int_{\Omega}\nabla\cdot\boldsymbol{V}^*\,d\Omega$ of 3.56 must be zero, according to the homogeneous Neumann boundary conditions associated with the scalar field Φ on all boundaries. Thus, in order to improve stability and smoothness of the solution to that Poisson's problem, one could resort to a shortcut which allows for the predicted velocity field to have in advance a null integral of its divergence, by redefining:

$$\left(\nabla\cdot\boldsymbol{V}^*\right)_{new}=\left(\nabla\cdot\boldsymbol{V}^*\right)_{old}-\frac{\int_{\Omega}\left(\nabla\cdot\boldsymbol{V}^*\right)_{old}d\Omega}{\Omega} \tag{3.57}$$

So that now, obviously, $\int_{\Omega}\left(\nabla\cdot\boldsymbol{V}^*\right)_{new}d\Omega=0$. But let us now suppose to deal with a non-stationary problem, consisting of a pulsating inflow condition such as that prescribed by 3.51, then, if that was the case, we would have:

$$\int_{\Gamma_{in}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{in}\neq-\int_{\Gamma_{out}}\boldsymbol{V}^*\cdot\boldsymbol{n}\,d\Gamma_{out}$$

Which means that, although small variations are considered for the inflow mass rate by means of the law 3.50, the right-hand side of 3.56 could no more nullify, thus highlighting the fact that homogeneous Neumann boundary conditions cannot be imposed on the whole frontier Γ as previously done, in order for equation 3.56 to be consistent and for the entire problem to be well-posed in general. This is due to the fact that, during the 1st half-step of the method, namely the prediction step, one imposes a new value of the mass rate entering the domain from the inflow section, different from that previously imposed at the beginning of the old time step, and this change in the flow rate thus affects only those grid cells whose stencil extends up to the inflow section, but not those farther than that. This means that a strong variation in the divergence of the predicted field occurs

within those cells next to the inflow section, which will then be adjusted, in turn, by the corrective pressure field, provided that the solution of the associated Poisson's problem is realistically carried out, thus setting up appropriate boundary conditions. The integral $\int_\Omega \nabla \cdot \boldsymbol{V}^* \, d\Omega$ will almost never nullify (as it has been observed during the performed simulations), and for this reason it would not be necessary to resort to the *escamotage* 3.57 to the end of improving the regularity of the solution, conversely it would be even more damaging, and the simulation would then explode.

As a consequence, we are thus supposed to deal with the subsequent problems:

$$\begin{cases} \dfrac{1}{\rho}\nabla^2\Phi = \nabla \cdot \boldsymbol{V}^* & \text{in} \quad \Omega \\ \nabla\Phi \cdot \boldsymbol{n} = 0 & \text{on} \quad \partial\Omega = \Gamma \end{cases} \tag{3.58}$$

Whenever one has to face a stationary inflow problem, such as imposing a traditional velocity profile which does not change its mass rate in time. On the contrary, in the case of a time varying inflow distribution of velocity, the following problem must be tackled instead:

$$\begin{cases} \dfrac{1}{\rho}\nabla^2\Phi = \nabla \cdot \boldsymbol{V}^* & \text{in} \quad \Omega \\ \nabla\Phi \cdot \boldsymbol{n} = 0 & \text{on} \quad \Gamma_{in} \cup \Gamma_{up} \cup \Gamma_{down} \\ \Phi = \Phi_{out} & \text{on} \quad \Gamma_{out} \end{cases} \tag{3.59}$$

Where the condition imposed on the outflow boundary, named Φ_{out}, is a completely general Dirichlet-like boundary condition, which could in turn be either constant or time-varying, say $\Phi_{out}(t)$. Right the latter will be exactly the case of our interest, involving the adoption of Windkessel models (see section 2.5) in order to provide physiologic time-depending pressure signals, i.e. $\Phi_{wk}^{k+1}(t) = \Delta t(p_{wk}^{k+1} - p_{wk}^{k})$, which will be imposed as Dirichlet boundary conditions on the outflow section of the domain, while a pulsating inflow is entering it through the opposite inflow section, defining then finally a well-posed problem. Several test cases have been carried out regarding the issue of posedness, and the most valuable results will be presented in the dedicated chapter 5.

An elliptic problem is therefore associated with the second substep of the FSM, which implies that a linear system of equations will be solved at each time step in order to obtain the corrective pressure field meant to ensure the incompressibility of the velocity field, throughout the following correction step. We thus address what has been stated in section 3.1, where an example of elleptic problem had been discretized by adopting a finite difference technique, employing a centered scheme thus leading to expression 3.14. The same holds in the case of the corrective pressure Poisson's equation associated with the projection step of the FSM, where the Poisson's problem 3.58 or 3.59 has to be faced. To begin with, by developing the differential operators, we have the general expression:

$$\frac{1}{\rho}\left(\frac{\partial^2\Phi}{\partial x^2} + \frac{\partial^2\Phi}{\partial y^2}\right) = \frac{\partial u^*}{\partial x} + \frac{\partial v^*}{\partial y} \tag{3.60}$$

Then, we can introduce the discrete differential operators $\frac{\delta}{\delta x}$, $\frac{\delta}{\delta y}$, $\frac{\delta^2}{\delta x^2}$ and $\frac{\delta^2}{\delta y^2}$, thus coming

to the formulation associated with each grid point:

$$\frac{1}{\rho}\left(\frac{\delta^2\Phi}{\delta x^2}\right)_{i,j} + \frac{1}{\rho}\left(\frac{\delta^2\Phi}{\delta y^2}\right)_{i,j} = \left(\frac{\delta u^*}{\delta x}\right)_{i,j} + \left(\frac{\delta v^*}{\delta y}\right)_{i,j} \tag{3.61}$$

And then, by introducing a central second order scheme for each side of 3.61, and recurring in addition to the definition of interface velocity components $\mathcal{U}$ and $\mathcal{V}$, we have that:

$$\frac{1}{\rho}\left(\frac{\Phi_{i-1,j} - 2\Phi_{i,j} + \Phi_{i+1,j}}{\Delta x^2} + \frac{\Phi_{i,j-1} - 2\Phi_{i,j} + \Phi_{i,j+1}}{\Delta y^2}\right) = \frac{\mathcal{U}_{i+1,j} - \mathcal{U}_{i,j}}{\Delta x} + \frac{\mathcal{V}_{i,j+1} - \mathcal{V}_{i,j}}{\Delta y}$$

$$\text{with} \quad i \in (1, N_x - 2),\ j \in (1, N_y - 2) \tag{3.62}$$

A remark is very worth making at this point: as stated by a theorem cited in [68], the problem set up this way will really be well-posed if and only if, provided that consistent boundary conditions are imposed as explained previously in this discussion, a divergence-free field is chosen as initial condition, and, furthermore, the divergence of the solution remains null on the domain's boundary at each instant of time. For the reason just mentioned, we then will impose that $\left(\frac{\delta u^*}{\delta x}\right)_{i,j}$ and $\left(\frac{\delta v^*}{\delta y}\right)_{i,j}$ must be equal to zero if $i = 0$, $i = N_x - 1$, $j = 0$ or $j = N_y - 1$, $\forall t \in (0, T)$.

Concerning the treatment of boundary's stencils, then, additional equations are needed for those nodes to which a Neumann-type condition is applied, whereas, in case of a Dirichlet-type one, the solution $\Phi_{i,j}$ will be already prescribed. Proceeding then separately, we have that:

- Neumann nodes: a condition of type $\frac{\partial\Phi}{\partial \boldsymbol{n}} = 0$ is applied, which thus means that we could have:

  $$\frac{\partial\Phi}{\partial x} = 0 \quad \text{on} \quad \Gamma_{in} \cup \Gamma_{out}, \quad \text{and} \quad \frac{\partial\Phi}{\partial y} = 0 \quad \text{on} \quad \Gamma_{up} \cup \Gamma_{down}$$

  Then:

  $$\left(\frac{\delta\Phi}{\delta x}\right)_{0,j} = 0, \quad \left(\frac{\delta\Phi}{\delta x}\right)_{N_x-1,j} = 0, \quad \left(\frac{\delta\Phi}{\delta y}\right)_{i,0} = 0 \quad \text{and} \quad \left(\frac{\delta\Phi}{\delta y}\right)_{i,N_y-1} = 0$$

  Proceeding, for instance, with the condition on the inflow section (the extension to the other wall is then straightforward), we could think of the approximation of the 2nd order derivative as a finite difference between two adjacent 1st order derivatives, that is:

  $$\frac{\left(\frac{\delta\Phi}{\delta x}\right)_{1,j} - \left(\frac{\delta\Phi}{\delta x}\right)_{0,j}}{\Delta x} + \frac{\Phi_{0,j-1} - 2\Phi_{0,j} + \Phi_{0,j+1}}{\Delta y^2} = 0$$

  Obtained with a forward scheme 3.1, and by keeping in mind that $\left(\frac{\delta\Phi}{\delta x}\right)_{0,j} = 0$, because of the Neumann condition, and by approximating then $\left(\frac{\delta\Phi}{\delta x}\right)_{1,j}$ this time with a backward scheme 3.2, we have finally that:

  $$\frac{\Phi_{1,j} - \Phi_{0,j}}{\Delta x^2} + \frac{\Phi_{0,j-1} - 2\Phi_{0,j} + \Phi_{0,j+1}}{\Delta y^2} = 0$$

And this holds therefore for each of the other boundaries on which a homogeneous Neumann-type condition is imposed.

- Dirichlet nodes: a condition of type $\Phi = \Phi_D$ is directly applied, thus the solution is already specified. This will involve the surrounding node equations, evidently, and in order to give an example, we could show what happens to the neighbouring nodes when such a condition is applied at the outflow boundary Γ_{out}, that is $\Phi_{N_x-1,j} = \Phi_{out}$. We thus have that:

$$\frac{\Phi_{out} - 2\Phi_{N_x-2,j} - \Phi_{N_x-3}}{\Delta x^2} + \frac{\Phi_{N_x-2,j-1} - 2\Phi_{N_x-2,j} + \Phi_{N_x-2,j+1}}{\Delta y^2} = \left(\nabla_\delta \boldsymbol{V}^*\right)_{N_x-2,j}$$

Whenever one aims to impose directly a pressure condition instead, such as for the case we should deal with, regarding Windkessel-type boundary conditions, providing a value for $p_{wk}(t)$, then the related value of Φ_D to be imposed could be formulated as:

$$\Phi_D^{k+1} = \Phi_{wk}^{k+1} = \Delta t(p_{wk}^{k+1} - p_{wk}^k)$$

Where p_{wk}^{k+1} is the resulting pressure obtained thanks to the Windkessel 0D model at time $k+1$, useful to complete the time integration step of the FSM, whereas p_{wk}^k instead is the same one resulting from the previous time step.

Finally, we are supposed to deal with a linear system which assumes the compact form:

$$\boldsymbol{A}\boldsymbol{\Phi} = \mathbf{b} \tag{3.63}$$

Where:

$$\boldsymbol{A} \in \mathbb{R}^{N_xN_y \times N_xN_y}, \quad \boldsymbol{\Phi} \in \mathbb{R}^{N_xN_y}, \quad \text{and} \quad \mathbf{b} \in \mathbb{R}^{N_xN_y}$$

and where the coefficient matrix $\boldsymbol{A}$ exhibits the following penta-diagonal structure:

$$a_{r,c} = \begin{cases} -\dfrac{2}{\rho}\left(\dfrac{1}{\Delta x^2} + \dfrac{1}{\Delta x^2}\right) & \text{if} \quad c = r \\ \dfrac{1}{\rho \Delta x^2} & \text{if} \quad c = r \pm 1 \\ \dfrac{1}{\rho \Delta y^2} & \text{if} \quad c = r \pm N_x \end{cases}$$

In the previous definition, a new pair of indexes have been introduced referring to the elements of the matrix $\boldsymbol{A}$, since it could be more convenient to denote its elements adopting a different representation. Obviously, there exist a meaningful link between indexes (i, j) in the physical domain and (r, c) of the associated matrix, which is based on the following observations:

- $\boldsymbol{A}$ is composed by a number of rows equal to the total number of grid points we have in the computational domain, that is N_xN_y indeed, and each line of $\boldsymbol{A}$ corresponds then to a single grid point in the domain, representing the stencil exploited in order to approximate there the Poisson's equation. For this reason, the following relationship between each line r of $\boldsymbol{A}$ and the indexes (i, j) referring to the corresponding grid point holds:

$$r = jN_x + i$$

- within a row r of $\boldsymbol{A}$, as mentioned, an entire squared stencil used on the computational domain to the end of approximating the Poisson's equation must be represented, i.e. the surrounding points of (i, j): $(i+1, j)$, $(i-1, j)$, $(i, j+1)$ and $(i, j-1)$ of figure 3.9. Thus, additionally to the diagonal element $(r, c = r)$ of $\boldsymbol{A}$ corresponding to the investigated grid point (i, j), columns $r+1$ and $r-1$ will be involved, being associated to the right and left neighbours of (i, j), respectively, and then also those corresponding to $r + N_x$ and $r - N_x$, being associated this time to the upper and lower neighbours of (i, j), respectively, and therefore N_x points after or before with respect to (i, j);

A way more clear representation is provided by figure 3.10. Particular attention is needed

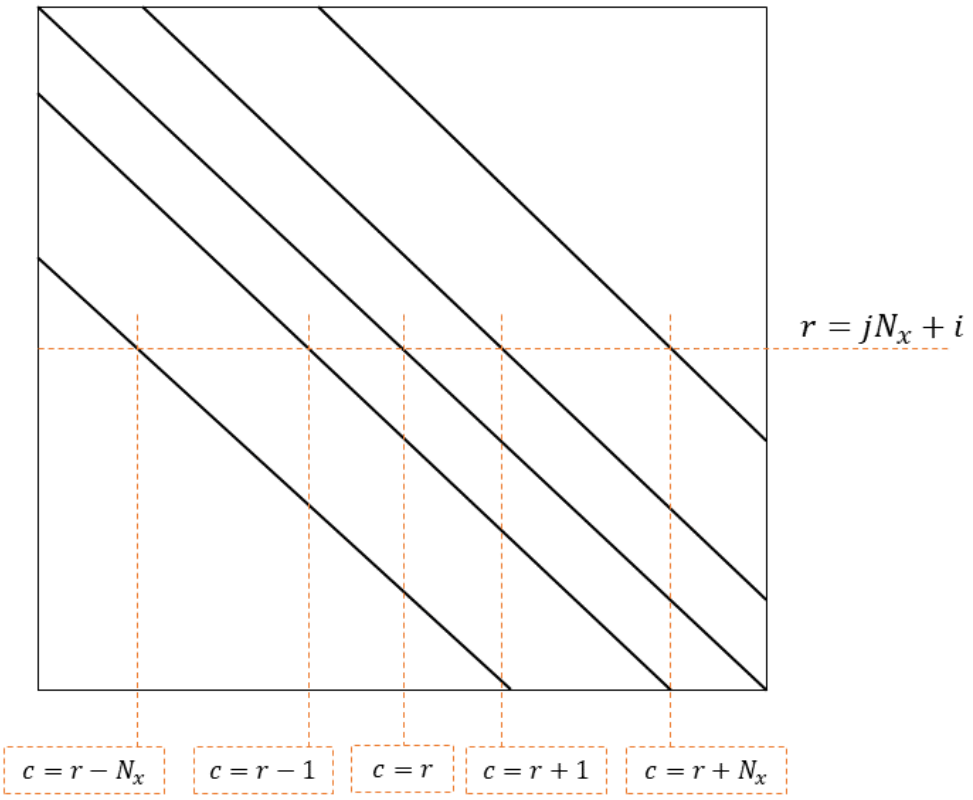


Figure 3.10: Schematic representation of Poisson's matrix assembling.

when dealing with the boundaries of the domain, and their treatment explained above is now taken into account. Depending on the position of the considered boundary (upper wall, inflow section, etc.) and also on the distance from its corners, different stencils are possible to be drawn. For instance, taking a point on the inflow section, for which a homogeneous Neumann boundary condition is applied by adopting the scheme aforementioned, only its right, lower and upper neighbours could be considered, since no left point exists in that case. But in case the investigated point of the inflow section laid exactly on the lower

corner , thus belonging also to the lower wall of the domain, its stencil would be further reduced, involving only its right and upper neighbours this time. As a consequence, the extension to all boundaries and corners follows the same idea. These features give the matrix $\boldsymbol{A}$ a block penta-diagonal structure. In order to provide a fully clear representation of the boundary condition treatment needed for assembling the coefficient matrix $\boldsymbol{A}$, one could refer to table 3.3, where stencil schemes and involved matrix elements are associated with each boundary of the domain.

Finally, Dirichlet-type boundary conditions need to be included too. whenever such condition was imposed on a boundary of the domain, say on the outflow section, one should consider that those nodes are no longer hosting unknown values of the solution Φ. For this reason, one could think of a reduced linear system, since the number of required equations drops due to the lower unknowns to be determined. However, a very practical way to proceed is to include those nodes in the system, by composing a corresponding number of rows in $\boldsymbol{A}$ whose filled elements only are those on the first diagonal, that is elements $(r, c = r)$, and by including the corresponding known value of Φ at right-hand side, as elements of vector $\boldsymbol{b}$. This way, the dimension of the system does not need to be changed, making the algorithm more straightforward to program and to modify. Only one last remark is worth doing at this point, regarding Dirichlet conditions imposition: instead of adding unit elements to the first diagonal of $\boldsymbol{A}$, in order to improve the linear system conditioning it would be better to insert scaled values with respect to the other elements of the matrix, thus a coefficient such as $\frac{1}{\rho \Delta x^2}$ will be employed on both those diagonal elements and the corresponding values of Φ in vector $\boldsymbol{b}$, obviously.

3.4 Alternative Methods for Incompressible NS Equations

This last section of the chapter is dedicated to the completion of the discussion on theoretical formulations and numerical methods devised in order to deal with incompressible problems, which are widely present in the literature. We will only consider briefly some of the most popular approaches to this aim, since we have already deeply introduced the method adopted for this master's thesis work, i.e. the Fractional Step Projection Method discussed within the previous section 3.3. What it follows, and much more than what will be reported here, could be found once again in [68], a greatly enlightening review on mathematical formulations of incompressible flows, on numerical method associated with them and, above all, on the derivation of proper and suitable boundary conditions which allow for the well-posedness of the problem. According to the introduction given by the author, the following partitioning is possible to be made regarding the issue of problem formulation:

- **velocity-based formulations:** which include the *Primitive-Variables* and the *Pressure Poisson Equation* ones, both already encountered in this thesis;
- **vorticity-based formulations:** among these, we could find the *Vorticity-Transport* formulation, or even the *Vorticity-Stream Function* one.

These are only a few of the known approaches devised with the aim of facing those challenging problems, as said, are they will be now briefly introduced in the following.

3.4.1 Velocity-Based Formulations

Primitive-Variables:

The first model we deal with is the most traditional one, which we have also already met within this work, since it represents exactly the most widely adopted formulation of the incompressible governing equations, based on the balance laws of mass and momentum derived in section 2.2 and 2.3. They are thus recalled here as a reminder, for the sake of completeness, written in non-dimensional form:

$$\begin{cases} \nabla \cdot \boldsymbol{V} = 0 \\ \dfrac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \dfrac{1}{Re} \nabla^2 \boldsymbol{V} \end{cases} \tag{3.64}$$

Where there appears the Reynold's number, defined as $Re = \frac{\rho |\boldsymbol{V}| L}{\mu}$. As already mentioned within this work, this is the most natural and physically meaningful representation of the governing equations, but, meanwhile, it poses also a hardly overcoming obstacle because of the coupling between the velocity and the pressure field it introduces. Basically, one would like to separate those variables in order to design simpler algorithms for the solution of such problems, and, furthermore, with the aim of deriving suitable and independent boundary conditions. The FSM cited above, and deeply explained in the previous section of this chapter, being right the one adopted for the purpose of this work, is based on this primitive variables formulation.

Pressure Poisson Equation:

Since no explicit relation is given for the pressure treatment in the previous formulation, in order to reach the decoupling discussed above a new set of equations is possible to be derived. Actually, we have already encountered also this new one, when we discussed the derivation of the pressure Poisson's equations for incompressible flows, with the aim of showing clearly the role played by the pressure terms in the system of equations, therefore governed by an elliptic equation not depending on time, and devoted to adjusting at each instant of time the incompressibility condition of the velocity field, restoring a divergence-free solution. We recall also such equation here below, for clarity:

$$\begin{cases} \nabla^2 p = -\nabla \cdot (\boldsymbol{V} \cdot \nabla \boldsymbol{V}) \\ \dfrac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \dfrac{1}{Re} \nabla^2 \boldsymbol{V} \end{cases} \tag{3.65}$$

As shown, now the new set of equations is composed by the same momentum balance equation as before, and the elliptic equation for the pressure in place of the continuity one for the velocity. This way, an explicit relationship for the pressure is thus given, even though the new order of derivation has reached two also for pressure terms, thus leading to the needing of an additional condition in order to well-define the problem. Moreover, determining those conditions for the pressure is not as straightforward as it could be for the velocity field. As it can be noticed, for the moment it does not seem possible to

separate pressure and velocity resorting to these formulations, making necessary to devise some new and innovative approach.

3.4.2 Vorticity-Based Formulations

Vorticity-Transport:

A different strategy would regard instead the involvement of new variables in the system of governing equations, by deriving a new form of those ones. One different way to tackle the problem of finding appropriate boundary conditions for the pressure could be, according to some authors, to eliminate its contribute from the equations. This can be done, for instance, by deriving a formulation concerning the curl of the velocity field, namely, the *vorticity*. Such formulation can be obtained by taking the curl $\nabla \times (.)$ of the momentum equation (by means of which the term regarding the pressure, ∇p, nullifies due to operational properties of ∇). Thus, assuming the definition of vorticity $\boldsymbol{\omega} = \nabla \times \boldsymbol{V}$, one has that:

$$\begin{cases} \nabla^2 \boldsymbol{V} = -\nabla \times \boldsymbol{\omega} \\ \dfrac{\partial \boldsymbol{\omega}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{\omega} = \boldsymbol{\omega} \cdot \boldsymbol{V} + \dfrac{1}{Re} \nabla^2 \boldsymbol{\omega} \end{cases} \tag{3.66}$$

Where we are now supposed to deal with a transport equation for the new quantity representing the vorticity field (*vorticity Helmholtz's equation*), and where the first relation, added in order to close the problem, has been obtained by taking again the curl of the definition of vorticity given above, and by applying the property according to which $\nabla^2 \boldsymbol{V} = \nabla(\nabla \cdot \boldsymbol{V}) - \nabla \cdot (\nabla \cdot \boldsymbol{V})$, still considering that $\nabla \cdot \boldsymbol{V} = 0$, obviously. The last formulation, then, allows for the dealing with incompressible problems from a different point of view, but the needing for proper initial and boundary conditions has not changed. While the former are fairly easy to find, simply by taking the curl of the old initial condition for the velocity field, for instance, the latter are not as trivial.

Vorticity-Stream Function:

The last formulation presented is again reachable by moving a step forward from the previous one, since it still exploits the vorticity equation. This time, a stream function ψ is introduced, starting from the cartesian components of the velocity field, as:

$$u = \frac{\partial \psi}{\partial y}, \qquad v = -\frac{\partial \psi}{\partial x}$$

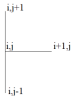
Then, the new system of equations becomes:

$$\begin{cases} \nabla^2 \psi = \omega \\ \dfrac{\partial \omega}{\partial t} \boldsymbol{V} \cdot \nabla \omega = + \dfrac{1}{Re} \nabla^2 \omega \end{cases} \tag{3.67}$$

This last approach could be advantageous, despite of the introduction of a new scalar field associated with other boundary conditions, in turn. Nevertheless, and unfortunately,

this approach falls when problems with geometrical dimension higher than two are faced. Actually, a very complex version of the stream function is still available also when 3-dimensional flows are taken into consideration, but their handling would not be that easy at all. Furthermore, the issue regarding boundary conditions for the vorticity field (a scalar field in 2D problems) is still present.

Table 3.3: Boundary condition handling for the assembling of coefficient matrix $\boldsymbol{A}$.

Boundary	**Stencil**	**Elements**	**Equation**
		$r = jN_x + i$	
		$\begin{cases} c = r \\ c = r + 1 \\ c = r + N_x \\ c = r - N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x^2} + \frac{\Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r - 1 \\ c = r + N_x \\ c = r - N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i,j} - \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j+1} - 2\Phi_{i,j} + \Phi_{i,j-1}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r + 1 \\ c = r - 1 \\ c = r - N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r + 1 \\ c = r - 1 \\ c = r + N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i+1,j} - 2\Phi_{i,j} + \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r + 1 \\ c = r - N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x^2} + \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r - 1 \\ c = r - N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i,j} - \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j} - \Phi_{i,j-1}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r + 1 \\ c = r + N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i+1,j} - \Phi_{i,j}}{\Delta x^2} + \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y^2}\right)$
		$\begin{cases} c = r \\ c = r - 1 \\ c = r + N_x \end{cases}$	$\frac{1}{\rho}\left(\frac{\Phi_{i,j} - \Phi_{i-1,j}}{\Delta x^2} + \frac{\Phi_{i,j+1} - \Phi_{i,j}}{\Delta y^2}\right)$

Chapter 4

Cartesian Monolithic Approach

So far, we have been introducing the mathematical model of incompressible fluid governing equations, and substantially the most widely spread numerical techniques developed in order to discretize it with the aim of providing an approximate solution to the problem. We have also addressed the issue of initial and boundary conditions, an essential aspect when considering the numerical formulation of physical phenomena, since their posedness relies strongly on the definition of proper and consistent external influences on the represented problem, as well as their direct implementation, their coupling with the computational domain, and so on. Our primary goal, though, was to perform simulations of type B aortic dissections, as explained in chapter 1, thus involving deforming bodies immersed in non-stationary fluid flows, with also moving external walls and a strong influence coming from neglected downstream compartments of the entire cardiovascular system, represented through appositely devised boundary conditions (i.e. Windkessel models, section 2.5). Beside the needing of suitable boundary conditions for the external frontier of the domain, deeply discussed in section 3.3.3, no specific treatment has been introduced yet in order to deal with immersed moving and deforming bodies, which consists substantially of a *Fluid-Structure Interation* problem (FSI in the following). The investigated situation could be that depicted in figure 4.1. More specifically, we will deal with a 2-dimensional problem referred to a simple reconstruction of an aortic channel, containing a floating and deforming membrane associated with the detached *tunica intima* causing the dissection, as shown in figure 4.2, in case of rigid external walls, or in figure 4.3, including also a deforming aortic channel's structure. Therefore, suitable techniques need to be devised

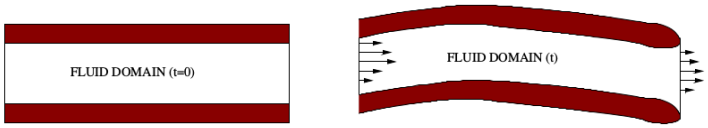


Figure 4.1: Representation of a deforming aortic channel domain, example of FSI-like problem.

in order to deal with such FSI problems, meaning that a way to impose appropriate boundary conditions even on immersed bodies has to be considered, as well as on the

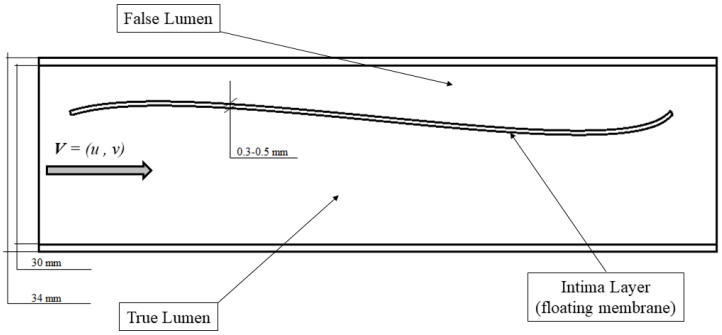


Figure 4.2: Aortic dissection domain representation: rigid aortic walls case.

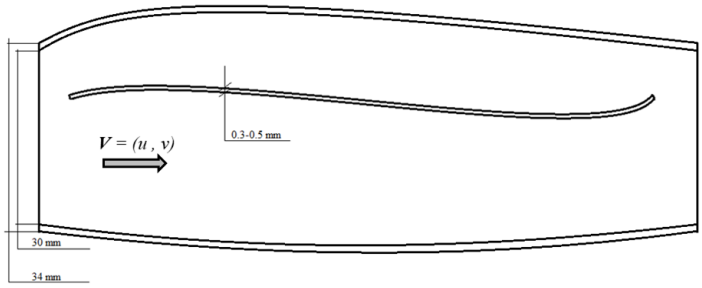


Figure 4.3: Aortic dissection domain representation: deforming aortic walls case.

external moving boundaries. Traditionally, one used to compute the whole displacement field associated with all moving and deforming bodies immersed in the domain, by adopting some structural models when necessary, starting from the fluid's velocity and pressure field solutions. Nevertheless, this approach led to the necessity of regenerating a new mesh after each time step computation, since the geometrical configuration had changed, and then new different boundaries had to be taken into account, in order to apply the most suitable boundary conditions. This type of approach is named as *lagrangian*, and relies on body-fitting meshes to allow for the problem solution, by making them evolve according to specific transport laws or by regenerating them after each time integration, as mentioned. In addition to the huge computational cost required by these direct and traditional methods, one has also to admit that in the case of multi-phase flows, complex configurations involving merging bodies or complicate topological mutation of the domain geometry, it would certainly be prohibitive to resort to those techniques. Thus, some new and innovative methods have been proposed in order to overcome such difficulties. By changing the perspective from which we look at the problem, therefore meaning by adopting an *eulerian* stand point instead, we would be able to handle those deforming and moving immersed bodies at once, with an incredibly reduced amount of efforts, moreover allowing for a straightforward imposition of boundary conditions on their surfaces, and therefore leading to a completely new and effective fashion to tackle FSI problems.

In the following, an overview of the most popular free interface localization techniques is first presented, focusing in particular on that adopted for the aim of this thesis work;

subsequently, the issue regarding the imposition of boundary conditions is addressed and deeply investigated, to conclude then, finally, with a summary of the so-called *fully eulerian monolithic approach*, the fundamental stand point which this entire work is based on.

4.1 Localization of Free Interfaces

The contents of the following section are taken from the review work [49], and also partially from [52], and from [32], where a systematic and clear classification is proposed for all those methods devised with the aim of studying the evolution of free interfaces within a given domain (fluid or not). As a reference, see figure 4.4. As mentioned, one first classification could be that distinguishing between:

- lagrangian methods: the whole grid evolves in time following the movement of all immersed bodies and their interfaces, being regenerated after each time integration step. It represents the classical choice (body-fitting meshes) in case of fixed geometries, but it turns out to be largely disadvantageous when FSI problems are considered;
- mixed lagrangian-eulerian methods: they are based on an eulerian description of the fluid evolution, an issue tackled through the adoption of fixed grids, but still keeping a lagrangian description of moving boundaries associated with immersed bodies, which are fitted with some lagrangian markers or different parametrized geometric entities, whose time evolution is carried out directly from a lagrangian stand point;
- eulerian methods: both the fluid and the immersed bodies undergo an indirect eulerian treatment, via the adoption of fixed grids and auxiliary functions, among the other strategies, exploited in order to study their motion.

Furthermore, another distinction can be proposed by focusing directly on the strategies adopted to follow and describe those moving interfaces, oftentimes integrating some eulerian and lagrangian aspects, that is the one between *front-tracking* and *front-capturing* methods.

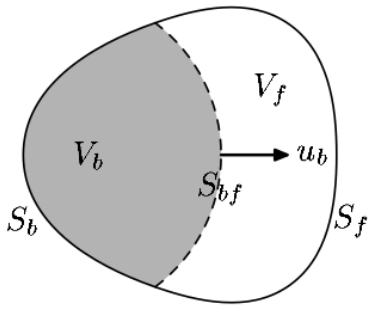


Figure 4.4: Free interfaces and moving boundaries typical domain configuration.

4.1.1 Front-Tracking Methods

Within this section we address to those methods concerned with the localization of freely-moving interfaces by means of direct strategies, such as lagrangian marker fitting, applied to the investigated surfaces or to the interfaces between fluid phases, or even volume tracking approaches, both described below.

Generic Front-Tracking:

All immersed bodies are localized through their surface, which is fitted with some special lagrangian markers, that are particles whose evolutionary behaviour will be studied from a lagrangian point of view, following their path in time and therefore allowing for the reconstruction of the body's interface after each time integration step. Note that the surface described by those markers is clearly only an approximation of the actual body surface, due to the fact that curved body cannot be perfectly reconstructed via piecewise linear interpolation of contour particles. See figure 4.5 for more clarity on this aspect. According to this approach, the treatment of markers evolution is almost completely

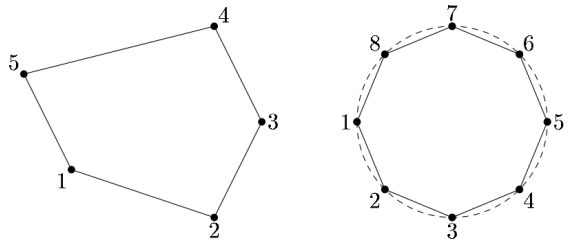


Figure 4.5: Lagrangian markers describing a body surface in accordance with a front-tracking approach.

decoupled from that of the fluid, managed by its governing equations, apart from the fact that the particle's trajectories depend directly on the velocity field coming from the solution of the equations, thus, by solving a problem such as:

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{X}_k}{\mathrm{d}t} = \boldsymbol{V}(\boldsymbol{X}_k, t) \\ \boldsymbol{X}_k(0) = \boldsymbol{X}_{k,0} \end{cases} \tag{4.1}$$

Where the trajectory of the k-th marker is observed, and the same procedure has to be applied to all the particles used to describe the body's surfaces. We should also specify that this method is not necessarily lagrangian, instead, according to the classification mentioned in the introduction to this section, it could also belong to a mixed eulerian-lagrangian family of techniques, since an eulerian stand point could be adopted, beside the lagrangian description of the particle behaviour in time, in order to solve the fluid flow on a fixed grid. A similar situation is reported in figure 4.8. In addition, some authors have also proposed a different version of such methods, according to which a thin region of the body next to the investigated moving surface is filled with lagrangian particles, in

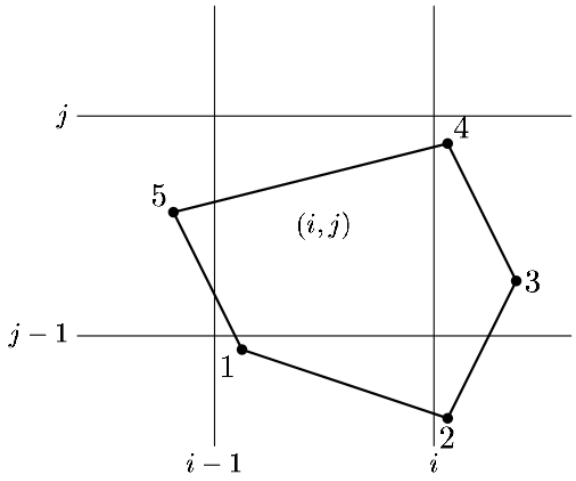


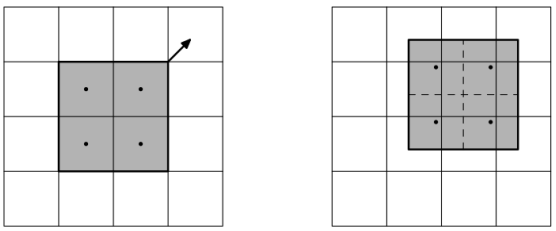
Figure 4.6: Lagrangian markers describing a body surface in accordance with a front-tracking approach onto a fixed cartesian grid.

order to provide a representation of its thickness. The geometrical properties of immersed bodies and of their surfaces, then, are determinable thanks to the interpolation techniques adopted for the surface reconstruction, thus throughout piecewise linear or curvilinear methods, always polynomial-based. It is worth remarking, finally, that these approaches could also be extended to 3D problems (where simple linear interpolation leaves the place to more complicated 3-dimensional surface reconstruction), but they turn out to be quite inefficient when topological changes occur within the problem domain, being then not very suitable for the case of our interest, as well as for multi-phase flow problems, for instance.

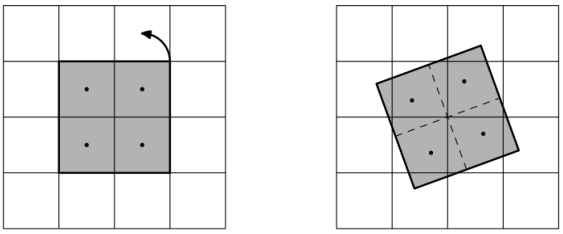
Volume-Tracking:

Another mixed eulerian-lagrangian method which employs the same lagrangian markers idea as before, but according to which now the latter are disseminated all over body's volumes, within each cell representing the internal region of them or crossed by their surface contours. Then, all particles are let evolve in time according again to the solution to the flow equations, in a lagrangian manner, and their trajectories are thus computed step by step, allowing for the reconstruction of the body shape at any instant of time (figure 4.7).

A better precision in representing the volume occupied by the solid bodies could be reached by adding more particles to each cell, as shown in figure 4.7, so that a volume percentage of the cell belonging to the body's volume could be estimated by counting the markers laying within the body's surface. Moreover, a particle refinement is possible to be operated in the proximity of the body's surface, in order to combine those benefits coming from a multiple particle-per-cell technique with a less expensive handling of internal cells, where only one marker each is placed.



(a) *Translation.*



(b) *Rotation.*

Figure 4.7: Lagrangian markers describing a body's volume in accordance with a volume-tracking approach onto a fixed cartesian grid.

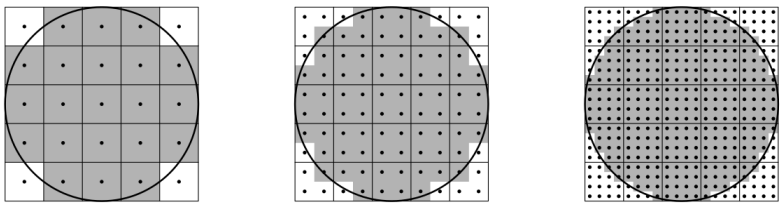


Figure 4.8: Lagrangian markers describing a body's volume in accordance with a volume-tracking approach onto a fixed cartesian grid, with multiple particle-per-cell.

Immersed Boundary:

This method, again exhibiting mixed eulerian-lagrangian features, will actually be the one adopted by the current work, and it will be more deeply investigated in the next dedicated section 4.2, since it represents exactly a way of imposing boundary conditions in an implicit manner. To the aim of the present introduction to free interfaces localization, we only mention it as part of those techniques capable of denoting the presence of an interface, and thus of an immersed body, moving through the fluid domain. Basically, it consists of the same eulerian treatment reserved to the fluid equations handling as before, associated with a direct lagrangian convective study of the particles describing again the shape of the immersed body. In addition to that, it is now possible to account for the interaction between the body and the fluid itself, by introducing a discrete function (or continuous, as it will be shown) meant to provide the needed information regarding the presence of the body, and thus affecting the fluid behaviour itself. Usually, a structural elastic model based on a stress-strain relationship is adopted, but this rather concerns the

next section 4.2 on the imposition of immersed boundary conditions.

4.1.2 Front-Capturing Methods

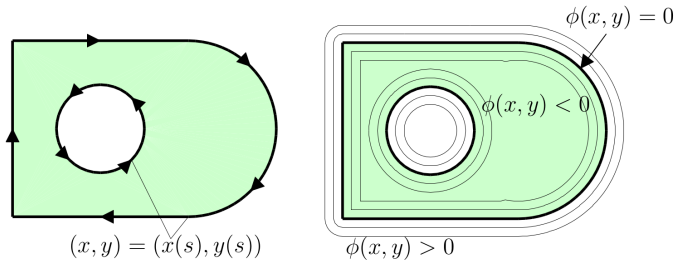
Among these methods dealing with a different selected approach for the localization of free interfaces, based this time on an indirect (or implicit) localization, reached by literaly capturing the moving layer within the fluid domain via the exploitation of some auxiliary function, or numerical quantity, one can find for instance the well-known Level-Set method (or LevelSet in the following, largely presented in their dedicated section 4.1.3), the Volume-of-Fluid method (VOF), the Phase-Field method, or even other combined approaches such as a coupled LevelSet/VOF method, or again a particle/LevelSet one.

Level-Set Method:

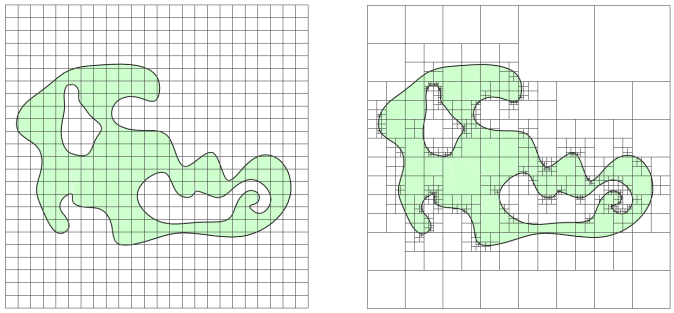
It consists of a pure eulerian approach developed for the interception of freely-moving interfaces within a fluid domain. Being an indirect eulerian method means that, contrarily to what has been said above about particle tracking along their trajectories, given by the velocity field solution of the governing equations for the fluid, this method operates the capturing procedure by defining a scalar function of the space (2 or 3-dimensional, being thus surprisingly ductile in that sense), namely the *Level-Set function*, whose level curves have a precise meaning, as the 0-level one, conventionally, is associated right with the body's surface which needed to be described somehow. This way, a fixed grid can still be employed in order to adopt an eulerian stand point to deal with the fluid motion, as usual, and no lagrangian markers need to be placed or applied neither to the body's surfaces, nor within their volumes. Then, a pure convection equation is solved at each instant of time to the aim of computing the motion of these level curves inside the computational domain, as they were transported by the flow in a mere passive-scalar manner, that is:

$$\frac{\partial \phi}{\partial t} + \boldsymbol{V} \cdot \nabla \phi = 0 \tag{4.2}$$

The resulting new profile of those level curves will then describe the new shape of the immersed body, represented by their 0-level, as anticipated, and the imposition of boundary conditions, as well as the FSI problem handling, would then be enormously simplified, enabling also for more complicated topology changes and more complex flow patterns to be represented. To give an idea, in figure 4.9 there are provided some examples of applications of these auxiliary scalar fields, whose 0-level curves allow for the individuation of the body's surfaces. As it regards its definition, an *euclidean distance function* is normally chosen, so that each value of ϕ in the domain will correspond to a precise distance from the 0-level curve, i.e. the body's surface (different signs are then used to distinguish points internal to the body from those external ones). Beside its powerful capacity and ductility, this approach reserves also an important drawback: the Level-Set function needs to be periodically reinitialized in order to restore its distance-function properties. All the discussion about this particular method will be resumed in the dedicated section 4.1.3, since it will be the strategy adopted to the aim of the current work, and thus also a numerical treatment of the problem will be provided, in order to associate it with the previous algorithms regarding the governing fluid equations handling.



(a) *Explicit vs. implicit treatment of a geometry topology.*



(b) *Cartesian vs. Quadtree/Octree grids.*

Figure 4.9: Examples of level-Set function exploitation for the description of complicated topologies (reference in [61]).

Volume-of-Fluid Method:

Similarly to the previous case of Level Set functions, even with VOF methods an auxiliary scalar field is defined all over the computational domain. By denoting with α this function, defined as the ratio between the volume associated with the first fluid with respect to the entire cell volume (or the volume of the body over that of the entire cell), one basically has that:

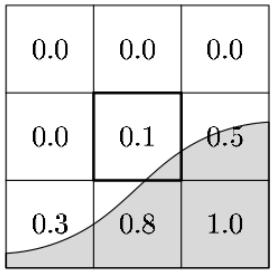
$$\alpha(\boldsymbol{x},t)=\begin{cases}1 & \text{if} \quad \boldsymbol{x}\in\Omega_s\\ 0 & \text{if} \quad \boldsymbol{x}\in\Omega_f\end{cases} \tag{4.3}$$

Where Ω_s and Ω_f refer respectively to the portion of the domain occupied by the solid (or by another fluid, in the case of two phase fluid problems), and to that filled with the fluid. Any cell containing both volume of solid bodies and fluid (or both the fluids) exhibits a value of α reasonably comprised between 0 and 1. Then, a convection problem is solved in order to study the evolution of those volumes within the domain, as time advances, and therefore a pure advection equation as the following one is associated with the problem:

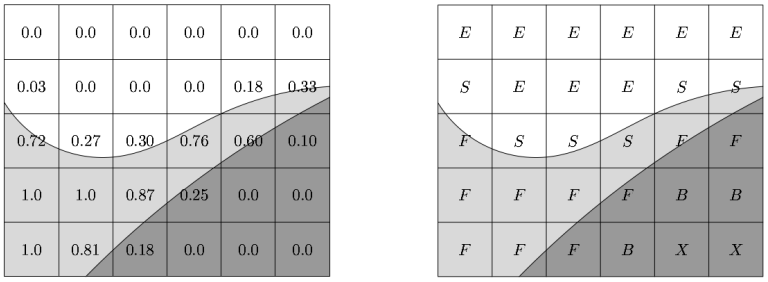
$$\frac{\partial\alpha}{\partial t}+\boldsymbol{V}\cdot\nabla\alpha=0 \tag{4.4}$$

Due to the discontinuous nature of the function α defined above, suitable upwind methods need to be employed in order not to allow for the interface to diffuse too much because of the diffusive errors introduced by those stabilized algorithms. Moreover, geometrical reconstruction of features such as curvatures and normals are not that straightforward,

being linked to the necessity of interpolating somehow that discontinuous distribution of α over the domain in order to provide an analytical expression. On the other hand, though, VOF method is the one exhibiting as main advantage the best mass conservation properties, more accurate than that associated with Level Set functions. Figure 4.10 shows two examples of application to a classic two phase problem, and to a free-surface one, where E denotes empty cells, F is associated with fluid cells, S indicates those cells containing the interface between the previous two phases, and finally B or X refer to body cells.



(a) *Two-phase problem.*



(b) *Three-phase free-surface problem.*

Figure 4.10: VOF examples of application on different multi-phase problems.

Phase-Field Method:

Another popular method concerning the implicit localization of moving interfaces is the one represented by the phase-field approach. According to this model, a configuration in which a solid body is immersed in a fluid medium, or alternatively with two different fluids constituting a two-phase domain, is modeled by a phase-field function based on a model of energy, capable of representing also a thin region of transition corresponding to the interface between the two materials. It thus consists of a diffuse interface method, such as a version of the volume tracking method already encountered in section 4.1.1 where a certain thickness was given to the interface representation. Typically, these phase field functions are based on some energy function equations, such as Cahn-Hilliard or Allen-Cahn ones (see [52]), which are both convection-diffusion equations including some specific source terms devised with the aim of accounting for surface tensions, interface diffusion, etc.

Among their main advantages, one should consider their ability to easily allow for the computation of geometrical properties of the moving interfaces, their straightforward extension to 3-dimensional cases, and their great adaptivity to complex topological changes in the domain, as well as we saw in the case of Level-Set functions. As a fundamental drawback, one should account for a grid refinement in those regions next to the represented interface, because of the strong change undergone by the phase function in describing those thin layers.

Level-Set/Volume-of-Fluid Method:

The present coupled method has been devised with the aim of combining the main advantages offered by a Level-Set approach, on the one hand, and by a VOF method on the other hand. This means that the geometrical properties associated with the interface, say its normals and curvatures, are reconstructed through the exploitation of a Level-Set function, whereas its movement within the fluid madium is obtained, by solving the classical advection equation previously introduced, thanks to the VOF approach, due to its much better mass conservation features. These two methods thus need to be properly synchronized, meaning that the VOF α field is extracted from the reconstruction allowed by the geometrical properties linked to the Level-Set information, the former, then, is advected by the velocity field until the completion of the time integration step, ensuring a good mass conservation, and finally a new Level-Set is defined as a distance-function by reinitialization with respect to the new surface represented by α distribution, and so all over again.

Particle/Level-Set Method:

This last introduced approach aims again to improve mass conservation properties of Level-Set functions, by introducing some lagrangian markers in the representation of the investigated moving interfaces, which are advected by the field and studied in a lagrangian way, as explained in the case of front-tracking strategies in section 4.1.1. Therefore, it actually consists of a mixed lagrangian-eulerian method, since a certain number of particles are studied in a lagrangian manner, following their trajectories when transported by the fluid. The goal of such method is thus to allow for a combined check on the correct evolution of the interfaces described by the Level-Set function, compared to the motion of these markers, assumed as the true interface displacement. Therefore, in the case a particle indicating a certain position of the interface was not in agreement with the corresponding information carried by the associated level-set function, then the latter would be properly reinitialized and redefined according to the actual position represented by the former, ensuring then a correct evolution of the interface by means of fluid transport.

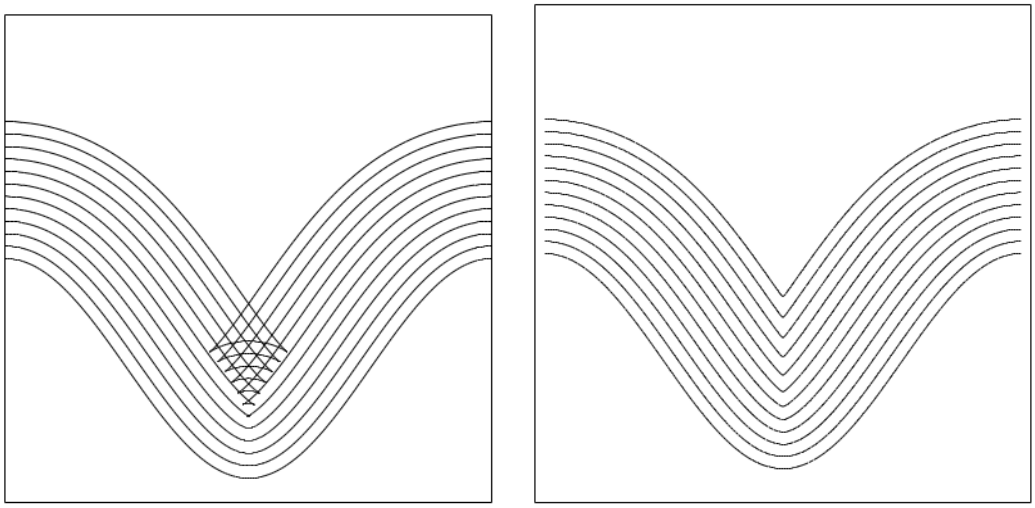
4.1.3 The Level-Set Method

This section will be completely dedicated to the introduction of Level-Set methods, already encountered in the previous section 4.1.2 where front-capturing most spread techniques for the localization of freely-moving interfaces were presented. Such methods, first proposed by S. Osher and J. A. Sethian with their most popular work [59], in 1988, allow especially

for the implicit study of moving bodies immersed in fluid domains, which is the case of our interest, but they are also employed in other propagating interfaces situations, such as for the modeling of ocean waves, burning flames evolution, and several different applications largely diffused in the literature. The following discussion will be based on the review work [74] again by Sethian, but many other introductions to the subject can be found all over the literature, being a widely spread field of research concerning FSI issues, to cite some, see [73, 43, 61, 40]. Then, we recall again the two fundamental works by the supervisors of this thesis at Inria (Bordeaux), in [15] and [14], where Level-Set methods are applied to the investigated problems.

Fundamentals of Front Evolution:

As specified by many authors, choosing an eulerian approaches for the description of moving interfaces and boundaries immersed in the physical domain instead of more traditional lagrangian methods leads to several undoubtable advantages concerning the handling of the complete devised model, among which one can find: first of all, and perhaps the most important aspect of this enumeration, the enhanced capacity of dealing with complex problems regarding front propagation, that can include topology changes (merger and breakage) within the domain, arising of singularities, discontinuities, respect of the weak (in the sense of differentiability) and entropic solutions criteria, etc.; then, the possibility of describing sharp corners and cusps characterising the moving surfaces is also guaranteed, the computation of geometrical properties such as normals and surface curvatures is enormously simplified, and not less relevant is their great adaptivity to parallelisation, when associated with huge simulations of complex 2D or 3D phenomena; finally, when some assumptions are possible to be made regarding front advancement, faster and more efficient methods have also been formulated. Concerning the mentioned issue of weak and entropic solutions, briefly, one could consider the well-known problem of flame propagation, thus assuming a cosine curve initial profile and letting it propagate normally to itself according to a lagrangian approach, irregularities soon appear in the solution, and those methods turn out to be not suitable for their physical description. What one would obtain by solving a standard lagrangian system of ODEs following each point along its trajectory would be the so-called *swallowtail* profile, whereas a weak and entropy-respecting solution could be carried out, on the other hand, by adopting an eulerian approach such as the here presented Level-Set method. Both the described situations are depicted in figure 4.11. In the previous example, we spoke about entropic solution in terms of the general entropy principle, according to which no new information can be created during the evolution of a problem. Keeping in mind the flame propagation example, *once a particle is burnt, it stays burnt*, meaning that once a corner, i.e. an irregularity, a shock, appears in the solution, the latter is no longer reversible, and it cannot any more be related to the original profile, thus determining an irreversible loss of information.



(a) *Lagrangian solution: swallowtail profiles.* (b) *Eulerian weak and entropic solution.*

Figure 4.11: Comparison between traditional lagragian and eulerian approaches to the flame propagation problem.

Time-Dependendent Level-Set Formulation:

Let us suppose to deal with a given geometry definition, with its external surface defined onto the domain. We can think of identifying that interface by means of a higher dimension function ϕ of space and time (say, a hypersurface if we are concerned with a 2D original problem), namely the level-Set function, whose 0-level corresponds exactly to the front we are interested in representing within our domain. Therefore, we could aim to study the evolution in time of such moving front by propagating that function, and by pursuing its 0-level as always connected the surface's shape of our deforming or moving immersed body. See figure 4.12 to have a simple example of a circular propagating surface. Thus, at

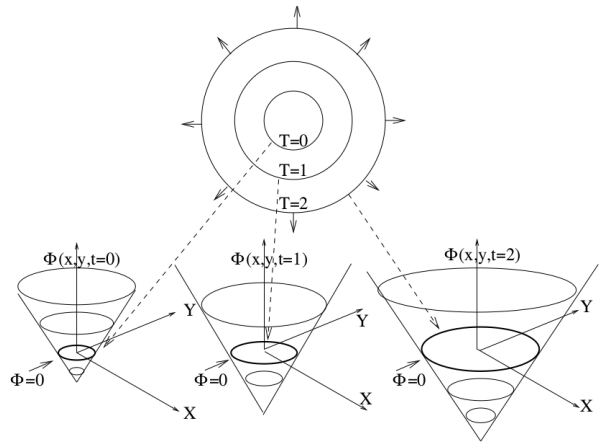


Figure 4.12: Initial value formulation's illustration of the Level-Set method.

each time instant, we could intercept the moving external surface Γ of a body immersed in a domain Ω as that defined as $\Gamma = \{\boldsymbol{x} \in \Omega : \phi(\boldsymbol{x}, t) = 0\}$. Then, by applying the chain rule to the definition of ϕ, which means by studying its evolution in time from an eulerian stand point, one obtains the following transport PDE:

$$\frac{\partial \phi}{\partial t}(\boldsymbol{x}(t), t) + \frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} \cdot \nabla \phi(\boldsymbol{x}(t), t) = 0 \tag{4.5}$$

Being then, in our case:

$$\frac{\mathrm{d}\boldsymbol{x}(t)}{\mathrm{d}t} = \boldsymbol{V}(\boldsymbol{x}, t)$$

We finally reach the desired form of the transport Level-Set function equation, defining the problem:

$$\begin{cases} \dfrac{\partial \phi}{\partial t} + \boldsymbol{V} \cdot \nabla \phi = 0 \\ \phi(\boldsymbol{x}, t = 0) = \phi_0 \end{cases} \tag{4.6}$$

Where therefore the 0-level set of the hypersurface ϕ is simply advected in an eulerian manner by the velocity field $\boldsymbol{V}$, solution to the Navier-Stokes problem, giving then rise to the coupling between the two approaches. Afterwards, the information coming from the new profile of the Level-Set, representing the external shape of the immersed bodies, will affect the solution of the fluid within Ω, and so all over again. Problem 4.6 is thus an initial value formulation of the Level-Set approach, but other strategies have also been developed in that sense, briefly reported in the following. To the aim of this work, equation 4.6 needs to be properly discretized and numerically solved in time, by exploiting the same schemes introduced in chapter 3, in particular WENO schemes presented in section 3.1.1 have been proposed in this work, with the purpose of minimizing the diffusive error introduced by the numerical approximation of the solution. Practically, one can formulate the following discrete problem referring to those schemes presented in the previous sections, and already recalled above:

$$\begin{cases} \dfrac{\phi_{i,j}^{k+1} - \phi_{i,j}^{k}}{\Delta t} + u_{i,j}^{k} \left(\dfrac{\delta \phi}{\delta x} \right)_{i,j}^{k} + v_{i,j}^{k} \left(\dfrac{\delta \phi}{\delta y} \right)_{i,j}^{k} = 0 \\ \phi_{i,j}^{0} = \left(\phi_0 \right)_{i,j} \end{cases} \tag{4.7}$$

Where $i \in (0, N_x - 1)$ and $j \in (0, N_y - 1)$, and in which, for the spatial approximated derivatives $\left(\frac{\delta \phi}{\delta x}\right)_{i,j}^{k}$ and $\left(\frac{\delta \phi}{\delta y}\right)_{i,j}^{k}$, one chooses the most suitable scheme among those presented in section 3.1.1 depending on whether the point (i, j) belongs to the boundaries $i = 0 \vee i = N_x - 1$ and $j = 0 \vee j = N_y - 1$, to the surrounding nearest cells, say $i < 3 \vee i > N_x - 4$ and $j < 3 \vee j > N_y - 4$, or to the remaining internal domain. Thus, we have the following distinction:

- (i, j) on the boundaries: first order unbalanced finite difference (scheme 3.1 or 3.2), for instance, considering the inflow section $i = 0$:

$$\left(\frac{\delta \phi}{\delta x} \right)_{0,j}^{k} = \frac{\phi_{1,j}^{k} - \phi_{0,j}^{k}}{\Delta x}$$

- (i, j) next to the boundaries: second order centered finite difference (scheme 3.3), for instance, considering cells with $i = 2$:

$$\left(\frac{\delta\phi}{\delta x}\right)^k_{2,j} = \frac{\phi^k_{3,j} - \phi^k_{1,j}}{2\Delta x}$$

- (i, j) internal: adoption of WENO schemes presented in section 3.1.1, that is schemes 3.30 and 3.31.

As it regards the definition of the initial condition ϕ_0, finally, as anticipated in the previous section 4.1.2 the most popular idea is that of choosing a distance-function definition, such as a simple euclidean distance one, so that each grid points will be associated then with a value of ϕ corresponding to its euclidean distance from the 0-level set, assuming as positive or negative those values associated with points respectively inside or outside the represented bodies, or vice versa. Thus, one can define (where d denotes the euclidean distance function, see figure 4.13 as a 1-dimensional example):

$$\phi_0(\boldsymbol{x}) = \pm d$$

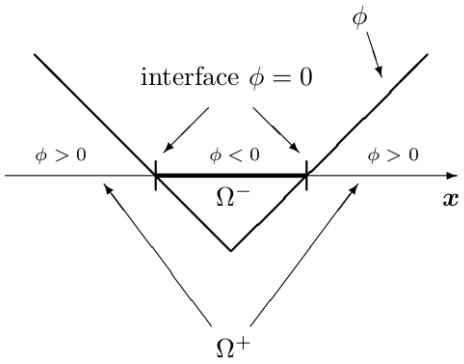


Figure 4.13: 1D representation of a Level-Set function defined as an euclidean distance function.

Stationary Level-Set Formulation:

If we consider that, on the one hand, the previous introduced formulation represents an initial value time-evolutionary problem, where the progressive position of the interface is always associated with the propagation of the 0-level set of the function $\phi(\boldsymbol{x}, t)$, exactly as it was a matter of a passive scalar transport problem, starting from some initial configuration $\phi_0(\boldsymbol{x})$, on the other hand a different formulation of the same evolutionary problem is possible to be devised. Indeed, by considering a front's propagation velocity keeping always the same sign, V_n, denoted that fashion since associated with the normal component of the front's propagation velocity, one could think of characterizing the

progressive time-evolutionary position of the moving surface by computing its arrival time for each grid point, that is a quantity $T(\boldsymbol{x})$, consisting of a boundary condition, and thus leading to an evolutionary problem such as:

$$\begin{cases} |\nabla T(\boldsymbol{x})| V_n = 1 \\ T(\boldsymbol{x} \in \Gamma(\boldsymbol{x}, t = 0)) = 0 \end{cases} \tag{4.8}$$

In figure 4.14 a schematic explanation is given for this different approach, which is nothing but a boundary value problem, contrarily to the previous initial value one. Anyway, we will not be concerned with it, since in our problem the front propagation velocity is always subjected to continuous changes of its value, and thus we will not focus more on the topic.

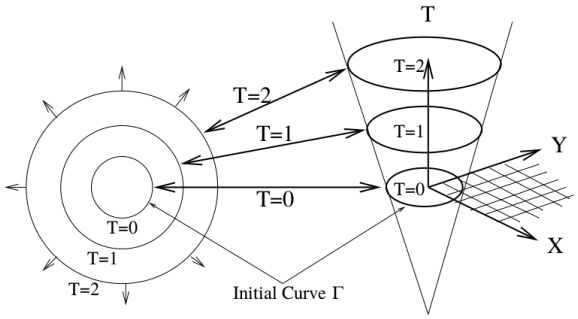


Figure 4.14: Boundary value formulation's illustration of the Level-Set method.

Some Proposed Improvements and Adaptations:

As mentioned, a lot of work has been done over the development of more efficient, fast and reliable techniques involving Level-Set functions as tools for pursuing freely-moving interfaces within a fluid domain. In particular, two improved methods have been proposed in order to provide a significant speedup for both the time-dependent and the stationary formulations presented above. Those innovative approaches are known as the *Narrow-Band* Level-Set scheme, for the former, and the *Fast-Marching Method* for the latter. Again, since we will not deal with the boundary value formulation of the problem, i.e. with the stationary one, we will leave alone the fast marching methods associated with them, but detailed information can be found in all those works cited above. As it regards the so-called narrow-band scheme developed for the initial value formulation, we will provide in the following just a brief introduction to it (see the same works for further details).

This method was first proposed by D. Adalsteinsson and Sethian in 1994 (see [8]), and it supposes to consider only a narrow band of cells in the neighbourhood of the moving interface for the computation of its displacement according to the procedure described previously. This approach could obviously lead to a huge save of computational resources and efforts, since we are no longer compelled to execute the transport algorithm over the whole domain grid, especially when 3-dimensional problems are considered. On the other

hand, such method would require the same implementation difficulties as a lagrangian tracker one, but it will maintain the advantages connected to the usage of an implicit eulerian representation approach, discussed in the introduction to this section. Some illustrations are provided here below in figures 4.15 and 4.16.

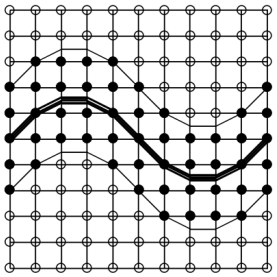
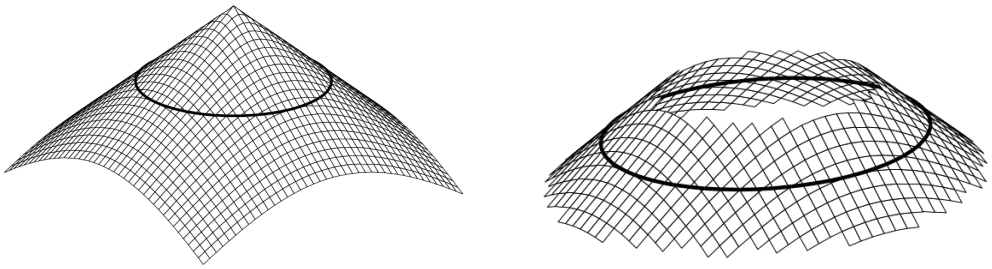


Figure 4.15: Narrow-band basic conceptual illustration.



(a) *Distance function over the whole domain.*

(b) *Distance function over a narrow-band portion of the domain.*

Figure 4.16: Comparison between the signed distance function ϕ, the Level-Set function, with and without the adoption of a narrow-band approach.

Reinitialization:

Resuming the initial value, or time-dependent, formulation of the Level-Set method, unfortunately, as pointed out by many authors in the literature, as we make the scalar function transport by the fluid within the domain, by solving the associated convection equation 4.6 step by step, exploiting the introduced discrete scheme 4.7, we would notice that the appreciated properties associated with the euclidean distance function adopted as first initialization of ϕ are progressively lost. It is fairly straightforward indeed to observe that $|\nabla\phi| = 1$ for any euclidean distance function, a desirable property which will be completely altered as the numerical solution advances in time, being thus no longer true. These occasionally large variations in the gradient of the Level-Set could lead to the arising of instabilities which will then cause the failure of the method, making them clearly necessary to have a smooth gradient behaviour all over the represented surface, with as less oscillations as possible for $|\nabla\phi|$. For that reason, proper *reinitialization* algorithms

have been formulated in order to restore the euclidean distance properties of ϕ, meaning then in order to ensure $|\nabla\phi| = 1$ again.

Nevertheless, reinitialization does not exclude some risks, connected to the fact that if after each time integration such a procedure is performed, the represented surface changes somehow its shape, being affected by the numerical method applied in order to restore its euclidean distance function features, that is by the artificial viscosity affecting the solution and always included in stabilized methods. Thus, an acceptable and convenient trade-off needs to be devised and performed.

To the aim of implementing reinitialization, different methods already exist in the literature, and the most popular ones are that involving the resolution of a specific PDE, called for this reason *reinitialization equation*, until reaching its stationary state, or alternatively the already encountered fast-marching method, which we will leave alone, as said. Concerning the reinitialization equation then, it basically consists of the enforcement of the euclidean distance function property previously stated, according to which $|\nabla\phi| = 1$, on the current Level-Set function ϕ_R (R meaning *to reinitialize*). Therefore, a time-dependent PDE could be written down as:

$$\frac{\partial\phi}{\partial\tau} = \text{sign}(\phi_R)(1 - |\nabla\phi|) \tag{4.9}$$

Where a non-physical time variable τ has been introduced, having nothing to do with the actual time of the problem t, and where the sign(.) function could be formulated as:

$$\text{sign}(\phi_R) = \frac{\phi_R}{\sqrt{\phi_R^2 + \varepsilon^2}}$$

Where for the quantity ε one usually adopts the grid spacing Δx, as done in the present work. Two additional considerations are worth making at this point: first, in order to avoid instabilities connected to the numerical method associated with the resolution of equation 4.9, we had better stop the procedure after a few steps, which will turn out to be sufficient if the initial function ϕ_R to reinitialize is already close enough to a signed-distance function; then, we do not need to perform the reinitialization over the whole domain, but only in those regions next to the interface, i.e. the 0-level set to reinitialize, exploiting the same ideas proposed by narrow band methods. An excellent overview on reinitialization techniques, involving their pros and cons, and showing in a clear way what are the main consequences deriving by a too huge amount of reinitialization processes (as well as by a too reduced one), is reported in [45].

In our work, reinitialization is therefore performed after each certain number of time iterations, decided a priori, following a step-by-step really simple procedure described below:

1. definition of the $\text{sign}(\phi_R(x, y))$ map all over the domain $(0, N_x - 1) \times (0, N_y - 1)$, that is:

$$\text{sign}(\phi_R)_{i,j} = \frac{\phi_{R_{i,j}}}{\sqrt{\phi_{R_{i,j}}^2 + \varepsilon^2}}$$

2. construction of the term $G = |\nabla\phi| - 1$ of 4.9 associated with each grid point (i, j), in the spirit of upwind schemes, thus:

$$(d_x^+)_{i,j} = \frac{\phi_{i+1,j} - \phi_{i,j}}{\Delta x}, \qquad (d_x^-)_{i,j} = \frac{\phi_{i,j} - \phi_{i-1,j}}{\Delta x}$$
$$(d_y^+)_{i,j} = \frac{\phi_{i,j+1} - \phi_{i,j}}{\Delta y}, \qquad (d_y^-)_{i,j} = \frac{\phi_{i,j} - \phi_{i,j-1}}{\Delta y}$$

$$\begin{cases} G_{i,j} = \left(\sqrt{\max\left(\max(d_x^-,0)^2, \min(d_x^+,0)^2\right) + \max\left(\max(d_y^-,0)^2, \min(d_y^+,0)^2\right)} - 1 \right)_{i,j} \\ \phi_{i,j} > 0 \end{cases}$$

$$\begin{cases} G_{i,j} = \left(\sqrt{\max\left(\min(d_x^-,0)^2, \max(d_x^+,0)^2\right) + \max\left(\min(d_y^-,0)^2, \max(d_y^+,0)^2\right)} - 1 \right)_{i,j} \\ \phi_{i,j} < 0 \end{cases}$$

3. performation of the temporal iteration over τ, according again for simplicity to the explicit Euler scheme 3.16:

$$\phi_{i,j}^{h+1} = \phi_{i,j}^h - \Delta\tau \,\mathrm{sign}(\phi_R)_{i,j}\, G_{i,j} \tag{4.10}$$

Where typically one chooses $\Delta\tau = \mathcal{O}(\Delta x)$.

The present algorithm is therefore performed five times before starting the time iteration of the actual Navier-Stokes problem, in order to make the initial Level-Set function equal to signed distance function, afterwards it will be still executed once every given number of physical time iterations.

To conclude this section dedicated to freely-moving interfaces localization techniques, we propose below, in figure 4.17, an extraordinary applied example of a Level-Set function approach on a drop break-up due to an impulsive acceleration, improved with a refinement strategy next to the investigated deforming surface.

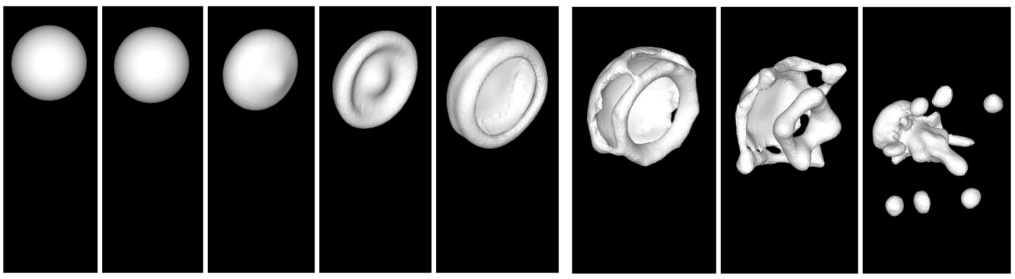


Figure 4.17: Drop break-up visualization obtained thanks to a Level-Set method improved with grid refinement next to the interfaces (reference in [43]).

4.2 Imposition of Implicit Boundary Conditions

Now that we have addressed the problem of moving interfaces within a fluid domain, and presented the large number of methods already developed to the aim of intercepting them, we must provide a way to impose case-specific boundary conditions onto them, in order to allow for the actual and realistic fluid structure interaction dynamics to be finally taken into consideration. Doing that means, briefly, to be capable of quantifying the effect of the fluid on the deforming structures, or on the moving bodies immersed in the physical domain, and also the contrary, that is studying how the motion of immersed boundaries affects the fluid solution within the subsequent time steps. Since now we dispose of an implicit representation of the moving surfaces onto where we would like to apply such boundary conditions, we could exploit the same philosophy and keep an eulerian point of view in developing this next step of our method, thus referring to those commonly defined in the literature as *implicit boundary conditions*. Substantially, we can classify these classes of methods into two huge categories:

- **Immersed Boundary Methods:** the most interesting considering our final goal, and indeed including also the method adopted for this work. We had already mentioned this approach while introducing interface localization tools in section 4.1, but now we will focus more deeply on its exploitation for the imposition of boundary conditions on immersed moving, elastic or rigid bodies, rather than for their mere interception;

- **Ghost Fluid Methods:** a different alternative approach which will be briefly introduced at the end of this section for completeness;

Starting with the first class listed above, what it follows is directly taken from a very excellent review article on immersed boundary methods carried out by R. Mittal and G. Iaccarino, in 2005, reported in [53], but it is fundamentally inspired to the very first work concerning the adoption of such implicit strategies with the aim of studying the cardiac dynamics of the heart and all its valves, due to C. S Peskin in 1972, that can be found in [62]. Since then, many other similar works have been proposed by authors, all employing almost the same strategies presented in that first publication, but applying them to the most various fields of research, from biofluid dynamics to aerodynamics, see for example contents of [72, 9].

Basically, the following methods will all be based on a domain configuration thought as an omni-comprehensive region Ω, composed of a certain number of solid rigid or elastic bodies denoted through Ω_s^i and immersed in a global fluid medium Ω_f. The situation is schematically reported in figure 4.18. A common important feature which highlights once again the reason why we are interested in such innovative methods is the fact that they are based primarily on cartesian orthogonal grids, thus with extremely simple meshes not body-fitting and not moving in time, crossing all immersed bodies represented via the implicit techniques previously introduced. Therefore, the term *immersed boundary* (hereafter IB) denotes all those approaches devoted to the imposition of boundary conditions on those bodies freely-moving within a viscous flow domain, resorting to a simple cartesian grid which does not fit their shape. Evidently, this discussion could also be extended

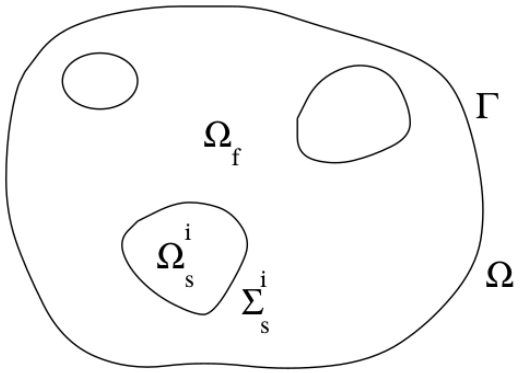


Figure 4.18: Reference domain involved in the implicit boundary conditions imposition.

to multi-phase fluid flows. Imposing boundary conditions according to an IB approach on a cartesian and fixed grid cutting through the body surfaces, thus, means that some modifications need to be added to the fluid governing equations in order to account for the presence of these objects, somehow, and at least in those regions in the proximity of them. However, one should admit that this procedure, in turn, could compromise conservation properties of the adopted scheme, as well as accuracy, especially when high Reynolds numbers occur, and therefore strong solution gradients within boundary layers next to the walls occur, not disposing this way of a body-conforming grid which could be properly refined with the aim of well describing those quick and confined transition regions. For that reason, a global cartesian grid needs to be generated with a high enough resolution over the whole domain, increasing then the total computational effort required, but enabling for a sufficiently accurate computation of such solutions.

By keeping for the moment a general point of view, the imposition of IB conditions, as mentioned, involves the introduction of a source term either in the continuous fluid model equations 2.27 or in their discrete form 3.43 (here the discrete formulation of the prediction step equation of the FSM has been considered, presented in section 3.3). This term could be thought as a forcing function capable of taking into account the effect of the body's boundary on the fluid, in a fashion that we will discuss soon in the following, focusing on some peculiar examples. Now, depending on whether this forcing contribution is introduced in the continuous formulation or in the discretized one, we will be concerned either with the so-called *continuous forcing methods*, or with the *discrete forcing* ones.

4.2.1 IB: Continuous Forcing Methods

To these methods belongs also the very first one proposed by Peskin in [62], aimed at simulating the cardiac hemodynamics, as anticipated. This is why, then, such approach is more suitable for the application to elastic bodies rather than in the case of rigid ones, but some extensions were proposed in that sense. Since the original version of the method was devised in order to account for the interaction between elastic cardiac walls, related to heart muscle contraction, and blood flow within each of its chambers and through its valves, those boundaries represented by cardiac elastic fibers were tracked in a lagrangian

fashion, as presented in the previous section 4.1, by employing lagrangian markers, whose positions are determined by solving the same problem 4.1, that is:

$$\begin{cases} \dfrac{\mathrm{d}\boldsymbol{X}_k}{\mathrm{d}t} = \boldsymbol{V}(\boldsymbol{X}_k, t) \\ \boldsymbol{X}_k(0) = \boldsymbol{X}_{k,0} \end{cases}$$

Once known the trajectory of each lagrangian marker describing the behaviour of the immersed boundary in time, following this mixed lagrangian-eulerian method, the effect exerted by these fibers on the fluid needs to be represented: this can be done by adding a source term to the governing equation, a forcing contribute linked to the elastic deformation of such fibers, through some expression of the well-known Hooke's law, which could read, for example:

$$\boldsymbol{f}(\boldsymbol{x}, t) = \sum_k \boldsymbol{F}_k(t)\delta\big(|\boldsymbol{x} - \boldsymbol{X}_k|\big) \tag{4.11}$$

This last forcing contribute $\boldsymbol{f}$ must then be added to the continuous formulation of the incompressible Navier-Stokes equations 2.27, and it will then turn out to be effective only when $\boldsymbol{x} = \boldsymbol{X}_k$ of the k-th lagrangian marker, thanks to the Dirac's function properties, and therefore only when the immersed body is encountered in the domain.

When rigid body's boundaries are the case instead, some extensions are possible, even though they are not that straightforward. Indeed, considering the rigid limit of the previous formulation is not the most desirable situation, causing the problem to become more and more ill-posed. On the other hand, one could still consider an elastic but extremely stiff body, or, furthermore, as it was connected by a spring to an equilibrium point, whose forcing restoring term would thus be:

$$\boldsymbol{F}_k(t) = -k_{el}\big(\boldsymbol{X}_k - \boldsymbol{X}_k^{eq}(t)\big)$$

Where k_{el} is the spring's elastic constant, whereas $\boldsymbol{X}_k^{eq}$ is the k-th lagrangian point's equilibrium position. Because of the large values of k_{el} required by this approach, it is considered to be inefficient and to give rise to a stiff system of equations. Anyway, a different solution proposed is based on the following expression for the forcing term:

$$\boldsymbol{f}(t) = \alpha \int_0^t \boldsymbol{V}(\tau)\, d\tau + \beta \boldsymbol{V}$$

The above reported equation, in other words, represents a damped oscillator model, and we decided to mention it because, by choosing $\alpha = 0$ and $\beta = \frac{\chi}{\lambda}$, where χ is a characteristic function equal to 1 inside an immersed body and null elsewhere, whereas λ is a permeability constant associated with a global porous medium (typically $\lambda << 1$) which now constitutes the whole fluid and immersed bodies domain $\Omega = \Omega_f \bigcup_i \Omega_s^i$, we finally find the IB scheme adopted in the current work, called *Penalisation method*. The latter aims to consider the whole domain as a unique continuous material, characterized by regions exhibiting different values of permeability to the fluid, through the combined term $\frac{\chi}{\lambda}$, allowing thus for the system of governing equations to be solved once and for all with no respect for the explicit treatment of the immersed boundaries, since all the necessary associated

information are carried by that forcing term. Such a method, in addition, was used also by the supervisors of this work in their already cited publications [15] and [14], where they substantially modified the continuous formulation of the incompressible momentum equation so that:

$$\frac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{V} + \sum_{i=1}^{N_b} \frac{\chi_i}{\lambda}(\boldsymbol{V}_{b,i} - \boldsymbol{V}) \tag{4.12}$$

Where N_b denotes the total number of immersed bodies, and $\boldsymbol{V}_{b,i}$ is the moving body velocity associated with each of them. Clearly, considering what stated above, when we consider a point $\boldsymbol{x}$ not belonging to any immersed body, thus lying in a portion of the continuous porous medium characterized by high fluid permeability , then all $\chi_i = 0$, and the previous equation reads exactly like the traditional momentum balance equation:

$$\frac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V} = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{V}, \quad \text{if} \quad \chi_i = 0, \forall i$$

Conversely, whenever one of the immersed body is encountered, its related $\chi_i = 1$, and being $\lambda << 1$ as stated above, the other terms turn out to be completely negligible with respect to the new penalisation one. Therefore, the above model equation simply reduces to:

$$\boldsymbol{V} = \boldsymbol{V}_{b,i} \quad \text{if} \quad \chi_i = 1$$

This way, one can implicitly impose IB conditions on a specified number of immersed bodies freely-moving within the fluid domain. As anticipated, although these methods are really fascinating because of their great ductility and adaptivity, they are not well suited for the representation of sharp velocity gradients associated with high Reynold's number flows; in addition, the motion equations need to be solved also inside the immersed bodies, thus leading to a waste of computational efforts.

One last observation is worth making concerning this approach: here we described a situation in which the imposed penalised velocity $\boldsymbol{V}_{b,i}$ associated with the i-th was known a priori, say by a desired motion law, or whatever one could think of applying to a moving immersed body. However, in the already cited work proposed by M. Bergman, reported in [15], the author shows how it is possible to reproduce also self-propulsion, by exploiting such penalisation approach in accordance with a Newton's motion law for the computation of the body's velocity. That way, those penalised velocities will not be imposed, but, on the contrary, they will depend on the fluid action itself.

As it regards our work, more specifically, we adopted penalisation in order to impose IB conditions on the external aortic walls (upper and lower domain frontiers, where a no-slip condition is applied, thus $\boldsymbol{V} = 0$) and, if necessary, on some objects used to hold the dissection membrane, the floating tunica intima, thus called *holders* in the following, by adopting some proper motion law (e.g. harmonic) or by keeping them fixed too.

4.2.2 IB: Discrete Forcing Methods

Differently to the previous category, this class of IB methods, as anticipated, aims to introduce a forcing term directly in the already discretized governing equations, say those

associated with the prediction step of the FSM in our discussion over numerical methods (see section 3.3.2, scheme 3.43). For the purpose of the following contents, we now recall here a general space-discretized form of the momentum balance equation:

$$\left(\frac{\mathrm{d}\boldsymbol{V}}{\mathrm{d}t}\right)_{i,j} = -\boldsymbol{V}_{i,j}\cdot\left(\nabla_\delta \boldsymbol{V}\right)_{i,j} - \left(\nabla_\delta p\right)_{i,j} + \frac{1}{Re}\left(\nabla_\delta^2 \boldsymbol{V}\right)_{i,j} \tag{4.13}$$

Such methods turn out to be very useful when high resolution is needed, for example, in the case of those problems characterized by a high Reynold's number, where strong gradients of the solution field appear next to the immersed surfaces and boundaries. Contrarily to continuous forcing methods, which were responsible for a certain degree of spreading of the solution in the proximity of the immersed surfaces where boundary conditions were applied, a sort of numerical error due to the way the forcing term is introduced into the model, and connected to how it operates in order to affect the solution field in those regions where its selective term is activated (Dirac function, variable permeability, etc.), discrete forcing methods operates only by modifying directly the discretized equations, through the modification of the involved computational stencil, for example, in order to achieve the same goal of imposing IB conditions, but keeping a high level of local accuracy. Nevertheless, those methods could be much more hard-implementing, and, almost always, they lead to the well-known problem of *fresh cells*, presented afterwards. Among all discrete forcing-like methods, we primarily cite the following two approaches:

- **Ghost-cell finite difference approach:** concerning this method, again the work in [14] by M. Bergmann and A. Iollo, advisors of this thesis, turns out to be very enlightening, since it presents the *image point corrector* (IPC) technique based on the ghost-cell approach we are going to describe here. Further details can be found in their cited publication, but to briefly introduce the subject, basically one aims to improve the accuracy of the classic penalisation method previously presented, which is said to be 1st order accurate in space by the authors, up to the 2nd order of accuracy, by computing a correction for the velocity values of those solid cells which have at least one fluid neighbouring cell. The so corrected solid cell will therefore be named as *ghost-cells*, since they will lose their physical meaning. This way, one allows for a better representation of the velocity gradient next to the solid surface, since that velocity correction is actually computed by considering the correct desired value on the interface, that is the penalisation velocity $\boldsymbol{V}_b$. The value of the corrective velocity imposed on the ghost cell is computed exploiting a polynomial reconstruction of the velocity gradient, based on the solution field evaluated in the surrounding points (usually 3 in the fluid, and one on the boundary, obtained projecting the ghost point onto it, as shown in figure 4.19 where F_1, F_2 and F_3 are reference points belonging to the fluid, whereas B_2 is the one projected on the boundary, or, alternatively, one can consider B_1 taken as the midpoint between P_1 and P_2, orthogonal projection of the ghost point onto the boundary). While IPC is applied only on ghost cells, i.e. those with at least one neighbouring fluid cell, classic 1st order penalisation is imposed on all other solid cells.

- **Cut-cell finite volume approach:** this second case is based on the idea of restoring the conservation properties of finite volume schemes, a desirable condition

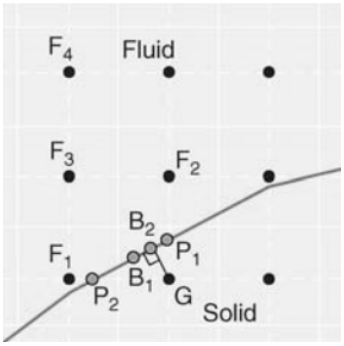


Figure 4.19: Correction ghost-cell velocity's reconstruction scheme.

which, unfortunately, had been lost when adopting IB methods to impose boundary conditions. By considering to deal again with a regular cartesian grid, such a methodology aims to individuate the intersection between the boundary and each cut grid cell, and then to build control volumes to be used in a finite volume formulation by selecting those cells whose centers belong to the fluid, eliminating their portion cut by the crossing solid boundary, as depicted in figure 4.20. Then, in order to proceed with a finite volume solver for the Navier-Stokes equations, mass, convective and diffusive fluxes have to be estimated on each cut-cell interface, as well as pressure gradients, and this can be done again through some proper polynomial interpolation approach (more details in [53]). Even this second discrete forcing approach is up to 2nd order accurate in space, as the ghost-cell one, and, additionally, it succeeds in satisfying conservation properties of mass and momentum.

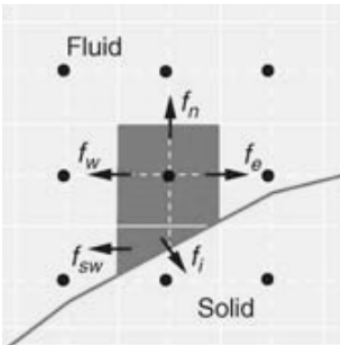


Figure 4.20: Cut-cell control volume scheme.

As previously anticipated, before concluding this section dedicated to immersed boundary methods and their general formulation, an issue regarding the specific case of freely-moving immersed bodies within a fluid medium needs to be addressed. We therefore aim now to introduce the problem of *fresh cells*, that is those grid points which belonged to the solid body up to the current time step, and thus involved in a discrete forcing IB approach such as ghost-cell or cut-cell just introduced, but which turn out to become fluid as the moving boundary crosses them (figure 4.21). Evidently, there does not exist any valid time

history for those points, since their behaviour is completely distorted by the IB method applied: a ghost cell has no physical meaning other than its purpose of ensuring the desired velocity gradient next to the boundary. Fortunately, this issue can be circumvented by adopting, in combination with a discrete forcing approach such as a ghost-cell one, also a continuous forcing method: if we introduce a 1st order penalisation for the same problem, then, whenever the moving boundary crosses the ghost point leaving it in the fluid medium, the natural smoothing previously discussed and always associated with that kind of penalisation, involving all grid cells next to the boundary, will restore a reasonable value of the solution on that point.

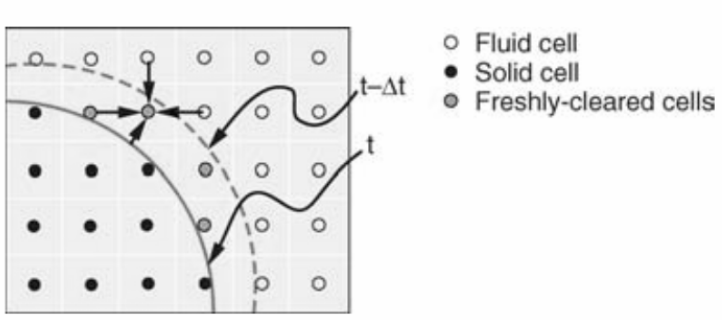


Figure 4.21: Illustration of the fresh-cells problem.

4.2.3 Ghost Fluid Method

Beside those belonging to the widely discussed family of Immersed Boundary methods, or IB, as we have always referred to, there exist a huge number of alternative techniques aimed to impose boundary conditions on floating and moving bodies immersed in a fluid domain in an implicit fashion, thus according to a substantially eulerian point of view. The one we would like to introduce within this section, as anticipated above, is typically known in the literature as the *Ghost Fluid Method* (GFM), and it could represent a valid opportunity when one intends to perform simulations including moving boundaries or multi-phase flows, especially in the case of inviscid flows, such as when Euler's equations are assumed as the mathematical model for the fluid motion description. The following given presentation will only be a brief introduction to the topic, for the sake of completeness on implicit imposition of boundary conditions, but no numerical treatment will be addressed specifically, since we will not be concerned with this kind of method for the aims or our work. These information are taken from [31], to cite one, but a large amount of different examples can be easily found. As anticipated, this method was originally devised for inviscid problems, and could therefore deal with interface discontinuities, contact surfaces and so on. Normally, when Euler's equations are considered, the inviscid limit of Navier-Stokes equation, formally corresponding to the case for which $Re \to \infty$, there no-longer exist no-slip conditions for the velocity field, when immersed walls are investigated. We rather speak about tangent condition, on the velocity component normal to the interface, $\boldsymbol{V} \cdot \boldsymbol{n}$. For this reason, first we will provide an inviscid version of the method, as it was originally thought, and then a viscous extension will also be shortly presented.

Basically, the idea is to consider the two fluids composing a two-phase inviscid flow problem as belonging to two different fields, and for both of which the corresponding govern equations are independently solved. When considering the interface between them, then, continuity condition is applied for those variable needing it, such as normal velocity components, pressure, etc., whereas a different approach has to be adopted to deal with discontinuous variables (entropy, temperature, tangential velocity components, etc.), by imposing the well-known *Rankine-Hugoniot jump conditions* in an implicit fashion, as explained in the following. These methods work at their best when combined with a reliable interface tracker, in the most general sense of the term, chosen among those presented in section 4.1, for instance. Here, as done by the authors of the cited publication, we will deal with a Level-Set approach, capable of the describing implicitly the motion of free interfaces within the entire fluid domain, taking into account also topology changes, merging and separation of bodies, etc. Then, thanks to the information carried by the Level-Set function, cells containing each fluid can be recognized and thus distinguished. Let us suppose, to begin with, to deal with a two-phase flow case, where each grid cell will therefore be occupied either by the fluid 1, or by the fluid 2, and whose identity can be found by looking at the sign of the associated Level-Set function, say, if positive we have fluid 1, otherwise we have fluid 2. At this point, a fluid state is defined in all cells, all over the grid. Now, according to the traditional GFM, a ghost fluid is also defined and associated with each grid cell in a way such that within each point it will be as two distinct fluids could exist at the same time, but independently one each other. This way, each grid point will contain the information concerning mass, momentum and energy associated with the real fluid which is actually hosted there, and those concerning also the second fluid, therefore named as the ghost fluid, which will be exploited to compute its evolution and its interaction with the real one next to the interface separating them, depending on the sign of the Level-Set function. Two different sets of Euler's equations need to be solved at each grid point, after each time step, and only the solution corresponding to the real fluid lying in each cell, accordingly to the sign of the Level-Set function, will be taken into consideration to show the final time solution.

More attention must be reserved to interfaces treatment, where the interaction between the two fluids needs to be taken into consideration. We thus distinguish two different ways to proceed, depending on whether we are dealing with a continuous variable or with a discontinuous one, across the surface:

- continuous variables: their value is simply copied from one fluid's cell to the adjacent second's one, since they are conserved through the interface. To this case belong the pressure and the normal component of the two fluids' velocity;

- discontinuous variables: this second case is more complicated, and it needs a special treatment concerning numerical extrapolation of one fluid's cell values towards the other's, through the interface, in order to populate at least a thin layer of ghost fluid's cells and allow then for a correct computation of these variables' evolution, avoiding their numerical smearing. Such procedure can be performed by computing the unit normal vector on the interface, exploiting once again the Level-Set function

as:

$$\boldsymbol{n} = \frac{\nabla\phi}{|\nabla\phi|}$$

And then, by solving a one-sided transport equation for the generic variable (.):

$$\frac{\partial(.)}{\partial t} \pm \boldsymbol{n}\nabla(.) = 0$$

Where the sign $\pm$ is chosen depending on which side of the fluid needs to be extrapolated towards the other one, according to the Level-Set function and thus to the resulting direction of the unit normal vector $\boldsymbol{n}$. All remaining variables belong to this category.

To generalize the method and thus take into consideration also moving interfaces, at speed denoted by $\mathcal{V}$ directed normally to the surface itself, along $\boldsymbol{n}$, we have to introduce fluxes of the involved variables, and then understand which of these are conserved or not across the interface. By calling F_ρ, $\boldsymbol{F}_{\rho\boldsymbol{V}} = (F_{\rho u}, F_{\rho v}, F_{\rho w})^t$ and $F_{\rho E}$, respectively, the mass, momentum and energy (E meant as total energy per unit mass, see section 2.2) fluxes, we can therefore write down the Rankine-Hugoniot jump conditions by assuming to consider the relative speed $\boldsymbol{V}\cdot\boldsymbol{n} - \mathcal{V}$ with respect to the moving interface, thus:

$$\begin{cases} \rho^G(\boldsymbol{V}^G\cdot\boldsymbol{n} - \mathcal{V}) = \rho^R(\boldsymbol{V}^R\cdot\boldsymbol{n} - \mathcal{V}) \\ \\ \rho^G(\boldsymbol{V}^G - \mathcal{V}\boldsymbol{n})(\boldsymbol{V}^G\cdot\boldsymbol{n} - \mathcal{V}) + p^G\boldsymbol{I}\cdot\boldsymbol{n} = \rho^R(\boldsymbol{V}^R - \mathcal{V}\boldsymbol{n})(\boldsymbol{V}^R\cdot\boldsymbol{n} - \mathcal{V}) + p^R\boldsymbol{I}\cdot\boldsymbol{n} \\ \\ \left(\rho^G e^G + \rho^G\dfrac{|\boldsymbol{V}^G - \mathcal{V}\boldsymbol{n}|^2}{2}\right)(\boldsymbol{V}^G\cdot\boldsymbol{n} - \mathcal{V}) + (\boldsymbol{V}^G\cdot\boldsymbol{n} - \mathcal{V})p^G\boldsymbol{I}\cdot\boldsymbol{n} = \\ \qquad = \left(\rho^R e^R + \rho^R\dfrac{|\boldsymbol{V}^R - \mathcal{V}\boldsymbol{n}|^2}{2}\right)(\boldsymbol{V}^R\cdot\boldsymbol{n} - \mathcal{V}) + (\boldsymbol{V}^R\cdot\boldsymbol{n} - \mathcal{V})p^R\boldsymbol{I}\cdot\boldsymbol{n} \end{cases}$$

Where the superscript R holds for the real fluid, whereas G refers to the ghost fluid, within each cell. This way, a system of equations is solved for each cell next to the interface, and the ghost fluid region is properly populated. At this point, if the interface velocity $\mathcal{V}$ is nothing but the normal fluid velocity (which is the same, as stated above, for both the fluids), meaning that we are dealing with a simple and non-reacting case of two-phase flow (and not, for instance, with a shock-wave separating to fluids characterized by different properties, another interesting potential application of GFM as reported in [31]), then the equations derived above reduce to conservation of normal velocity components and of pressure, leaving us with the extrapolating choice previously introduced for the non-continuous variables such as entropy and tangential velocity components, having some degrees of freedom concerning those quantities.

Before concluding this part, we introduce also the viscous contribution to extend the GFM to the complete Navier-Stokes equations. In order to define the shear stress tensor $\boldsymbol{\tau}$ (see section 2.2), we first need to compute all spatial derivatives over the fluid domain, for both the ghost and the real fluids. This surely means that special care must be reserved, once again, to those points next to the interface. However, since continuous solutions are

acceptable wherever no-slip conditions are imposed between the two fluids, thus leading to a smeared profile linking them two, a 2nd order centered scheme is applicable everywhere, with no care for those stencils crossing the interface itself. Such terms, finally, could be added to the fluxes F_ρ, $\boldsymbol{F}_{\rho\boldsymbol{V}}$ and $F_{\rho E}$ to complete the Rankine-Hugoniot system. Note that even the shear stress tensor deserves the same treatment as the other variables, being composed of a continuous and a discontinuous part, such that $\boldsymbol{\tau} = \boldsymbol{\tau}_C + \boldsymbol{\tau}_D$, for which the procedures stated above still hold.

4.3 Fully-Eulerian Monolithic Approach

This last section of the chapter is crucial for the purpose of the present work. In it they are contained all those choices and approaches which most characterise the adopted methodology, giving also the title to this master's thesis work. Actually, we have already discussed almost all the ingredients needed to conclude the set up of our computational problem, and not much has still to be said, but within this final theoretical part we especially aim to summarize what has been stated in the previous chapters, recalling all those choices being part of the following approach to the problem, merging them all together including some last notions, and consequently introducing the global definition of *Monolithic Approach*. We had already anticipated the difference between eulerian and lagrangian points of view, when the introduction of interface localization was the case, the former being associated with the study of tracking particles by means of computing their trajectories, driven by the fluid field itself, whereas the latter rather attempted to capture them implicitly, exploiting auxiliary fields or functions. This second approach allows for the adoption of fixed meshes, not necessarily body-fitting, where all immersed objects could move and deform freely, crossing all grid cells with no care for the correct explicit definition of their fluid-structure interfaces. Then, Immersed Boundary methods have also been proposed to the end of imposing boundary conditions on such floating bodies, again implicitly, oftentimes exploiting right the information provided by an interface localizer belonging to the aforementioned category. This way, the fluid can be described in a complete eulerian fashion, as well as the imposition of immersed boundary conditions on bodies contained within the domain is concerned, but in order to name our method as a *fully eulerian* one, also the deforming structure of elastic and visco-elastic bodies has to be described similarly. Once this is done too, our problem will be said as approached according to fully-eulerian fashion, furthermore consisting of dealing with a globally unique material showing different levels of porosity, or permeability, depending on the region we are looking at (i.e. on the presence of immersed floating bodies, and therefore on the information coming from the interfaces localizer), and thus tackled in such a way that the same govern equations are solved on each grid point independently of its nature, its position or whatever feature it should represent, being affected by penalisation terms where moving bodies equipped with such immersed boundary conditions are encountered, or accounting for visco-elastic effects whenever deforming objects interact with the fluid. A so described approach would therefore be named as *monolithic*, since it would be able to deal with the entire domain as it was composed of a unique medium, regardless of what would be contained in it, of how it would be moving and of what should be the best

explicit model to associate with each single grid point. Some detailed information about this aspect could be found, for example, in another work by prof. A. Iollo, but applied to compressible flows, in [29].

In the following, we first introduce the eulerian description of a fluid structure interaction (FSI) model, including hyperelastic materials immersed in an incompressible fluid domain, and then, finally, we resume all the presented techniques adopted to achieve the goal of the present master's thesis work.

4.3.1 Fully-Eulerian FSI Model

As shown in section 2.4, we can describe the displacement of any particle contained in our domain $\Omega = \Omega_f \cup \Omega_s$, the union of a fluid and a (or more) solid rigid or elastic immersed body, following a classic lagrangian approach, according to which each final position $\boldsymbol{x}(\boldsymbol{x}_0, t) \in \Omega \times (0, T)$, is related to its initial position $\boldsymbol{x}_0$ through the so-called *direct characteristic function*, or *forward characteristics*, $\boldsymbol{X}(t, \boldsymbol{x}_0)$, and thus defining the problem:

$$\begin{cases} \dfrac{\partial \boldsymbol{X}}{\partial t} = \boldsymbol{V}(t, \boldsymbol{x}) \\ \boldsymbol{X}(t = 0, \boldsymbol{x}_0) = \boldsymbol{x}_0 \end{cases} \tag{4.14}$$

Which, in other words, consists of following each particle along its trajectory, as already seen in the case of lagrangian markers, governed by problem 4.1. Conversely, if we suppose to study a fixed point of the domain $\boldsymbol{x} \in \Omega \times (0, T)$ by observing all those particles crossing it, attempting then to recover each initial position $\boldsymbol{x}_0$ by following the so-called *indirect characteristic function*, or *backward characteristics*, $\boldsymbol{Y}(t, \boldsymbol{x})$, and considering therefore that $\boldsymbol{x}_0 = \boldsymbol{X}(0, \boldsymbol{x}_0) = \boldsymbol{Y}(t, \boldsymbol{x}) = \boldsymbol{Y}(t, \boldsymbol{X}(t, \boldsymbol{x}_0))$, and that $\boldsymbol{x} = \boldsymbol{X}(t, \boldsymbol{x}_0) = \boldsymbol{x}(t, \boldsymbol{Y}(t, \boldsymbol{x}))$, equivalently, by differentiating the former one has that:

$$\begin{cases} \dfrac{\partial \boldsymbol{Y}}{\partial t} + \boldsymbol{V}(t, \boldsymbol{x}) \cdot \nabla \boldsymbol{Y}(t, \boldsymbol{x}) = 0 \\ \boldsymbol{Y}(t = 0, \boldsymbol{x}) = \boldsymbol{x} \end{cases} \tag{4.15}$$

Which represents a classic transport problem for the backward characteristic functions associated with each grid point $\boldsymbol{x}$ at time t, and whose initial condition must be specified. Basically, by adopting such a point of view means to look directly at the deformed configuration of an elastic or moving body, and trying then to reconstruct its initial configuration by moving backward along indirect characteristics of the solution, as depicted in figure 4.22. By their definition, it follows that backward and forward characteristics are related one each other such that:

$$\boldsymbol{X} = \boldsymbol{Y}^{-1}$$

Thus, coming back to the eulerian description of a FSI problem, we can now apply this approach to the representation of elestic or visco-elastic materials immersed in the fluid domain, computing their displacement field as shown, and obtaining consequently the deformation gradient tensor $\boldsymbol{F}$ (see section 2.4) by taking:

$$\boldsymbol{F} = \nabla \boldsymbol{X} = (\nabla \boldsymbol{Y})^{-1}$$

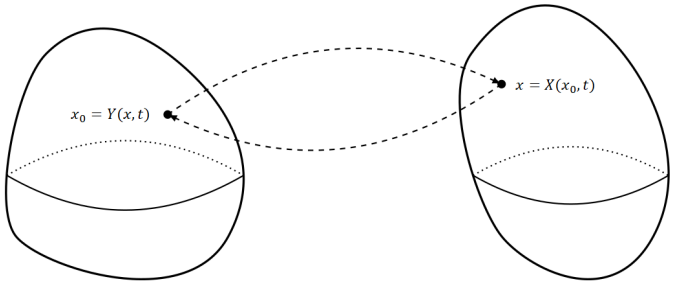


Figure 4.22: Illustration of a deforming-body's characteristic functions and their relationship.

Therefore, now it is possible to define a global coupled stress tensor to include in the Navier-Stokes equations set, leading to the aimed monolithic description we wanted to achieve. Such a tensor, $\boldsymbol{\sigma}$, is said to be coupled in the sense that it reduces to the normal viscous stress tensor introduced in section 2.2 when we are dealing with pure fluid, out of any immersed body (or, in the context of the monolithic stand point, when a region with high porosity is encountered), as stated by the interface localizer function, whereas it includes an elastic (or visco-elastic) contribution if deforming bodies are crossed. As a model of coupled stress tensor $\boldsymbol{\sigma}$, thus depending on the velocity field $\boldsymbol{V}$ for the viscous part and on the backward characteristics $\boldsymbol{Y}$ for the elastic one, we could resort to a relation such as:

$$\boldsymbol{\sigma}(\boldsymbol{V},\boldsymbol{Y}) = \big((1-\chi_e)\mu_f + \chi_e\mu_e\big)\big(\nabla\boldsymbol{V} + \nabla\boldsymbol{V}^t\big) + \chi_e\boldsymbol{\sigma}_e(\boldsymbol{Y}) \tag{4.16}$$

Where $\boldsymbol{\sigma}_e$ is the specific elastic model adopted for the considered problem (for our case, the Mooney-Rivlin hyperelastic one represented by relation 2.39, whereas the other appearing quantities are listed below:

- μ_f: fluid's dynamic viscosity;
- μ_e: solid material's visco-elasticity constant, since its behaviour is assumed to depend also on the elastic strain rate, other than on the strain itself;
- χ_e: characteristic function distinguishing those regions within moving obstacles from those belonging just to the fluid, and it can therefore be referred to the information carried by the interface localizer, say the Level-Set function (definition in the following);

As a proposal for the function χ_e, one could consider the following smeared definition associated with the sign of a Level-Set function ϕ (with $\phi > 0$ inside the immersed bodies, $\phi < 0$ elsewhere) rather than the classic heavyside function, oftentimes causing

the occurrence of numerical instabilities next to the interfaces:

$$\chi_e = \begin{cases} 1 & \text{if} \quad \phi > +\delta \\ 0 & \text{if} \quad \phi > -\delta \\ 1 - \frac{1}{2}\left(1 + \frac{\phi}{\delta} + \frac{1}{\pi}\sin\left(\pi\frac{\phi}{\delta}\right)\right) & \text{if} \quad -\delta < \phi < +\delta \end{cases}$$

Where the quantity δ is typically assumed to be $\mathcal{O}(\sqrt{\Delta x \Delta y})$.

Concerning the discretization of the model, finally, we adopted the same spatial schemes as for the Level-Set function transport problem 4.7 to numerically solve problem 4.15, that is, considering $\boldsymbol{Y} = (Y_x, Y_y)$:

$$\begin{cases} \dfrac{(Y_x)_{i,j}^{k+1} - (Y_x)_{i,j}^{k}}{\Delta t} + u_{i,j}^{k}\left(\dfrac{\delta Y_x}{\delta x}\right)_{i,j}^{k} + v_{i,j}^{k}\left(\dfrac{\delta Y_x}{\delta y}\right)_{i,j}^{k} = 0 \\ \dfrac{(Y_y)_{i,j}^{k+1} - (Y_y)_{i,j}^{k}}{\Delta t} + u_{i,j}^{k}\left(\dfrac{\delta Y_y}{\delta x}\right)_{i,j}^{k} + v_{i,j}^{k}\left(\dfrac{\delta Y_y}{\delta y}\right)_{i,j}^{k} = 0 \\ \boldsymbol{Y}_{i,j}^{0} = (\boldsymbol{Y}_0)_{i,j} \end{cases} \tag{4.17}$$

For the computation of the elastic tensor $\boldsymbol{\sigma}_e$ instead, the deformation gradient tensor $\boldsymbol{F} = (\nabla \boldsymbol{Y})^{-1}$ regarding all spatial derivatives of the backward characteristic functions, that is $\frac{\partial Y_x}{\partial x}$, $\frac{\partial Y_x}{\partial y}$, $\frac{\partial Y_y}{\partial x}$ and $\frac{\partial Y_y}{\partial y}$, the latter are obtained exactly the same way we derived the spatial velocity gradients exploited to write the viscous stress tensor components in section 3.2.2, using interface values extracted from cell center's ones. It is now possible to obtain each grid point component of the deformation gradient tensor $\boldsymbol{F}_{i,j}^{k+1}$, and thus the same holds for the Cauchy-Green's tensor $\boldsymbol{B}_{i,j}^{k+1}$ (refer once again to section 2.4), involved in the definition of the Mooney-Rivlin hyperelastic model, equation 2.39. Then, we wrote down the coupled stress tensor by means of a simpler relation than 4.16, that is:

$$\boldsymbol{\sigma}_{i,j}^{k+1} = \boldsymbol{\tau}_{i,j}^{k+1} + (\chi_e)_{i,j}^{k+1}(\boldsymbol{\sigma_e})_{i,j}^{k+1} \tag{4.18}$$

4.3.2 Concluding Adopted Model and Algorithm

To conclude the chapter, our final purpose is then to resume here all adopted approaches in order to clearly present the global method devised and employed to carry out such simulations, which consists of, as it has been named, a cartesian and fully-eulerian monolithic one. Our methodology is defined as cartesian since we employ only very simple and easy-coding orthogonal cartesian grids, structured and thus suitable for a finite differences approach implementation to the aim of numerically solving PDEs problems; afterwards, as largely anticipated above, the method is said to be fully-eulerian since such grid does not move in time with the geometry contained in it, thus remaining fixed and not fitting the immersed bodies, whose interfaces with the fluid medium are described through implicit techniques, mostly by Level-Set functions, advected with the fluid motion again in an eulerian way; lastly, as stated, being a monolithic approach

means that the whole domain is considered as it was composed of a unique and continuous material, characterized by different permeability to the fluid (expressed through some proper function, see section 4.2.1), and thus allowing for a complete general treatment regardless of the specific physical role associated with each grid point, for which always the same equations will be applied, as shown soon in the following, and whose actual behaviour will be dictated by the information carried by the Level-Set function, making the penalisation algorithm work, or activating the coupling stress tensor, or again leaving it governed by the simple fluid equations.

Substantially, the global problem we aim to deal with in general is the following one:

$$\begin{cases} \nabla \cdot \boldsymbol{V} = 0 & \text{in} \quad \Omega \\ \rho\Big(\dfrac{\partial \boldsymbol{V}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{V}\Big) = -\nabla p + \nabla \cdot \boldsymbol{\sigma}(\phi, \boldsymbol{Y}) + \displaystyle\sum_{i=1}^{N_b} \frac{\chi_i}{\lambda}(\boldsymbol{V}_{b,i} - \boldsymbol{V}) & \text{in} \quad \Omega \\ \dfrac{\partial \phi}{\partial t} + \boldsymbol{V} \cdot \nabla \phi = 0 & \text{in} \quad \Omega \\ \dfrac{\partial \boldsymbol{Y}}{\partial t} + \boldsymbol{V} \cdot \nabla \boldsymbol{Y} = 0 & \text{in} \quad \Omega \\ \boldsymbol{V}(\boldsymbol{x} \in \partial\Omega, t) = \boldsymbol{V}_\Gamma(t) & \text{on} \quad \partial\Omega \\ \boldsymbol{V}(\boldsymbol{x}, t = 0) = \boldsymbol{V}_0(\boldsymbol{x}) & \text{in} \quad \Omega \\ \phi(\boldsymbol{x}, t = 0) = \phi_0(\boldsymbol{x}) & \text{in} \quad \Omega \\ \boldsymbol{Y}(\boldsymbol{x}, t = 0) = \boldsymbol{Y}_0(\boldsymbol{x}) & \text{in} \quad \Omega \end{cases} \tag{4.19}$$

Where, as previously introduced, the coupled stress tensor could be thought as composed of $\boldsymbol{\sigma} = \boldsymbol{\tau} + \chi_e \boldsymbol{\sigma}_e$. The above reported problem could then be computationally formulated as follows:

1. definition of the domain geometry $\Omega = \big(x_0 + \Delta x(0, N_x - 1)\big) \times \big(y_0 + \Delta y(0, N_y - 1)\big)$, initial moving obstacles and walls' Level-Set function $\phi^{k=0}$, used for penalisation, and initialisation of velocity and pressure fields $u^{k=0}$, $v^{k=0}$ and $p^{k=0}$;

2. definition of the Poisson's matrix used to solve the homonymous elliptic problem at each time step;

3. definition of the Level-Set function associated with elastic bodies $\phi_e^{k=0}$, which will be transported with the fluid after each time step to compute the body motion, firstly reinitialized 5-6 times;

4. start of time iterations:

   (a) computation of the current time step Δt according to stability conditions (CFL condition) and updating of simulation time;

(b) transport of backwards characteristics $\boldsymbol{Y}^{k-1} \rightarrow \boldsymbol{Y}^{k}$ by solving the related problem 4.17, computation of elastic body's deformation;

(c) computation of coupled stress tensor elements $\boldsymbol{\sigma}^{k}$ by evaluating its viscous stress component τ^{k} all over the fluid-body domain, depending on the fluid field solution $\boldsymbol{V}^{k}$, and its elastic component $\boldsymbol{\sigma}_{e}^{k}$ inside the elastic body, thanks to the elastic level-Set function ϕ_{e}^{k}, depending on the deformation gradient tensor obtained via the backward characteristics field $\boldsymbol{Y}^{k}$;

(d) PREDICTION STEP: computation of the predicted velocity field by imposing desired velocity boundary conditions (see sections 3.3.2 and 3.3.3) $\boldsymbol{V}^{k} \rightarrow \boldsymbol{V}^{k+\frac{1}{2}}$, not satisfying the divergence-free condition;

(e) computation of face-centered velocities $\mathcal{U}^{k+\frac{1}{2}}$ and $\mathcal{V}^{k+\frac{1}{2}}$ starting from the predicted cell-centered ones $u^{k+\frac{1}{2}}$ and $v^{k+\frac{1}{2}}$;

(f) PROJECTION STEP: resolution of corrective pressure Poisson's problem to find the scalar field Φ^{k+1} capable of restoring the divergence-free condition on the velocity solution (see section 3.3.1), respecting the desired pressure boundary conditions;

(g) CORRECTION STEP: subsequent application of the corrective scalar field Φ^{k+1} on the velocity and pressure ones, in order to restore an incompressible solution and conclude the time integration, $\boldsymbol{V}^{k+\frac{1}{2}} \rightarrow \boldsymbol{V}^{k+1}$, $p^{k} \rightarrow p^{k+1}$;

(h) computation of moving body motion (membrane holders), if necessary;

(i) penalisation algorithm on the velocity field inside solid bodies (membrane holders) and beyond external aortic walls, by adopting a continuous forcing method as shown in section 4.2.1, that is expression 4.12;

(j) transport of the elastic Level-Set function $\phi_{e}^{k} \rightarrow \phi_{e}^{k+1}$ by solving the related problem 4.7;

(k) computation of the new Level-Set function ϕ^{k+1} based on the new geometric configuration for the membrane holders and aortic walls;

(l) reinitialization algorithm over ϕ_{e}^{k+1}, when necessary;

5. end of time iterations.

Chapter 5

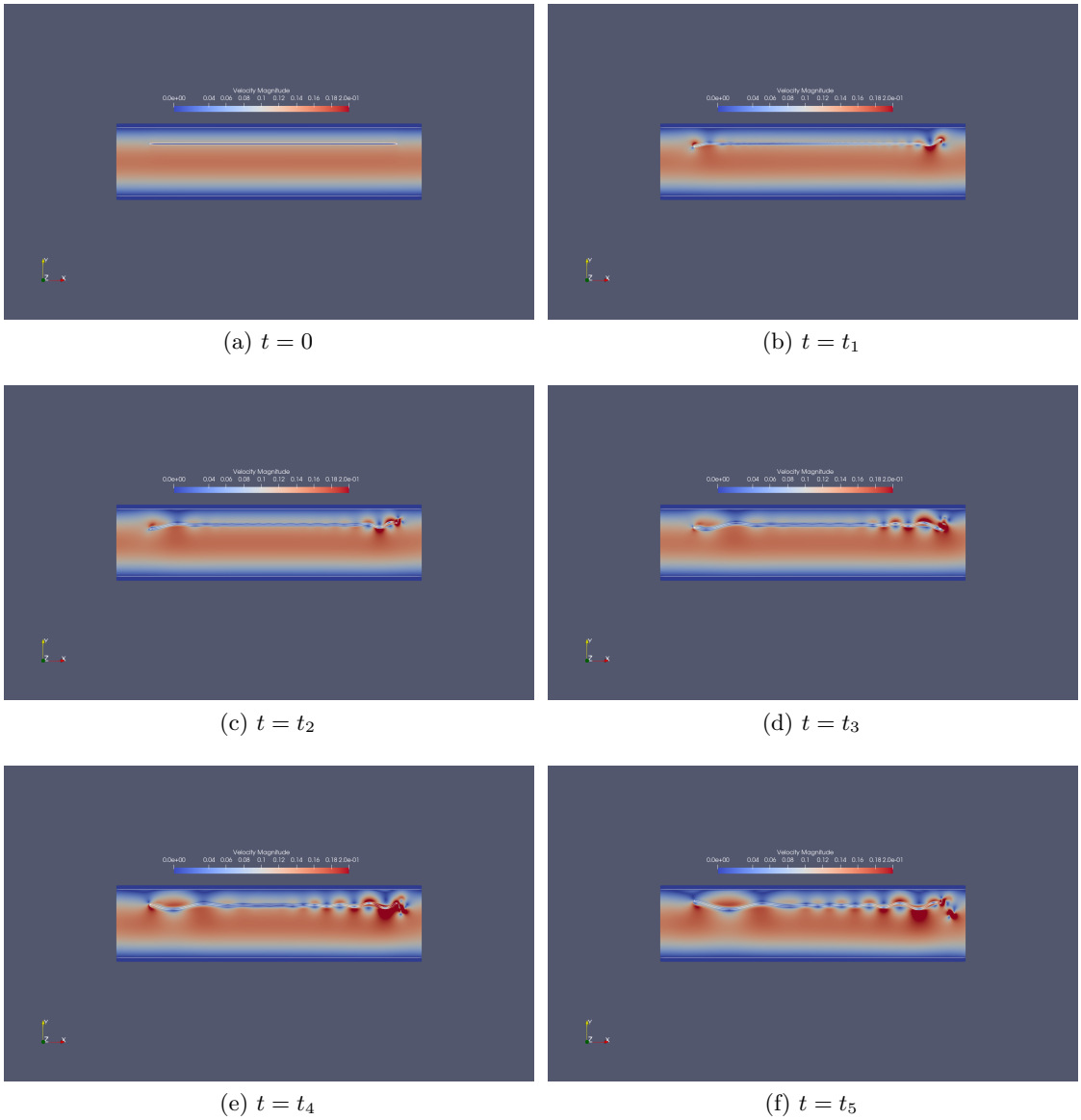
Numerical Results

This last chapter of the thesis is devoted to the collection of all numerical results obtained by the performed simulation over the introduced case of study, that is, a 2-dimensional type B aortic dissection occurring within the descending aortic channel, where the formation of a second *false* lumen is shown, beside the *true* one, and the interaction between the fluid and the moving obstacles characterising a realistic hemodynamics phenomenon is taken into consideration and highlighted. All the mathematical and numerical approaches and assumptions, as well as all methodologies and operative choices regarding boundary conditions, simulation parameters, etc., have already been widely discussed in the previous parts of this work, and here only brief mentions will be made in that sense. The following is organized in such a way that we will first introduce some very simple and not meaningful test-cases, in order to show the potentialities of the presented method, to proceed, then, with the more interesting simulations regarding non-stationary situations involving a beating-heart pump as inflow boundary condition, to be completed later by adding 0-dimensional model in order to ensure a more realistic behaviour of the studied object, as well as to guarantee the well-posedness of the problem. Afterwards, a model of compliant aortic channel is proposed, where the fluid-structure interaction is extended even to the external channel walls, and finally, to conclude, we show the most interesting results concerning a validated closed-loop multi-dimensional model, where the downstream effect of the neglected cardiovascular system is represented onto the 2D domain through its coupling with a 0D electrical analogue of the former, thus allowing for the regularization effects visible in the flow rate and pressure signals due to the compliant behaviour associated with large vessels, evaluated in the presence of an aortic dissection, with external compliant aortic walls and in the complete case where both elements are considered.

Images and animations were obtained via the open-source application ParaView (website at `https://www.paraview.org/`), plots are computed using MatLab, whereas all the presented results have been extracted from simulations performed exploiting the computing resources made available by MCIA's cluster *Curta* (website at `https://redmine.mcia.fr/projects/clu`

5.1 Preliminary Test-Cases

Following the algorithm description given in section 4.3.2, one has to consider the fact that the present code was originally devised for the simulation of self-propelled swimmers in a not-moving, or stationary-moving, fluid domain (see [15]), and this first results will then present an example case involving an elastic membrane immersed in a steady-state Poiseuille flow, and whose right and left holders are moving according to a given law, e.g. harmonic. The main purpose of this first simple situation is that of presenting the



(a) $t = 0$ (b) $t = t_1$

(c) $t = t_2$ (d) $t = t_3$

(e) $t = t_4$ (f) $t = t_5$

Figure 5.1: Steady-state flow test-case with harmonic moving holders.

potentiality of a monolithic model as it has been introduced in the previous chapters,

applied to an incompressible flow whose behaviour is handled via the fractional step method algorithm, and where a 1st order penalisation technique belonging to the immersed boundary continuous forcing class is employed to impose the motion of solid bodies as well as the no-slip condition on external rigid walls. Thus, in the context of this first simulation, a stationary Poiseuille flow is adopted as inflow boundary condition, whereas the no-stress condition is employed on the opposite outflow section, with homogeneous Neumann conditions for the corrective pressure field on all the four boundaries. The right holder undergoes a harmonic motion with twice the frequency of the left one, in phase opposition and with the same amplitude. Some snapshots of the solution are reported in figure 5.1.

In those figures, the membrane, the two holders and the rigid aortic walls have been put in evidence by showing the 0-level set of the related level-Set function, whose transport is responsible for the possibility to keep track of the elastic body's motion driven by the fluid and the oscillating holders. As one can notice by observing the above reported illustrations, the fluid field is barely disturbed within the lower half channel, but interesting structures arise all around the deforming membrane, where localized peak velocity zones appear alternatively with low velocity ones. By studying, for instance, the vertical distribution of one velocity component over the cross-section area of the channel at a given horizontal position, one can appreciate the boundary layer arising next to the external walls, which leads the velocity field from a null value, correspondingly to the imposition of a no-slip condition, up to the maximum central value, and also to the two sides of the floating membrane's surface, or of its holders, where the velocity field's solution links them with the upper and lower moving fluid regions. Finally, the oscillating motion of the elastic membrane is almost completely due to the imposed harmonic motion of its holders, and little affected by the fluid itself. One can thus appreciate the propagation of mechanic travelling waves along the elastic membrane, which propagate from each holder to collide, then, in the middle of the body and being therefore reflected and transmitted, giving rise to more complicated shapes.

Several different simulations have been performed by adopting these settings, when a steady-states flow is imposed all over the domain at each instant of time, in order to take confidence with the model and to test its simpler features by changing, among the others, some parameters such as the law of motion governing the holders, their position, the membrane dimension, the adopted spatial scheme, the time integration step (always respecting the CFL condition), etc. In addition, some attempts have also been carried out concerning a constant-regime 0D Windkessel model imposed as outflow condition, in order to calibrate its elements and to better understand its coupling effects with the higher-dimensional domain.

5.2 Beating Heart Test-Cases and New Improvements

Within the present section we would like to present some of the tested improvements to our simulation settings. Among these, one can obviously appreciate the pulsating inflow adopted as new Dirichlet-like boundary condition for the velocity field, applied from now on to the left inlet section of the domain, and formulated according to several different

configurations; a new geometrical initial profile for the elastic membrane, which will be let free to move on the right side, by removing the corresponding holder, and attached to the left one, now fixed at a given position; imposition of different velocity profile as inflow velocity distributions, beside the initial Poiseuille's, in order to select the most suitable one according to the solution computed inside the channel.

Pulsating Inflow Time Profile:

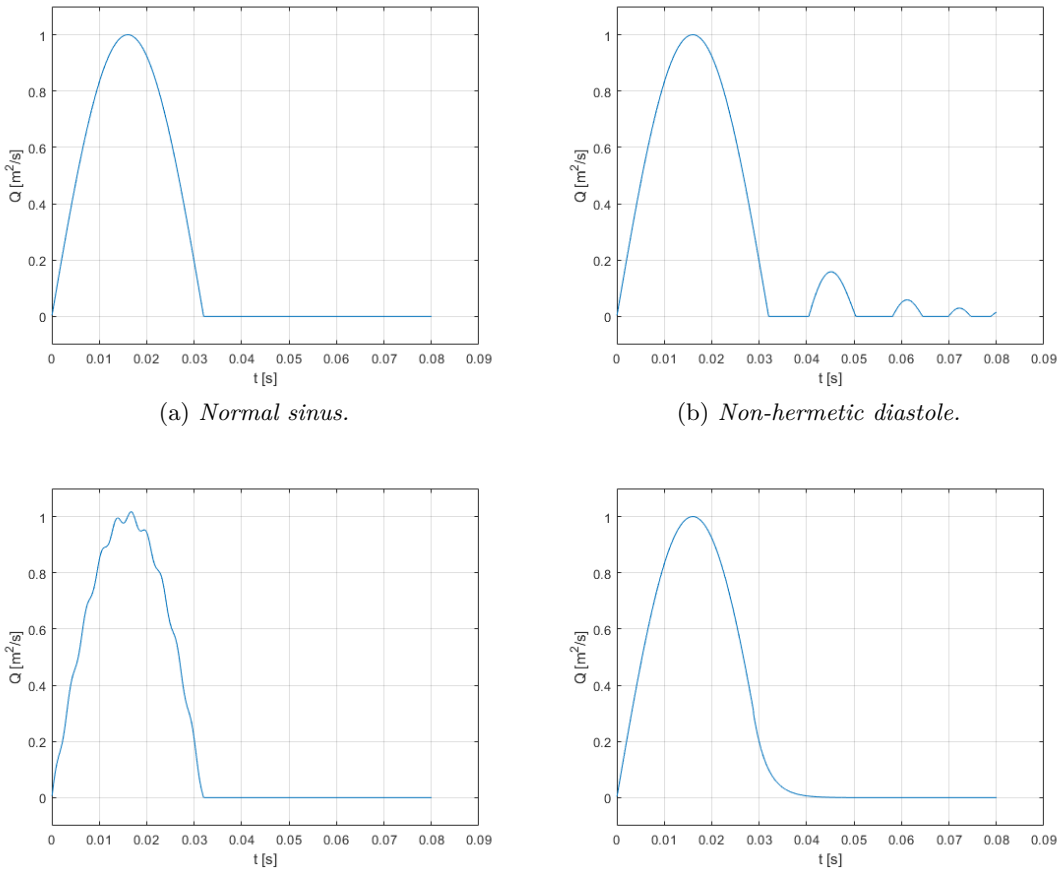
Following the discussion proposed in section 3.3.3 on velocity boundary conditions, we designed a pulsating flow rate according to the most common data associated with a human heart beating, whose mechanical behaviour is substantially divided in two phases (see section 1.2 for more details): *systole* and *diastole*. Approximately, we assumed for such phases of the RR cardiac cycle to be, respectively, the 40% and the remaining 60% of the entire RR beat, that is $T_s \approx \frac{2}{5}RR$ and $T_d \approx \frac{3}{5}RR$. Therefore, we defined a time-depending function $q(t)$ following the example provided by expression 3.50, and employed it by multiplying to the inflow velocity distribution $u_{in}(y)$ at each time instant, in order to build a time-depending velocity profile with a fixed spatial shape (discussed later), accordingly to expression 3.50. Different attempts have been carried out regarding this aspect, to begin with the simplest time-history composed of a sinusoidal profile during systole, and a flat null phase during diastole, to proceed, then, with an intermitting diastolic phase, with some non-hermetic pulses following the same sinusoidal systolic phase. Afterwards, a disturbed systolic profile is proposed and employed especially during the validation of the Windkessel 0D model, and its coupling with the 2D domain (as it will be shown in the following), whose stability properties needed to be established and checked. Lastly, we implemented a smoothed profile in order to improve the resulting pressure signal associated with the solution inside the domain, by linking the sinusoidal systolic profile to the flat diastolic one through an exponential decay connection. In figure 5.2 the explained proposals are shown.

Inflow Velocity Spatial Distribution:

Again by referring to what stated in section 3.3.3, different spatial distributions have been implemented and tested as inflow velocity conditions over the inlet section of the domain, that is, for $u_{in}(y)$, combined with the above reported time profiles. Proceeding in order, a traditional Poiseuille's parabolic profile was mainly employed to perform the most of the simulations, then some attempts have been made by adopting multiple types of Womersley's profiles, with different shapes, but finally, as the best case, according to obtained solutions, a 10th degree polynomial profile has been chosen as the definitive inflow velocity distribution. The mentioned cases are reported in figure 5.3.

Tunica Intima Membrane Definition:

In order to achieve more realistic behaviours, a different initial shape was given to the detached aortic membrane representing the *tunica intima* floating wall causing the dissection. As stated above, a unique holder has been employed to keep its position fixed



(a) *Normal sinus.* (b) *Non-hermetic diastole.*

(c) *Disturbed systole.* (d) *Smoothed sinus.*

Figure 5.2: Pulsating inflow time profile (1st beat, $RR = 0.08s$).

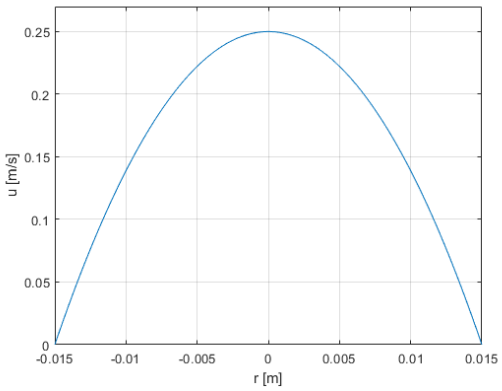
on the left, letting it free to move on the right side. As it regards the shape given to the membrane, an exponential profile has been adopted, according to a definition such as:

$$\begin{cases} y_{mUP}(x) = y_{mMAX} - \left(y_{mMAX} - y_{H1} - \dfrac{h}{2}\right) e^{k(x_{h1}-x)} - \dfrac{h}{2}, & \text{with} \quad x_{h1} < x < x_{h2} \\ y_{mDOWN}(x) = y_{mUP}(x) - h \end{cases}$$

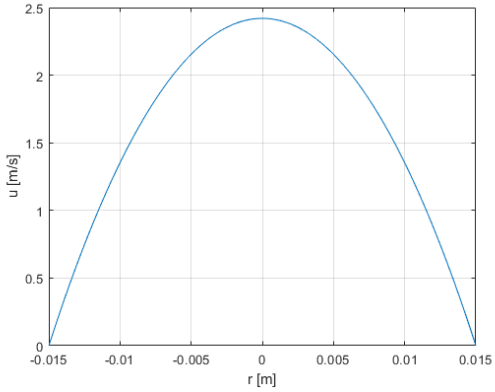
Where x_{h1}, x_{h2} and y_{h1} are the holders coordinate, y_{mUP}, y_{mDOWN} and y_{mMAX} are, respectively, the upper and lower vertical coordinates of the membrane profile, and the maximum vertical value of its midline, whereas h and k are, the former, the membrane thickness, the latter, a multiplicative constant dictating the steepness of the exponential profile (assume typically $k \approx 100$). An example is provided by figure 5.4.

Beating Heart Test-Case:

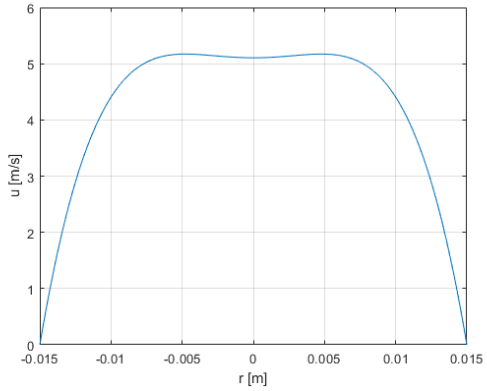
As an example, we propose here same snapshots taken from a simulation performed by adopting some of the aforementioned improvements, such as a normal sinus time profile as



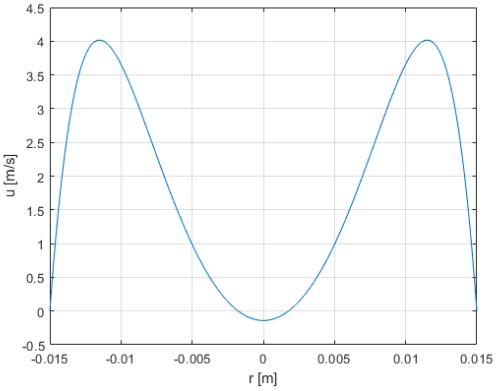
(a) *Poiseuille's.*



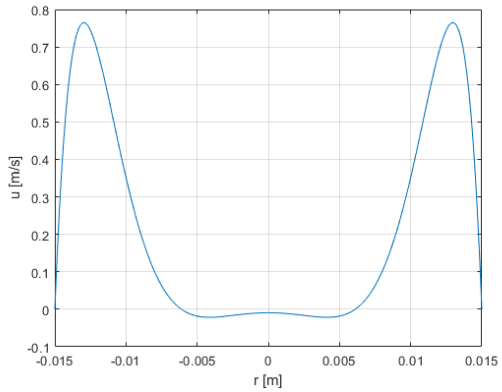
(b) *Womersley's,* $\alpha = 1$.



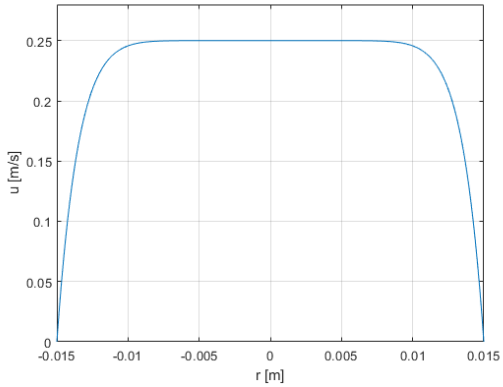
(c) *Womersley's,* $\alpha = 3$.



(d) *Womersley's,* $\alpha = 5$.



(e) *Womersley's,* $\alpha = 10$.



(f) *10th degree polynomial.*

Figure 5.3: Inflow spatial distribution profile, $RR = 0.08s$).

inflow condition, imposing also Poiseuille's velocity distribution on the same section, and the initial exponential shape described above for the floating membrane. In figures 5.5

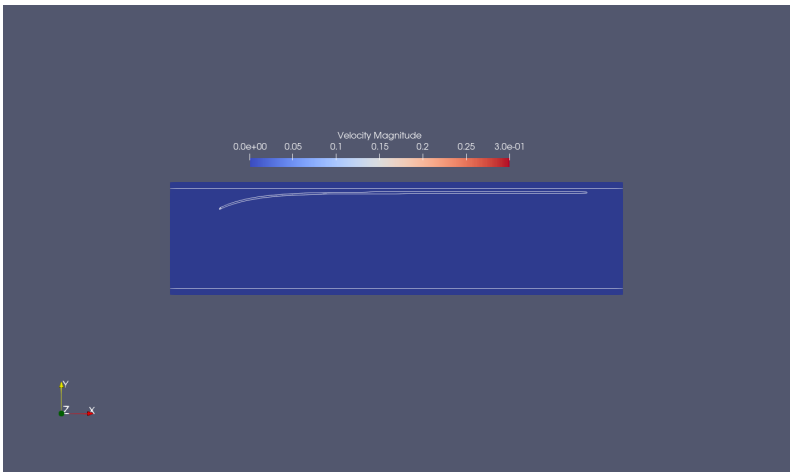
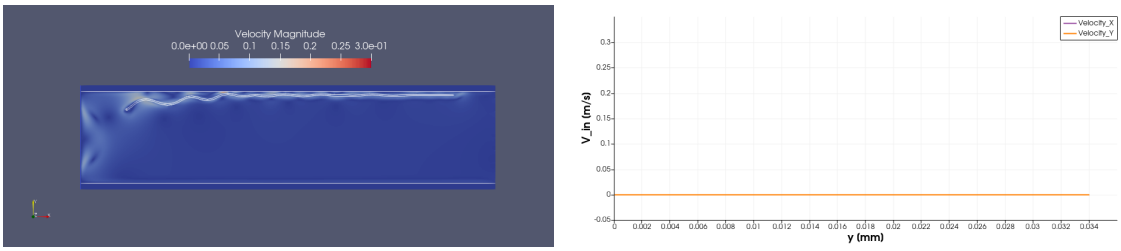


Figure 5.4: Exponential shape associated with the floating *tunica intima* membrane.

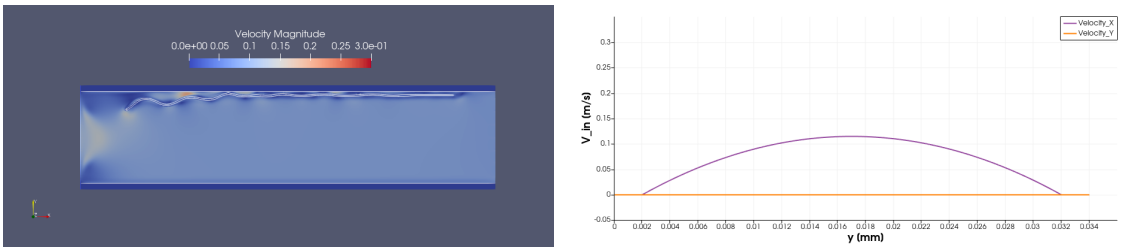
and 5.6 one can find different situations corresponding to subsequent time instants, with on the left the solution configuration within the aortic channel, and on the right the respective inflow velocity profile indicating which part of the heart beat we are dealing with (the snapshots are not taken starting from the very first heart beat). A remark is worth making at this point, concerning boundary conditions for the pressure Poisson's equation: at the very beginning of those simulations, before implementing 0D Windkessel models in order to provide the problem with more realistic and reliable conditions from the downstream vasculature, homogeneous Neumann conditions were kept on all the four boundaries of the domain for the corrective field $\Phi^{k+1} = \Delta t(p^{k+1} - p^k)$ (refer to section 3.3.3 for a detailed discussion on this topic). As argued, this aspect represents quite an issue for the posedness of the problem, since it leads to an ill-posed definition whenever non-stationary inflow conditions are applied on the velocity field at the inlet boundary. Therefore, a so-posed problem should not have admitted a valid solution, but this seems not to be the case since the above presented results coincide exactly with those subsequently obtained by applying Dirichlet conditions on the outflow boundary, for the corrective pressure field. As shown in the following, this ambiguity is due to the fact that the required tolerance for the Krylov's solver (a numerical tool devoted to the solution of the linear system associated with the elliptic Poisson's problem, see appendix B) was set to be very high, say up to 10^{-3}, and evidently, by decreasing such threshold tolerance up to 10^{-12}, for instance, the ill-posed nature of the problem defined with homogeneous Neumann conditions arises and the solution begins to explode in time. More detailed examples regarding this aspect will be given in the following sections, dedicated to the implementation of multiscale models and their utility.

5.3 Multiscale Models

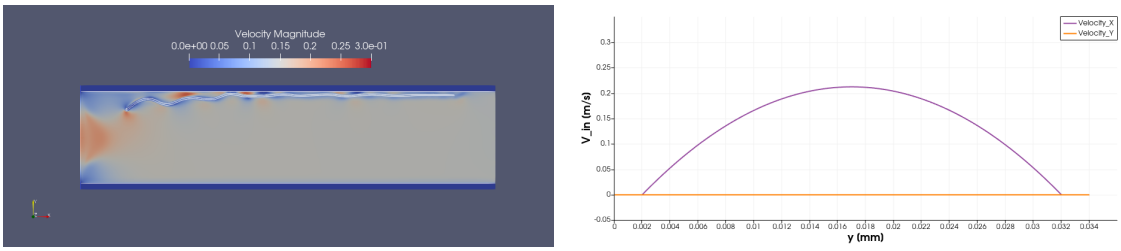
The present section includes some of the most relevant results concerning this master's thesis work, especially as far as the mathematical point of view of the problem posedness



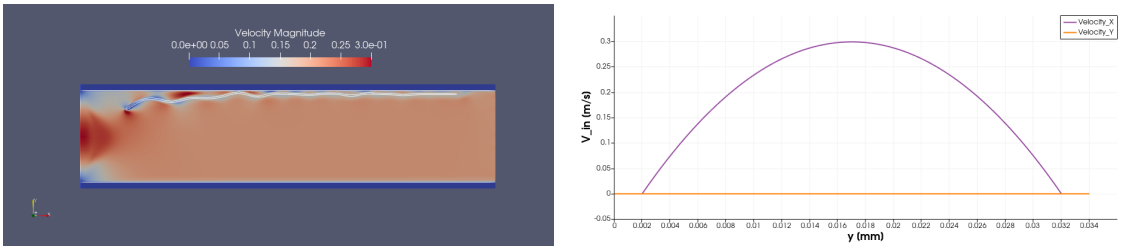
(a) $t = t_1$



(b) $t = t_2$



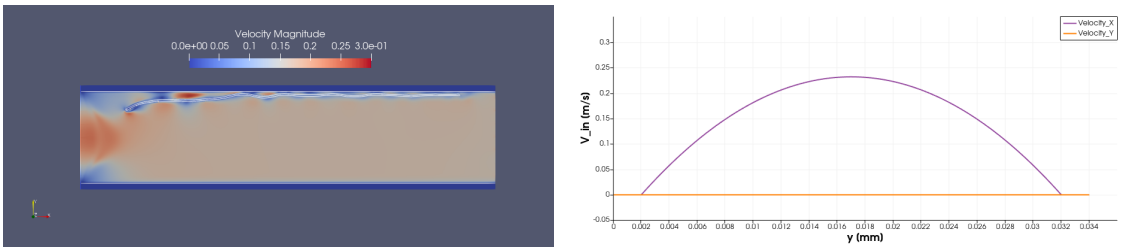
(c) $t = t_3$



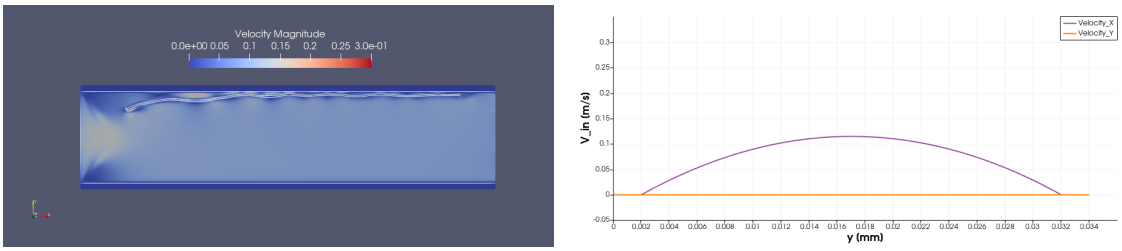
(d) $t = t_4$

Figure 5.5: Beating heart solution with freely-floating *tunica intima* membrane (1).

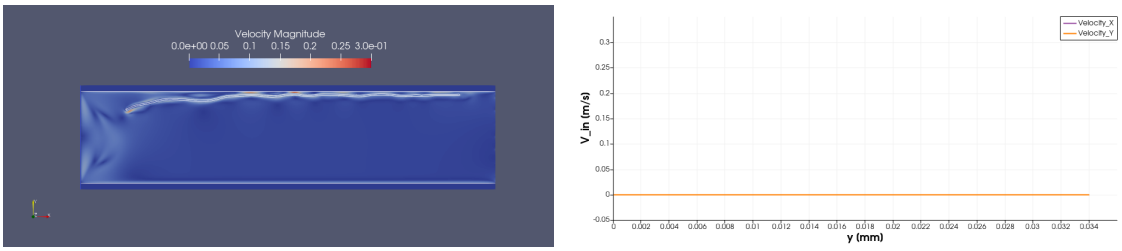
is concerned. the necessity to develop lower-dimensional models to supply the higher-dimensional ones, indeed, does not come uniquely from the desire to perform more realistic and faithful simulations characterized by true pressure distributions within the aortic channel, conversely, such needing arises since we aimed to adopt a time-dependent inflow condition by imposing a pulsating flow-rate resembling a true heart-beating pump. For that reason, the Poisson's problem linked to the projection stage of the fractional step method described in section 3.3 had to be provided with proper conditions on its boundaries, that is with at least one Dirichlet-like condition for the corrective pressure



(e) $t = t_5$



(f) $t = t_6$



(g) $t = t_7$

Figure 5.6: Beating heart solution with freely-floating *tunica intima* membrane (2).

field, establishing a priori the solution of the problem. This necessity comes from what has been widely discussed in section 3.3.3, briefly, because the problem represented by the following equation needs to be consistent:

$$\int_{\Omega} \nabla \cdot \boldsymbol{V}^* \, d\Omega = \frac{1}{\rho} \int_{\Gamma} \nabla \Phi \cdot \boldsymbol{n} \, d\Sigma$$

Which cannot be true as long as the right-hand side nullifies because of the imposed boundary conditions while the left-hand one remains necessarily different from zero, in average, over the whole domain, after each new flow-rate injection. Therefore, within this section, we first introduce the tuning procedure of the simplest 3-elements Windkessel model applied to our domain as a pure outlet condition, by means of a complete 0D model where the same flow-rate time law was applied as input condition, and the corresponding pressure was taken as the main output signal. Afterwards, we will present some attempts carried out in order to prove that conditioning property of the problem by showing the obtained results related to the imposition of different outflow pressure conditions, via the exploitation of a smaller domain not containing any elastic moving or deforming immersed body, in order to make the simulations faster and numerous, to go farther, then,

towards the last part of the section, where there will be presented the development of more interesting configurations with double exit sections, or with a closed-loop globally compliant multiscale cardiovascular system, in which the 2D domain is inserted as a specific portion of that (the separate section 5.5 will be dedicated to this last interesting problem).

WK3 Tuning Procedure:

The multiscale model we would like to adopt includes a 3-elements Windkessel circuit (see section 2.5) as outflow condition, providing a pressure signal to impose on the corresponding outflow section of the 2D domain, from which, in turn, the 0D model receives the time-varying flow-rate flowed throughout the entire aortic channel. The global multi-D model is therefore illustrated in figure 5.7. By simply removing the CFD model

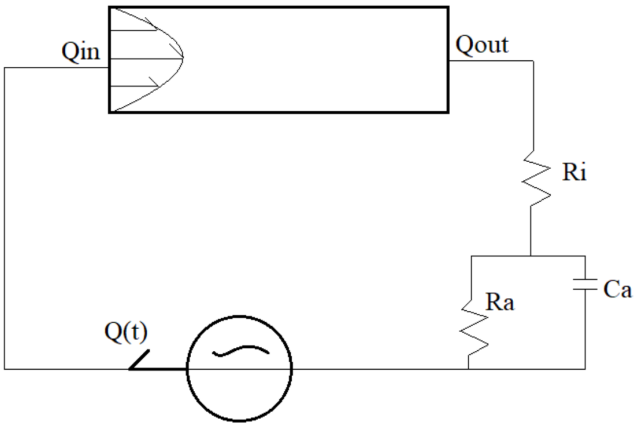


Figure 5.7: Multi-D model composed of a WK3 electrical analogue connected to the outlet section of the CFD model.

one can build a completely 0D electric circuit and then exploit it in order to tune its elements and reach the desired pressure signal range, taking into consideration the main features of the flow signal affecting the pressure solution that are: the frequency of the signal, or in other words the heart beat RR, and the peak flow rate Q_{max}, linked to the maximum central velocity within the aortic channel through the selected velocity profile. To this aim, we recall here the ODE 2.41 governing the electric circuit representing the investigated 0D model, where the values of electrical parameters R_a, R_i and C_a are those to be calibrated.

$$p(t) = Q(t)(R_a + R_i) - C_a R_a \frac{\mathrm{d}p(t)}{\mathrm{d}t} + C_a R_a R_i \frac{\mathrm{d}Q(t)}{\mathrm{d}t}$$

Considering a maximum axial velocity of $u_{max}(x,0) \approx 0.30\,m/s$, by adopting a Poiseuille profile and designing an aortic channel whose radius measures $R = 0.015\,m$, one has to deal with a maximum flow-rate of about $Q_{max} \approx 6.0 \times 10^{-3}\,m^2/s$, whereas a frequency equal either to $RR = 0.8\,s$ or to $RR = 0.08\,s$ will be considered (as anticipated, even some

constant flow test cases were made in order to prove the stability of the system, as well as including small temporal perturbations). Substantially, it is a matter of studying a *RRC* electric circuit where the input signal consists of an oscillating part and a constant one. In table 5.1 there are reported those so obtained tuned values useful to set up our 0D model, together with some values found in the literature (see [17]), to make a comparison. Therefore, in the following figures 5.8, 5.9 and 5.10 the resulting pressure

Table 5.1: WK3 tuning parameters.

$RR\,[s]$	$Q_{max}\,[m^2/s]$	$R_a\,[kg/(sm^4)]$	$R_i\,[kg/(sm^4)]$	$C_a\,[m^4s^2/kg]$
Biblio	n.s.	4.7×10^8	1.5×10^7	1.2×10^{-9}
0.08	6.0×10^{-3}	8.8×10^6	6.5×10^5	1.8×10^{-8}
0.80	6.0×10^{-3}	8.8×10^6	6.5×10^5	1.8×10^{-7}
const.	6.0×10^{-3}	1.2×10^6	6.5×10^5	0.0

signals are reported together with the associated flow-rate signals and those results coming by adopting parameter values equal to the literature's ones.

Problem Posedness Testing:

This paragraph, as anticipated, is dedicated to the issue of well-posedness of the problem, depending on the choice of its boundary conditions, especially regarding those for the pressure Poisson's equation introduced with the projection stage of the FSM. As stated above, the problem represented by the reported integral equation for the divergence of the predicted field $\boldsymbol{V}^*$, being the latter directly linked to the imposition of boundary conditions for the scalar solution field Φ, one has to deal with their choice, which could turn out to be rather a tough task. Here, we present the results obtained in terms of inflow and outflow mean pressures $\bar{p}_{in} = \bar{p}(0,t)$ and $\bar{p}_{out} = \bar{p}(L_x,t)$, of the corresponding flow-rates $Q_{in} = Q(0,t)$ and $Q_{out} = Q(L_x,t)$, and of a relative error associated with those flow-rates, with respect to a reference value $Q_{ref} = 6.0 \times 10^3 m^2/s$, and thus defined as:

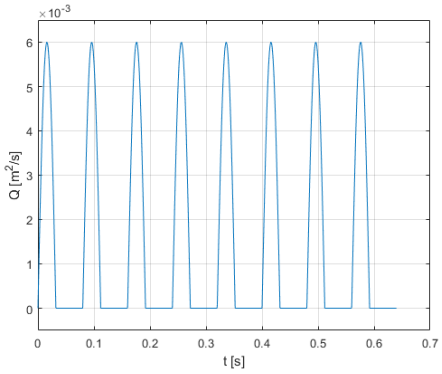
$$err_Q = \frac{|Q_{out} - Q_{in}|}{Q_{ref}}$$

Whereas the flow-rates and mean pressures are computed as:

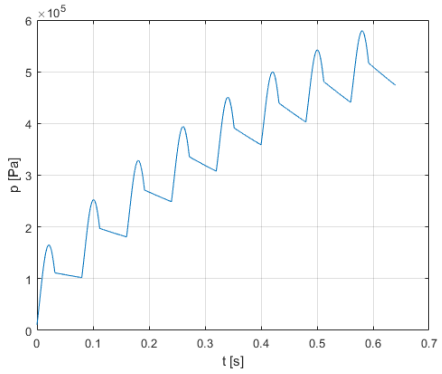
$$Q_s(x_s,t) = \int_S u(x_s,y,t)\,dy, \quad \bar{p}_s = \frac{1}{L_y}\int_S p(x_s,y,t)\,dy$$

Six different situations have been investigated to this aim, in order to check under which conditions the problem could be said as well-posed, thus verifying what already discussed within the dedicated section 3.3.3. In order:

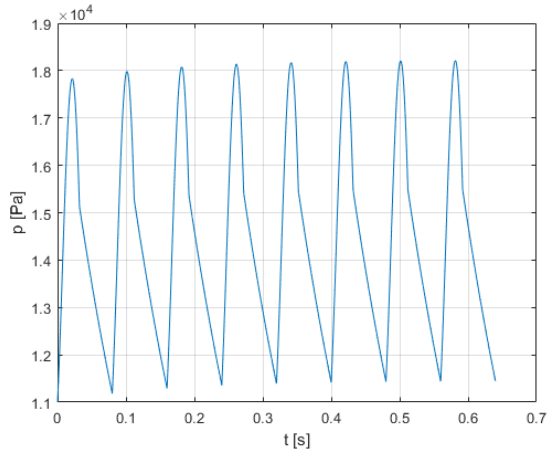
- four cases concerning homogeneous Neumann conditions on all the four boundaries, that is with two different heart beat periods $RR = 0.08\,s$ and $RR = 0.008\,s$, with a modified and non-realistic *superpulsing* inflow condition, and, lastly, with a lower Krylov's solver tolerance;



(a) *Flow-rate signal.*



(b) *Pressure signal with literature's tuning.*

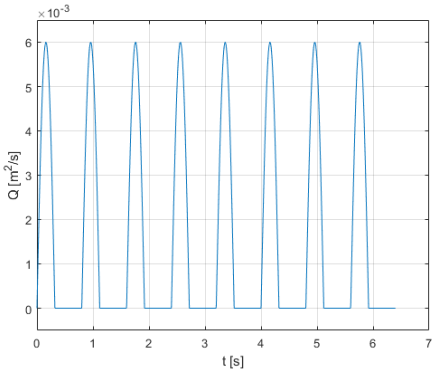


(c) *Tuned pressure signal.*

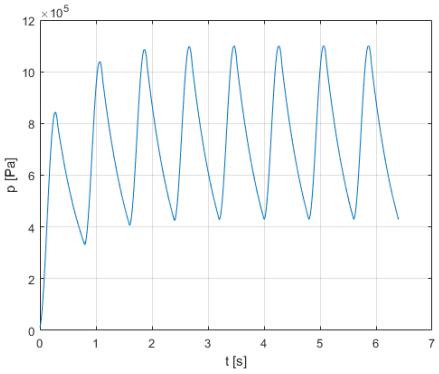
Figure 5.8: Results from tuning procedure with $RR = 0.08\,s$.

- one case regarding the imposition of a Dirichlet boundary condition on the outflow pressure, fixing a value for the pressure constant in time ($p_{out} \approx 11000\,Pa$);
- one last attempt involving the adoption of a 3-elements Windkessel model in the perspective of imposing a time-dependent Dirichlet condition of the outflowing pressure, $p_{out} = p_{wk3}(t)$, deriving from the previously presented ODE associated with the 0D model coupled to the 2D one.

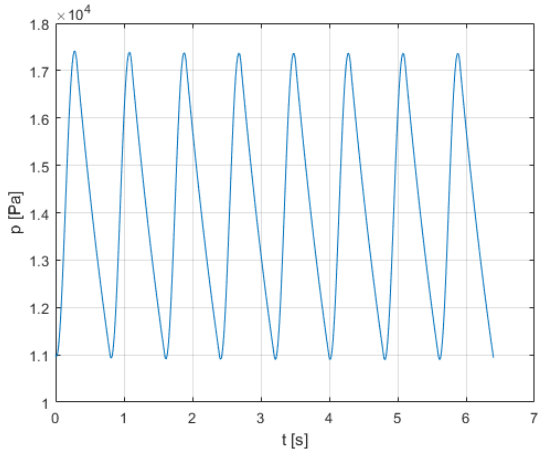
All mentioned test-cases have been carried out both in the presence of the corrective procedure shown in section 3.3.3, through expression 3.57, and in its absence, in order to prove when such improvement could actually be useful and when instead, on the contrary, it could turn out to be even more dangerous. All these results, carried out by means of a smaller simulation involving a reduced domain with no immersed elastic bodies, are sorted and reported in figures 5.11, 5.12, 5.13, 5.14, 5.15, and 5.16. As it can be appreciated by looking at the here reported figures, the issue of problem posedness has to be taken into considerations when choosing the nature of the boundary condition to be applied to



(a) *Flow-rate signal.*



(b) *Pressure signal with literature's tuning.*



(c) *Tuned pressure signal.*

Figure 5.9: Results from tuning procedure with $RR = 0.8\,s$.

the Poisson's problem. In fact, it has been proven how such conditions, when chosen as homogeneous Neumann-kind for all the boundaries, lead to an ill-conditioned problem, which ends up to explode when a sufficiently high accurate Krylov's solver is employed, both in the presence and not of the correction for the divergence of the predicted velocity field. Introducing this last correction, actually, seems to keep the solution under reasonable control almost until the end of the first cardiac cycle, but then it diverges exactly as the non-corrected one does. Considering Dirichlet boundary conditions for the outflow boundary, instead, both the cases of constant and time-varying pressure follow the same behaviour, that is, they ensure the well-posedness of the problem as long as the correction is not applied, thus confirming what a good way of imposing boundary conditions with pulsating flow is like, whereas they both soon diverge when the correction strategy is introduced, due to a basic inconsistency enforced on the problem.

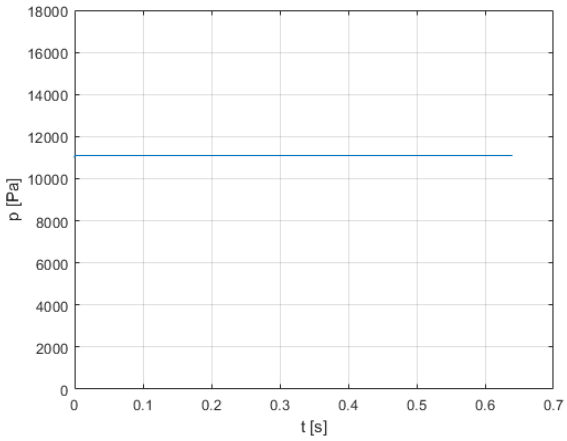
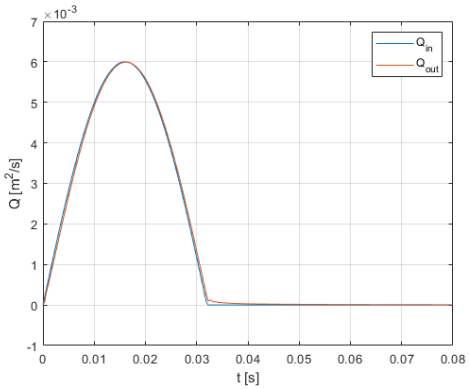


Figure 5.10: Results from tuning procedure with constant flow-rate.

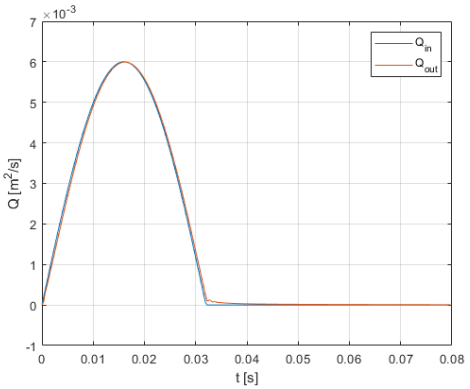
Complex Configurations:

Before concluding this section dedicated to multiscale-models and their utility in ensuring the well-posedness of the problem, as well as their capacity of providing better and realistic results, we would like to mention two different configurations we tried to tackle during this master's thesis work experience. The following cases involve more complicated models composed of either a larger number of lumped parameters, or requiring an augmented interaction between the 0D and the 2D components. The first problem we propose regards the introduction of the so-called *iliac bifurcation*, which can be encountered at the end of the aortic descending channel, within the abdominal region, and which basically consists of a splitting of the blood flow, separating it into the two fractions subsequently envoyed towards the downstream vasculature. In this simulation, two different outflow sections come into play, and therefore appropriate boundary conditions have to be devised and applied to both of them. To our aims, we adopted again two separate 3-elements Windkessel models, as shown in figure 5.17. Finally, the second improved configuration sees the CFD aortic channel added to a branch of the electric circuit representing the 0D model, forming a parallel with a capacitor C. This way, the compliant behaviour of the larger vessels is much more taken into account, and its dynamics affects the flow entering the 2D domain itself, in such a way that will be shown later, in the last section of the chapter, appositely dedicated to this new closed-loop configuration. This way, a strong feedback is introduced into the system, making the 0D model no longer equivalent to a pure outlet effect, not affecting the solution inside the higher-dimensional domain. As outflow condition of the 2D domain, a simple resistor took the place of the WK3 model, but several improvements can be done in this sense. In figure 5.18 this last introduced configuration is illustrated.

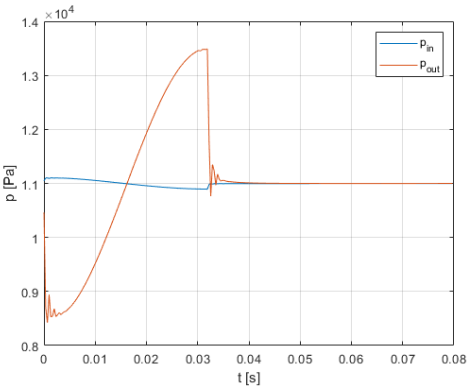
To conclude the section, we propose some snapshots extracted from a solution involving the application of a WK3 simple 0D model to the outflow section of the domain, including the floating membrane representing the dissection, in figure 5.19, together with the flow-rate and pressure time signals registered at the inflow and outflow sections of the 2D



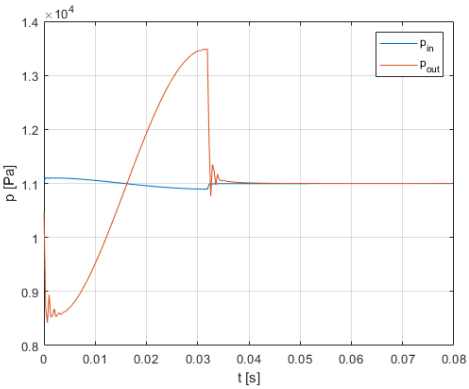
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



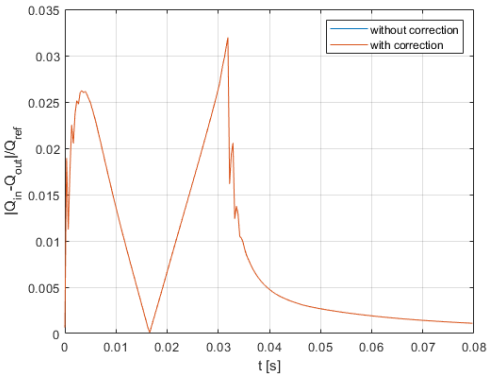
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

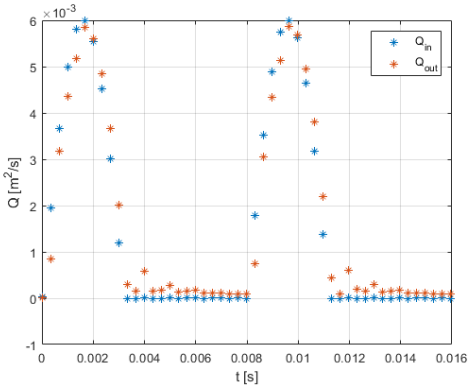


(d) *Inflow (purple) and outflow (green) pressures with correction.*

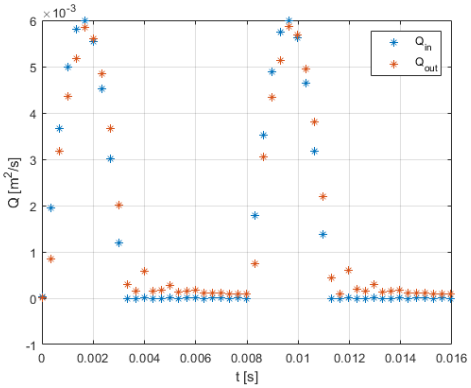


(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

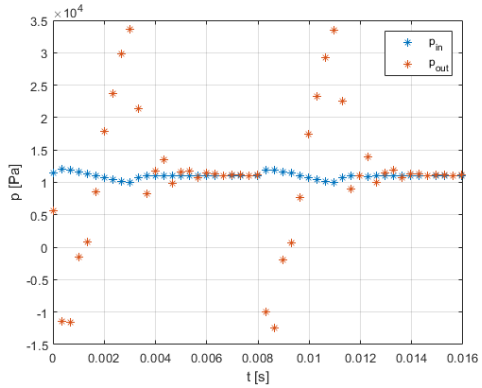
Figure 5.11: Homogeneous Neumann conditions on all boundaries, case $RR = 0.08\,s$.



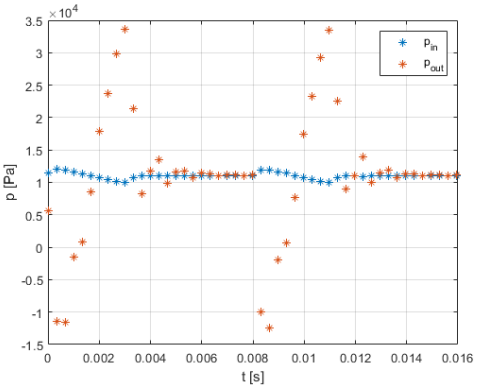
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



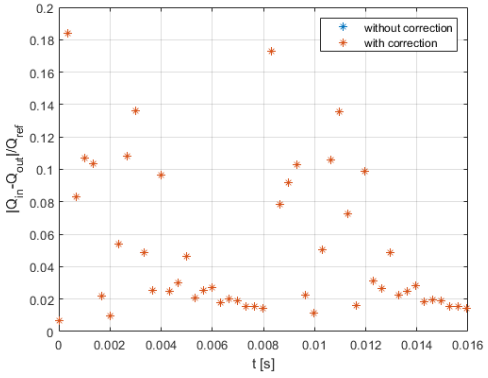
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

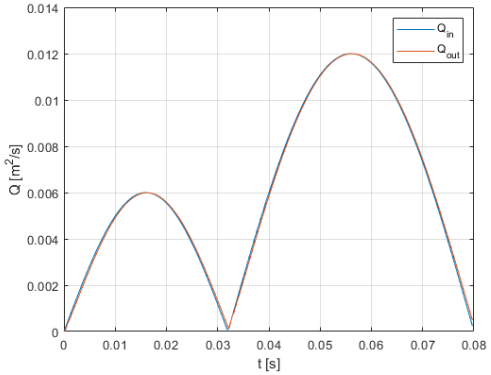


(d) *Inflow (purple) and outflow (green) pressures with correction.*

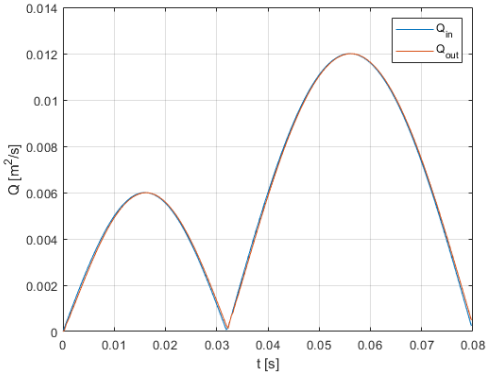


(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

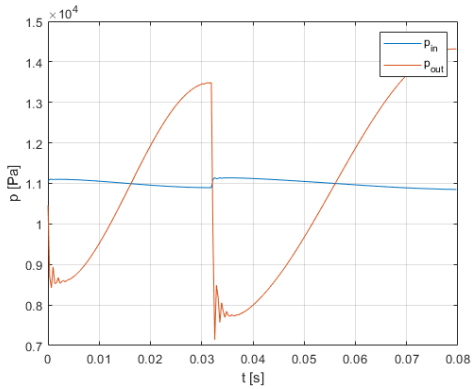
Figure 5.12: Homogeneous Neumann conditions on all boundaries, case $RR = 0.008\,s$.



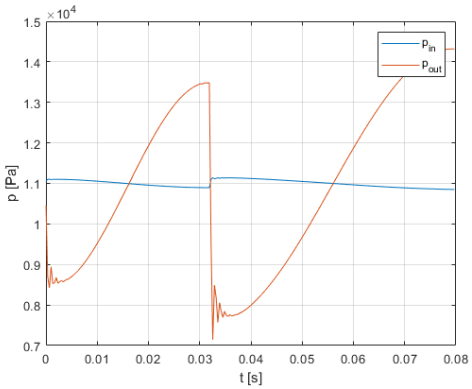
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



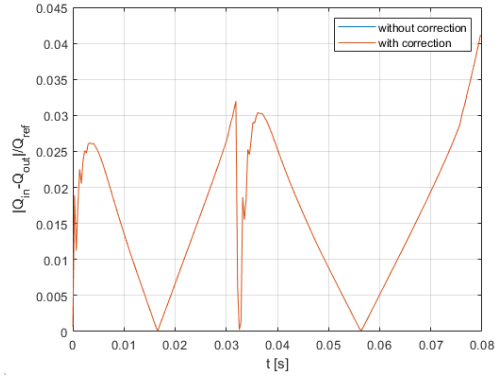
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

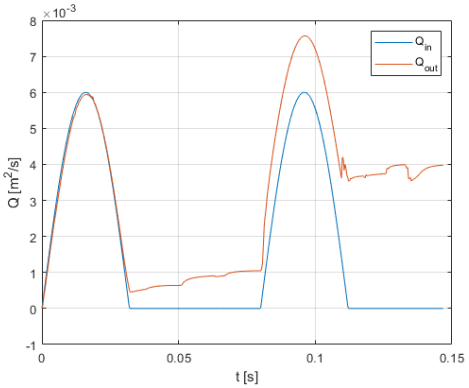


(d) *Inflow (purple) and outflow (green) pressures with correction.*

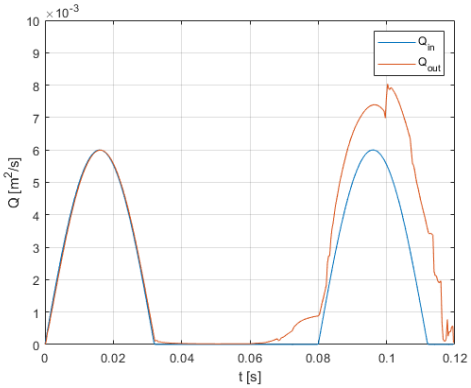


(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

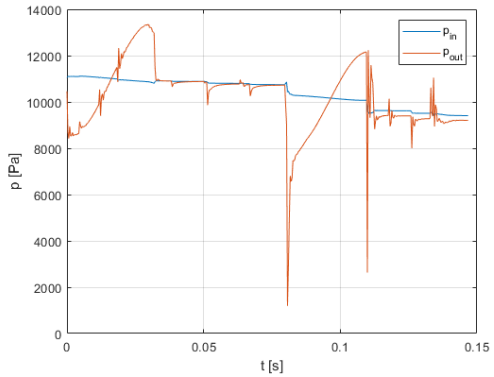
Figure 5.13: Homogeneous Neumann conditions on all boundaries, case $RR = 0.08\,s$ with *superpulse*.



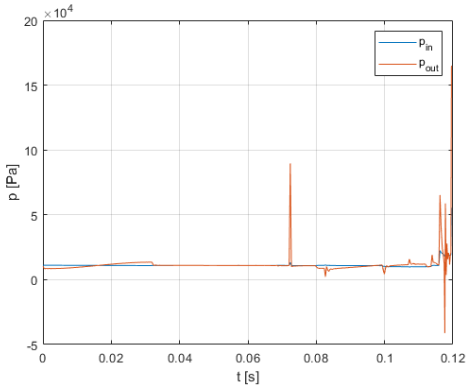
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



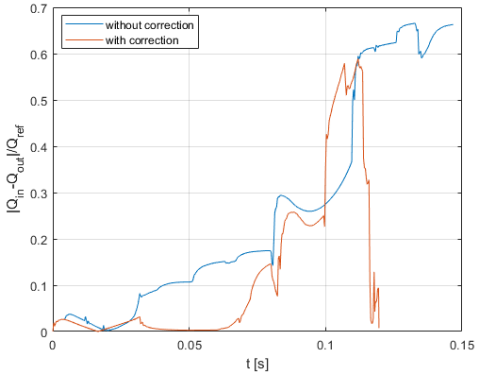
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

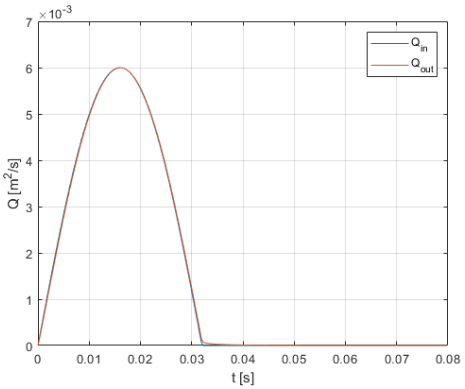


(d) *Inflow (purple) and outflow (green) pressures with correction.*

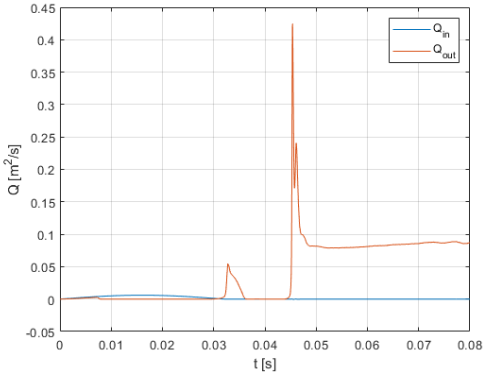


(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

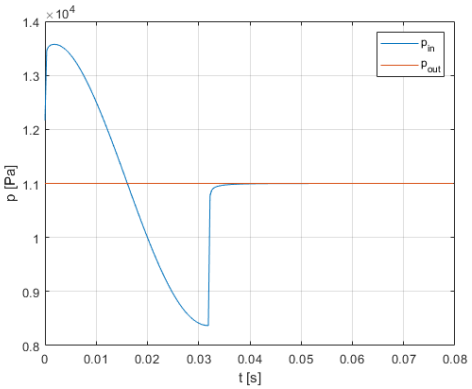
Figure 5.14: Homogeneous Neumann conditions on all boundaries, case $RR = 0.08\,s$ with low Krylov's solver tolerance (10^{-12}).



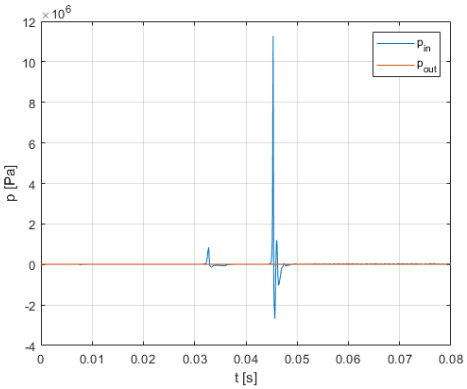
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



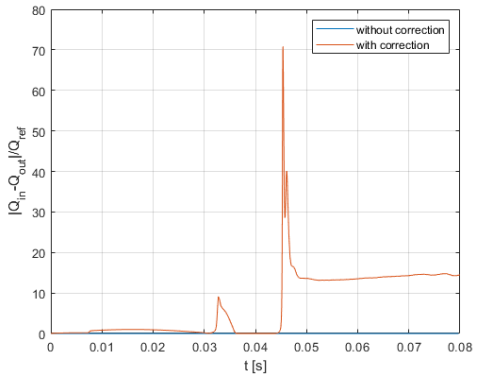
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

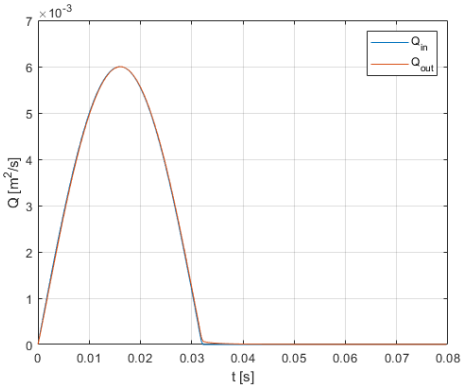


(d) *Inflow (purple) and outflow (green) pressures with correction.*

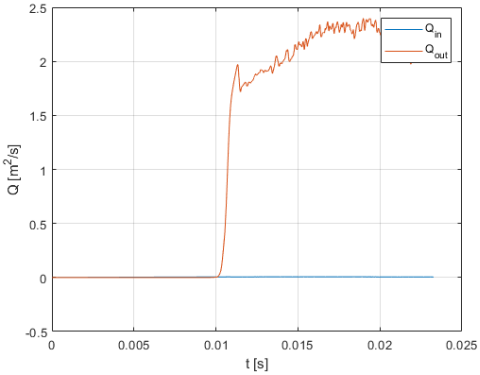


(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

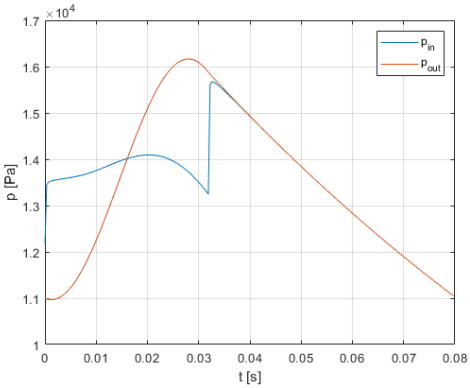
Figure 5.15: Constant Dirichlet conditions on the outlet boundary, case $RR = 0.08\,s$.



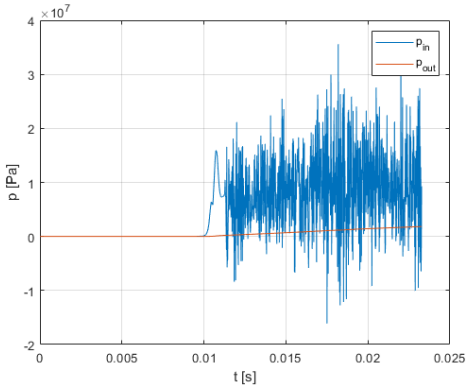
(a) *Inflow (purple) and outflow (green) mass-rates without correction.*



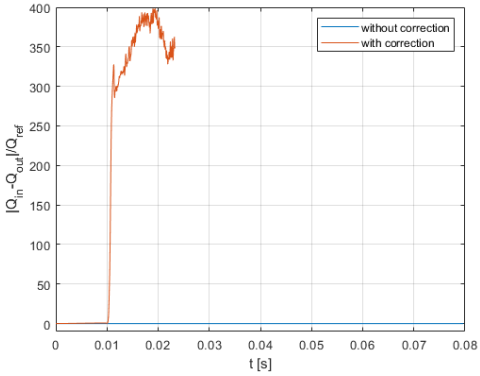
(b) *Inflow (purple) and outflow (green) mass-rates with correction.*



(c) *Inflow (purple) and outflow (green) pressures without correction.*

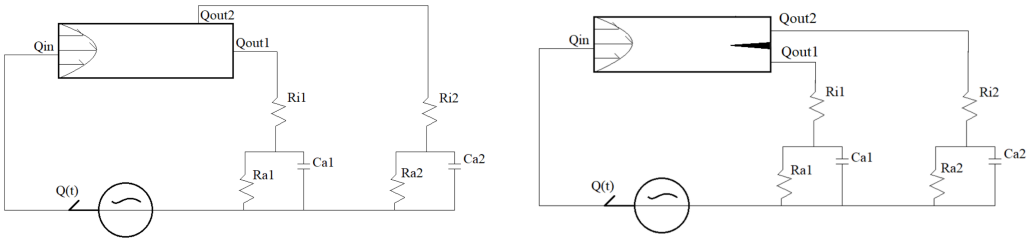


(d) *Inflow (purple) and outflow (green) pressures with correction.*



(e) *Flow-rates relative error (purple line: no correction; green line: correction).*

Figure 5.16: WK3 conditions on the outlet boundary, case $RR = 0.08\,s$.



(a) *Double exit WK3 application.* (b) *Iliac bifurcation WK3 application.*

Figure 5.17: Examples of WK3 application to a double exit aortic channel.

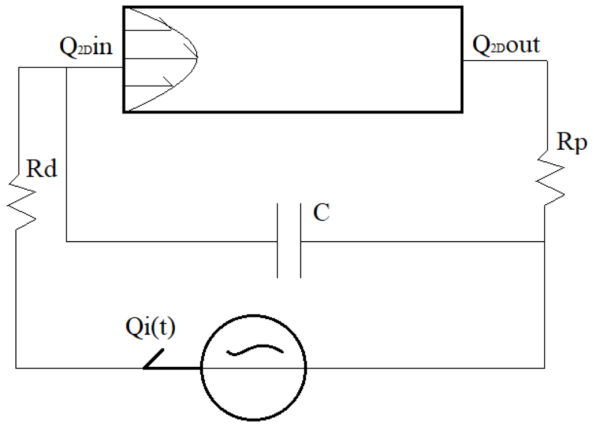
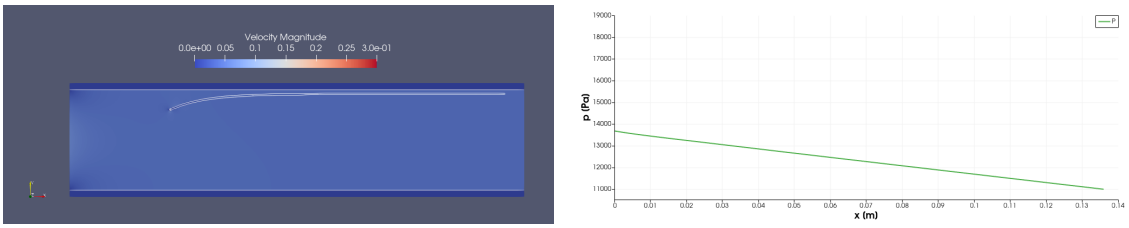


Figure 5.18: Closed-loop configuration of the multiscale model.

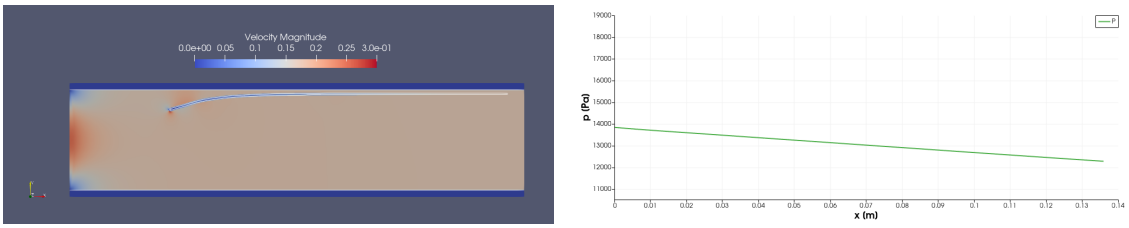
domain, in figure 5.20, and subsequently, the illustration of an entire systolic phase is reported in figure 5.21.

5.4 Compliant Aortic Models

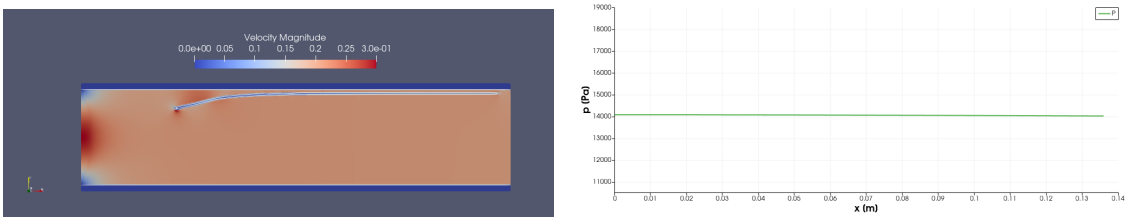
In this section, some different configurations still exploiting the newly developed 3-elements Windkessel model as outflow boundary conditions are presented, in order to maintain the well-posedness of the problem, but with the aim, on the other hand, of further improving the aortic mechanics, and get it closer to the reality. We had the purpose to take into account the compliant behaviour of large and elastic vessels, linked to their own structure which allows for important deformations of the channel's structure, thus storing a given quantity of blood during each systolic phase of the heart beat, and then releasing it, secondly, during the following diastolic phase. This way, the previous strongly pulsating blood flow signal is progressively regularized by the mechanics of the vessel, as well as the associated pressure signal is too, leading to a increasingly constant flow, which is more suitable to supply, consequently, all human tissues and organs linked to the peripheral circulation. This compliant mechanics, thus, is reproduced on our simple model domain by adding additional elastic bodies to replace part of the external rigid aortic walls. Several



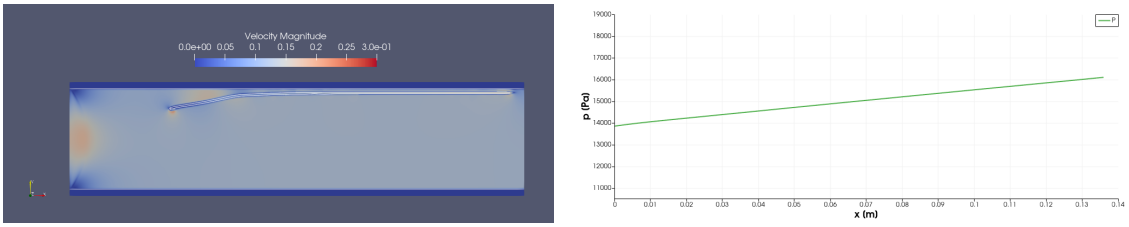
(a) $t = t_1$



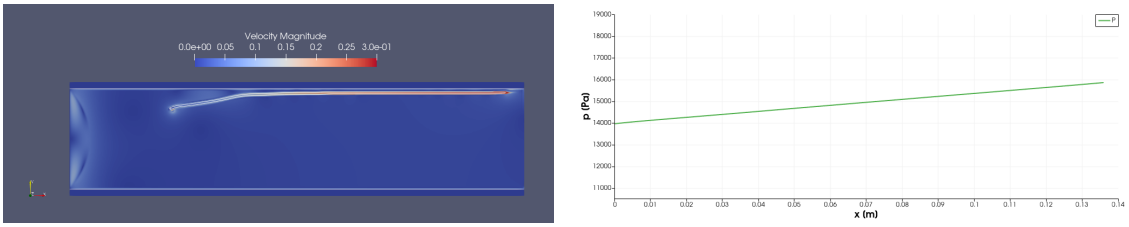
(b) $t = t_2$



(c) $t = t_3$

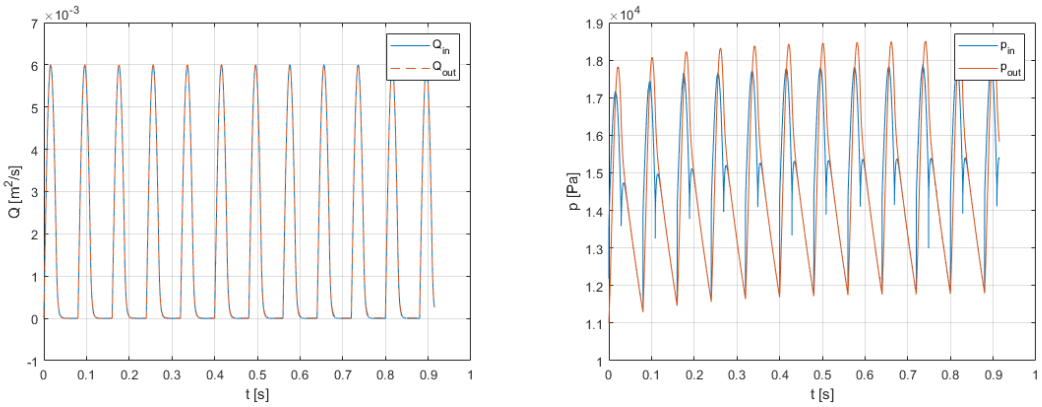


(d) $t = t_4$



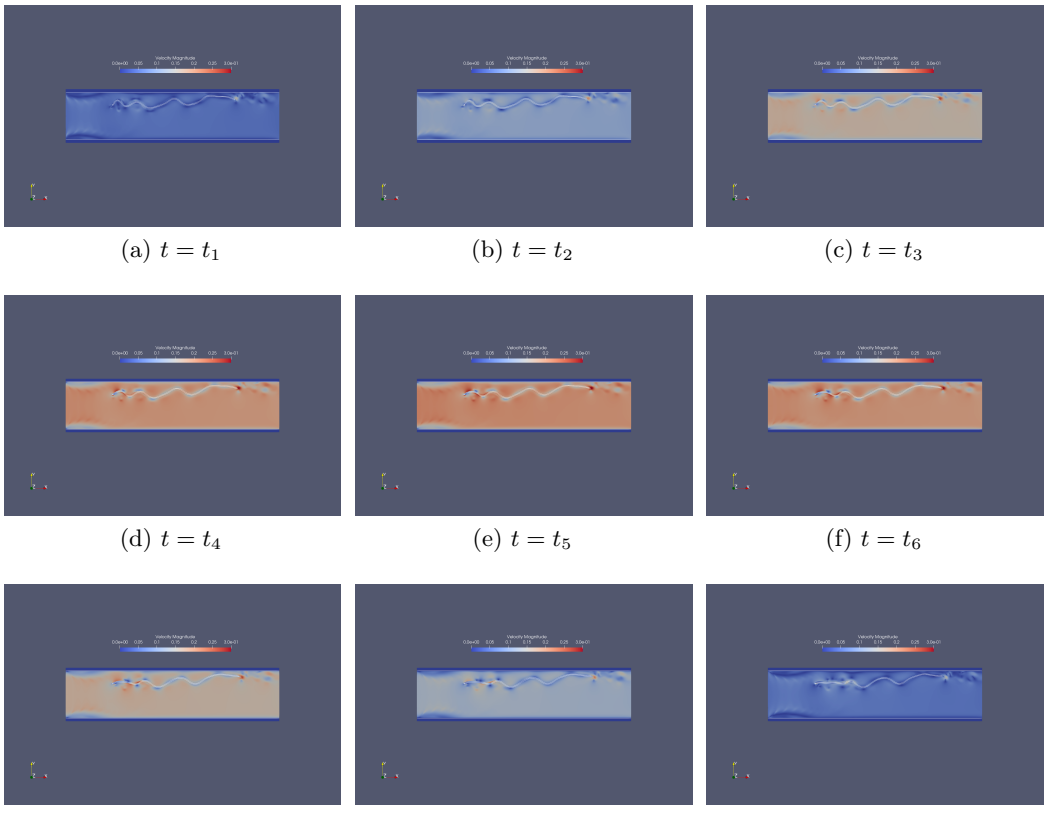
(e) $t = t_5$

Figure 5.19: Example of first systole with floating membrane and WK3 outflow conditions on pressure.



(a) *Flow-rate signal.* (b) *Pressure signal.*

Figure 5.20: Flow-rate and pressure time signals associated with the employment of WK3-type outflow boundary conditions (purple: inflow, green: outflow).



(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$

(d) $t = t_4$ (e) $t = t_5$ (f) $t = t_6$

(g) $t = t_7$ (h) $t = t_8$ (i) $t = t_9$

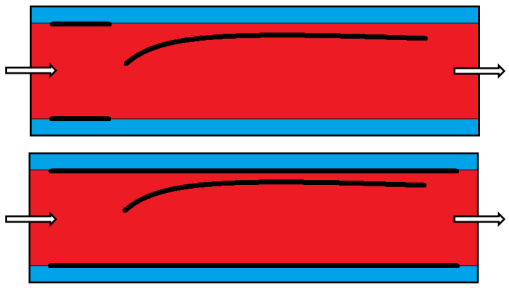
Figure 5.21: Snapshots of a heart beart with application of WK3-type outflow boundary conditions.

attempts have been carried out focusing on this primary goal, and some results will therefore be shown within this section, to come then, finally, to the definitive selected one, chosen to be the most suitable for our case of study.

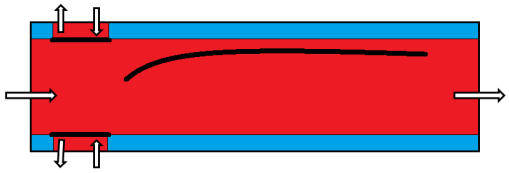
Basically, the idea is to create a new configuration for our problem, similarly to what was shown at the beginning of chapter 4, more precisely in figure 4.3, which, differently from what was depicted instead in figure 4.2, includes moving and deforming aortic walls composing the external aortic structure of our reproduced vessel. To give an idea, we tried to create simulations either by adding two new short membranes in front of the floating central body, leaving unaltered the following structure of the channel, or by designing two longer ones, covering almost the entire walls, from upstream to downstream, with respect to the floating central membrane.

Initially, we thought to design such additional membranes by assigning them the same thickness we adopted for the floating *tunica intima*, but this choice turned out to be not very satisfactory, since the stiffness of those bodies was not sufficiently elevate to allow for realistic behaviours. Thus, on the one hand, we decided to separate the level-Set functions associated with those new elastic bodies from that corresponding to the central floating membrane, so that in the end we dealt with two distinguished scalar functions $\phi_{e,mem}$ and $\phi_{e,wall}$. This way, we could assign different values of stiffness to such bodies, by changing the parameters involved in the hyperelastic Mooney-Rivlin model. On the other hand, one could also resort to an increment of the wall thickness, by designing elastic membranes with higher diametral width. Furthermore, it is not sufficient to design two membranes attached to the rigid upper and lower walls, where the penalisation algorithm is applied (see section 4.2), since their displacement would be nullified by the immersed boundary treatment itself, involving also all the closest fluid cells. Some attempts have therefore been performed by adopting membranes with width equal to the entire walls, but their excessive rigidity did not allow for any sort of deformation. Consequently, it was pretty clear the necessity for designing additional fluid chambers beyond those membranes, to let them move and deform freely and no longer be affected by the penalised rigid walls. However, the penalisation technique was still employed to keep the holder of the floating *tunica intima* fixed, and to maintain the new deforming external membranes attached to the remaining rigid portion of the aortic walls, still describing the very external boundary condition of the domain, and delimiting the newly introduced additional fluid chambers. Accordingly to a schematic representation of what has been discussed so far, we propose, in figure 5.22, respectively, the design associated with the very first attempt involving the two membranes attached to the aortic walls, to proceed than with the introduction of the external fluid chambers, with open boundaries, not very appropriate at all since there is no way to allow for the fluid to exit from the domain (applying a do-nothing condition, for instance) and then to come back freely, numerically speaking; then, more elaborate configurations are shown, involving extended fluid chambers beyond the aortic walls, but not connected with the central channel and therefore leading to undesired deformations of their terminal parts, since the incompressibility of the reservoir fluid needs to be respected; finally, the best option showing a connection between the additional chambers and the central channel is proposed, which will be right the one adopted for the following purposes of this work (notice that even longer membranes could be employed in

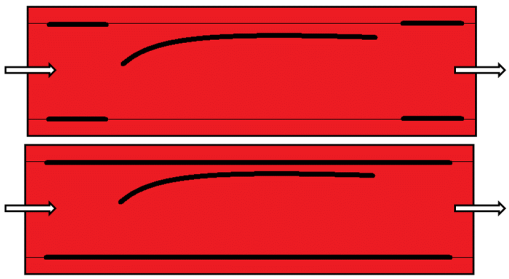
this case). As a remark concerning the method techniques discussed within the previous chapter, consider that the 1st order penalisation is employed to the aim of fixing the moving elastic membranes, representing the new compliant aortic walls, to the remaining rigid wall separating the central channel from the behind standing reservoir chambers, by means of overlapping them. As an example of results, consider the solution snapshots depicted in figure 5.23, where a double pair of membranes has been employed, and compare them to the associated time flow-rate and pressure signals reported in figure 5.24, clearly exhibiting the compliant regularising behaviour described above on the plotted quantities, as the injected blood flows throughout the deforming channel. As done so far, WK3-type boundary conditions are employed on the outflow section. More interesting examples of application of such deforming walls will be shown in detail, together with their effect on the fluid mechanics within the domain, in the following last section.



(a) *Attached membranes configuration.*



(b) *Open-boundary reservoir chambers configuration.*



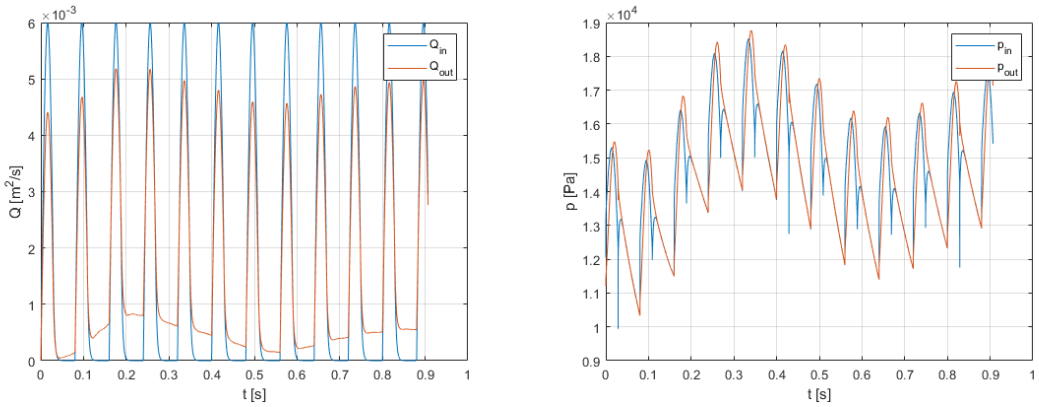
(c) *Closed reservoir chambers configuration.*

(d) *Open reservoir chambers configuration.*

Figure 5.22: Schemes of aortic compliant walls configuration (red: blood flow, blue: penalised wall).

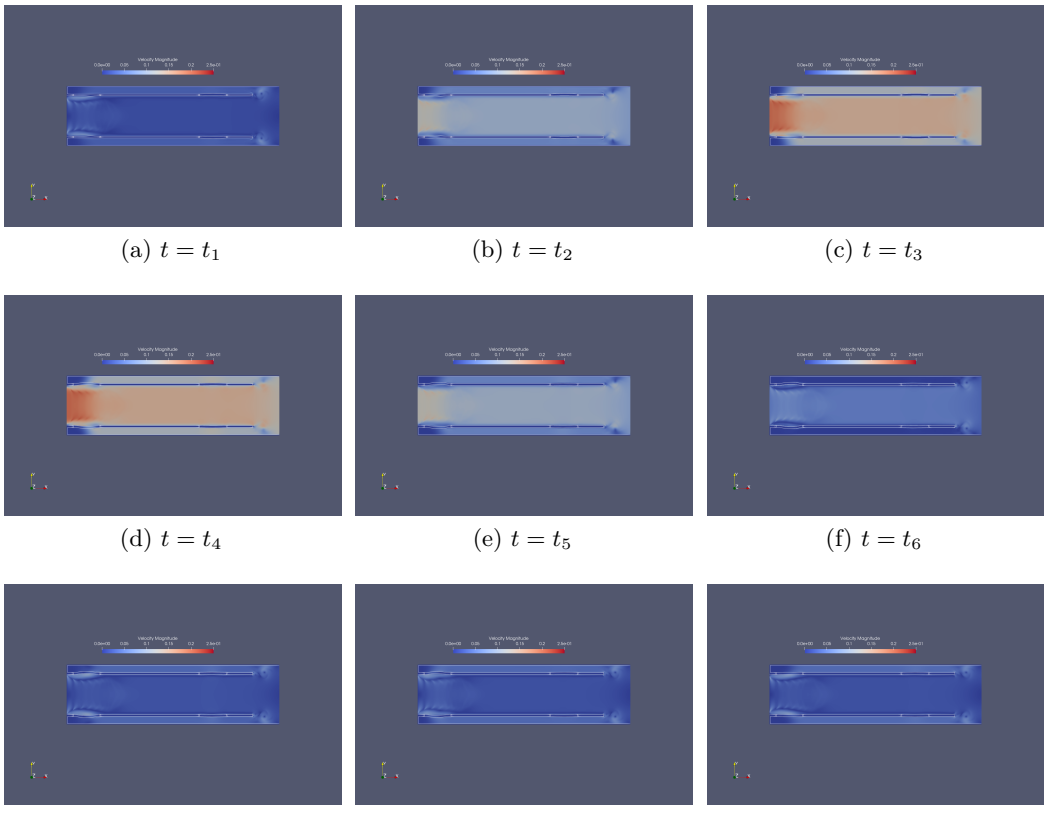
5.5 Final Closed-Loop Configuration

Within this last, concluding part of the chapter, and of this thesis as well, we have the purpose of showing the most interesting and complete results obtained by employing the closed-loop configuration of the multiscale-model already presented and depicted in figure 5.18. As stated, this configuration allows for an actual feedback given by the 0D model on the CFD higher-dimensional one representing our aortic channel branch



(a) *Flow-rate signal.* (b) *Pressure signal.*

Figure 5.23: Time waveshapes of the inflowing (purple) and outflowing (green) flow-rate and pressure signals.



(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$

(d) $t = t_4$ (e) $t = t_5$ (f) $t = t_6$

(g) $t = t_7$ (h) $t = t_8$ (i) $t = t_9$

Figure 5.24: Snapshots of the solution concerning a compliant vessel realised with a double pair of elastic membranes and WK3 outflow boundary conditions.

where the type B dissection is occurring. Clearly, the capacitor inserted in parallel with respect to the 2D domain is responsible for collecting blood flow during the systolic phase of the cardiac cycle, and for its subsequent release during the following diastolic phase, both dictated by the generator acting as a pump and positioned upstream, whose flow-rate waveshape is the usual sinus one already proposed in all the previous simulations. Therefore, an augmented compliant behaviour is possible to be taken into account, allowing us then to mimic the compliant effect of all upstream large vessels on the represented branch including the dissection, and to study, thus, the consequences on the disease.

As it regards the model of the 0D circuit and its coupling with the 2D domain, giving rise to a multi-D closed loop configuration, the equations for the capacitor and the resistors are the same already reported in section 2.5, and their numerical treatment follows the same idea presented in section 3.2.3. A basic idea is provided here below, let us call Q_i, Q_C and Q_{2D}, respectively, the mass-rate flowing throughout the main branch and ejected by the pump, the one within the capacity's branch and, finally, that throughout the 2D domain's one. Then, by assuming the constitutive law of the capacitor, according to which $Q_c = C\frac{\mathrm{d}p_C}{\mathrm{d}t}$, and thanks to the Kirchhoff's current law, we have that:

$$Q_{2D} = Q_i - C\frac{\mathrm{d}p_C}{\mathrm{d}t} = Q_i - C\frac{\mathrm{d}(p_{2D} + p_{R_p})}{\mathrm{d}t}$$

Therefore, being $p_C = p_{2D} + p_{R_p} = Q_{2D}(R_{2D} + R_p)$, where one can assume the Poiseuille's law to hold, for simplicity, that is, considering $R_{2D} = \frac{2L_x\mu}{\pi R^4}$, we have thus that, by adopting as usual an explicit Euler time discretization scheme 3.16:

$$Q_{2D}^{k+1} = \frac{Q_i^{k+1} + C(R_p + R_{2D})\frac{Q_{2D}^k}{\Delta t}}{1 + \frac{1}{\Delta t}C(R_p + R_{2D})}$$

Whereas, as it regards the pressure imposed as outflowing Dirichlet condition for the corresponding Poisson's problem, widely discussed in section 3.3.3, we can simply apply it by considering the constitutive law of resistors as:

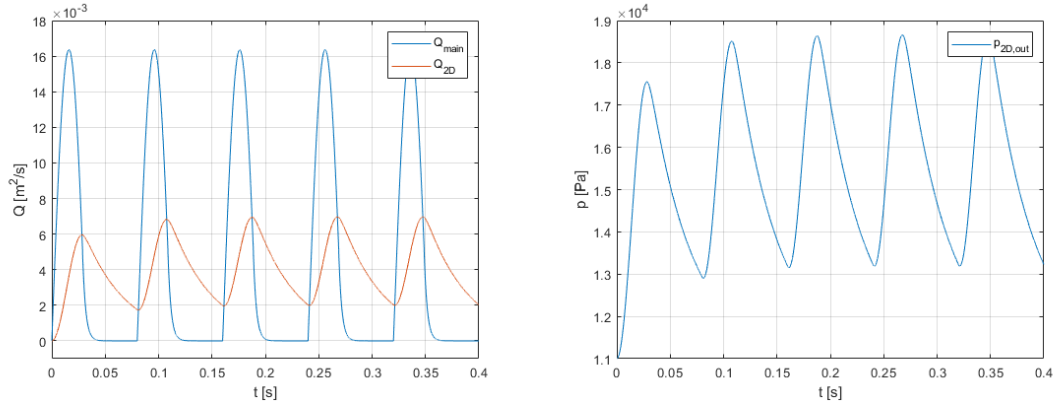
$$p_{out}^{k+1} = Q_{2D}^{k+1}R_p, \quad \text{then:} \quad \Phi_{out}^{k+1} = \Delta t(p_{out}^{k+1} - p_{out}^k)$$

This way, the 0D model has been defined, discretized and coupled with the 2D one, and as far as the tuning procedure of its parameters is concerned, by assuming an injection flow-rate Q_i with a sinusoidal time behaviour as described by the usual law 3.50, with peak at $Q_{i,max} \approx 1.6 \times 10^{-2}\, m^2/s$, in order to keep a reasonable value of the inflowing $Q_{2D,max} \approx 6.0 \times 10^{-3}\, m^2/s$, one can find them listed in table 5.2. The above cited tuning

Table 5.2: Tuning parameters adopted for the closed-loop configuration.

Component	Tuned value	$Q_{i,max}\,[m^2/s]$	$RR\,[s]$
$R_p\,[kg/(sm^4)]$	1.1×10^6	1.6×10^{-2}	0.08
$C\,[m^4s^2/kg]$	3.6×10^{-8}	1.6×10^{-2}	0.08
$R_d\,[kg/(sm^4)]$	6.5×10^5	1.6×10^{-2}	0.08

procedure has been carried out by defining a completely simplified 0D model, where the 2D domain was replaced by a resistor R_{2D} defined as previously, and whose results, in terms of outflowing pressure $p_{out}(t)$ and inflowing mass-rate $Q_{2D}(t)$, compared with the main pumped $Q_i(t)$, are shown in figure 5.25.



(a) *Main (purple) and 2D inflow (green) mass-rate time waveshapes.*

(b) *2D outflow pressure time waveshape.*

Figure 5.25: Tuned solution obtained via the complete 0D closed-loop model.

Thus, in terms of results, three different cases are proposed in the following, all exhibiting a closed-loop configuration, such as the one here presented, and listed here below:

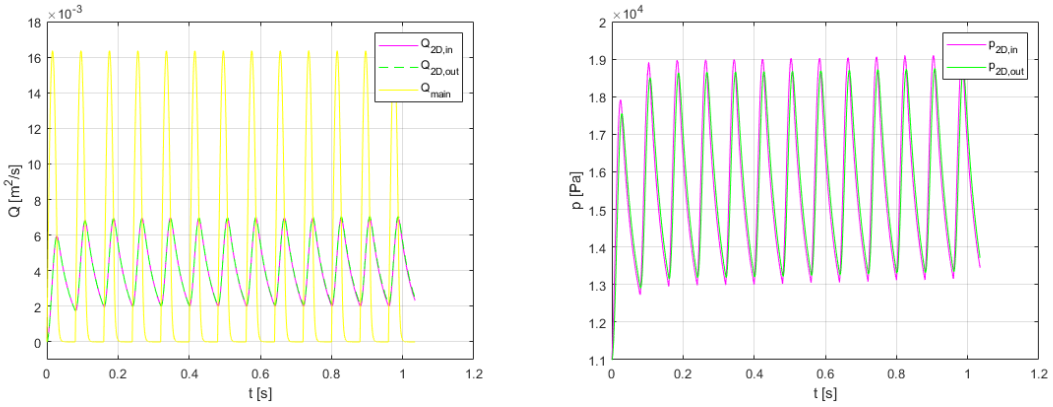
- freely-floating *tunica intima* membrane with rigid aortic walls;
- compliant elastic aortic walls with no floating *tunica intima* membrane;
- complete study with both compliant elastic aortic walls and floating *tunica intima* membrane.

For all of them, the smoothed time heart-beating profile has been adopted, together with the 10th degree polynomial spatial distribution for the x-wise component of the velocity field, on the inflow section, which have been chosen accordingly to what had been discussed in the previous section 5.2.

Floating Membrane:

The first example we report within these last paragraphs includes a long floating membrane representing the *tunica intima* resulting from the dissection, initially placed according to an exponential profile as previously discussed, and where rigid aortic walls are adopted as external constraints for the blood flow. The membrane is fixed to a holder from its left side, and it is let free to move from the right one, causing, as it can be noticed from the solution, the formation of vortex structures evolving towards the outflow section, potential sources of noise. First, we add the two representations of the waveshapes associated with the flow-rates and the pressures registered, respectively, at the pump level of the circuit

(the blood mass ejected by the heart), and on the inflow and outflow sections of the 2D domain; then, some snapshots of the solution are depicted in figure 5.27, and, in particular, in figure 5.28 is reported the detailed velocity profile extracted from two cross-sections of the channel, at a given time, in order to show the structure associated with the boundary layer and its description. As it can be noticed, the solution corresponding to the here



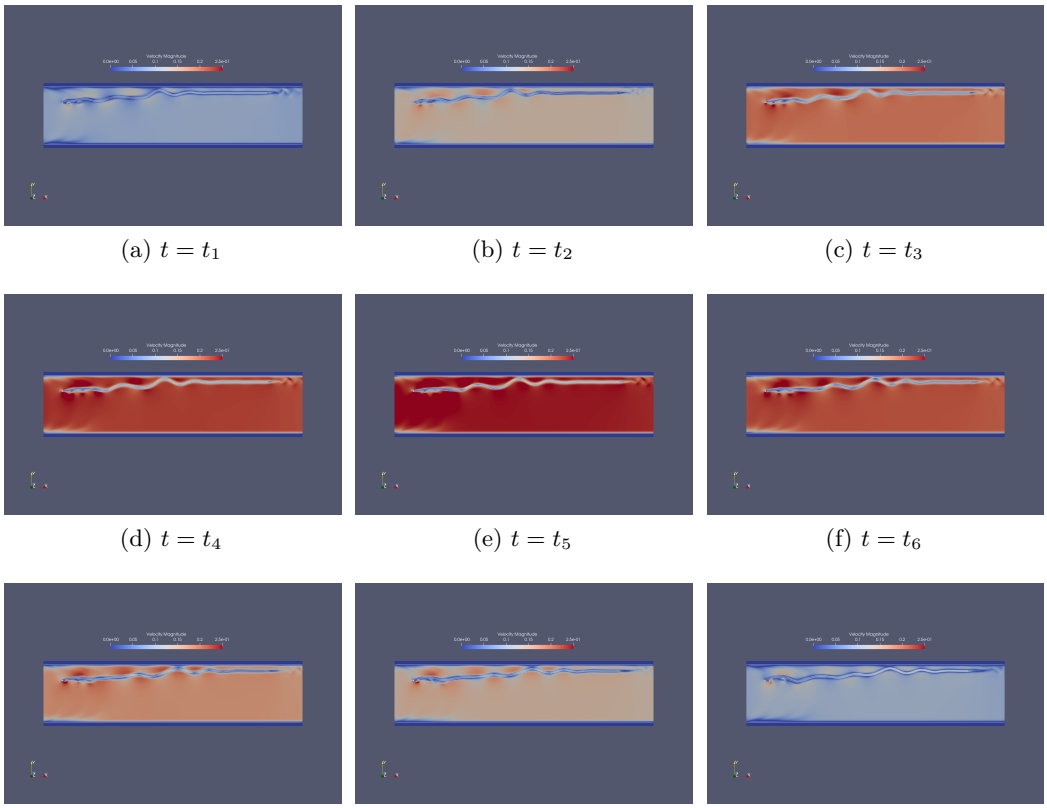
(a) *Flow-rate waveshapes.* (b) *Pressure waveshapes.*

Figure 5.26: Time waveshapes of mass-flow and pressure for the floating membrane case with rigid external walls, closed-loop configuration.

described case is still in good agreement with that extracted from the 0D domain, showing a perfect conservation of mass through the simulated aortic channel, accordingly to the fact that rigid walls have been employed in it, and thus no additional compliant effects are reproducible. Conversely, the compliant effect due to the external capacitor makes the flow inside the 2D domain more regular and continuous with respect to that ejected by the heart pump, affecting the behaviour of the membrane, in comparison to what had been shown with regards to the previous solutions, employing a simple WK3 model as boundary conditions with no feedback on the higher-dimensional domain.

Compliant Elastic Walls:

Thanks to the following simulation, the compliant effect due to elastic walls can be taken into account and showed as an interesting result. As explained, the deformation of the walls allows the blood to be progressively stored during systole, and then released during diastole, enhancing the regularization of the mass-flow and pressure signals already affected by the external capacity of the multi-D model. However, an initial transitory has to be considered, in order to allow for the membranes representing the elastic aortic walls to assume their deformed configuration and to correctly give rise, then, to the mentioned behaviour. This initial transitory lasts about 10-11 beats, and only once it has been completely exhausted the expected behaviour can be appreciated. In figure 5.29 the waveshapes associated with mass-flow and pressure are reported, once again, corresponding to the blood mass ejected by the heart, and to the mean value computed at the inflow and outflow sections of the 2D domain, together with some zoomed images of those curves.



(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$

(d) $t = t_4$ (e) $t = t_5$ (f) $t = t_6$

(g) $t = t_7$ (h) $t = t_8$ (i) $t = t_9$

Figure 5.27: Snapshots of the solution involving a floating membrane immersed in a blood flow with rigid boundaries, closed-loop configuration.

Additionally, in figure 5.30 is reported the progressive time average of the inflowing and of the outflowing mass-rates, obtained as:

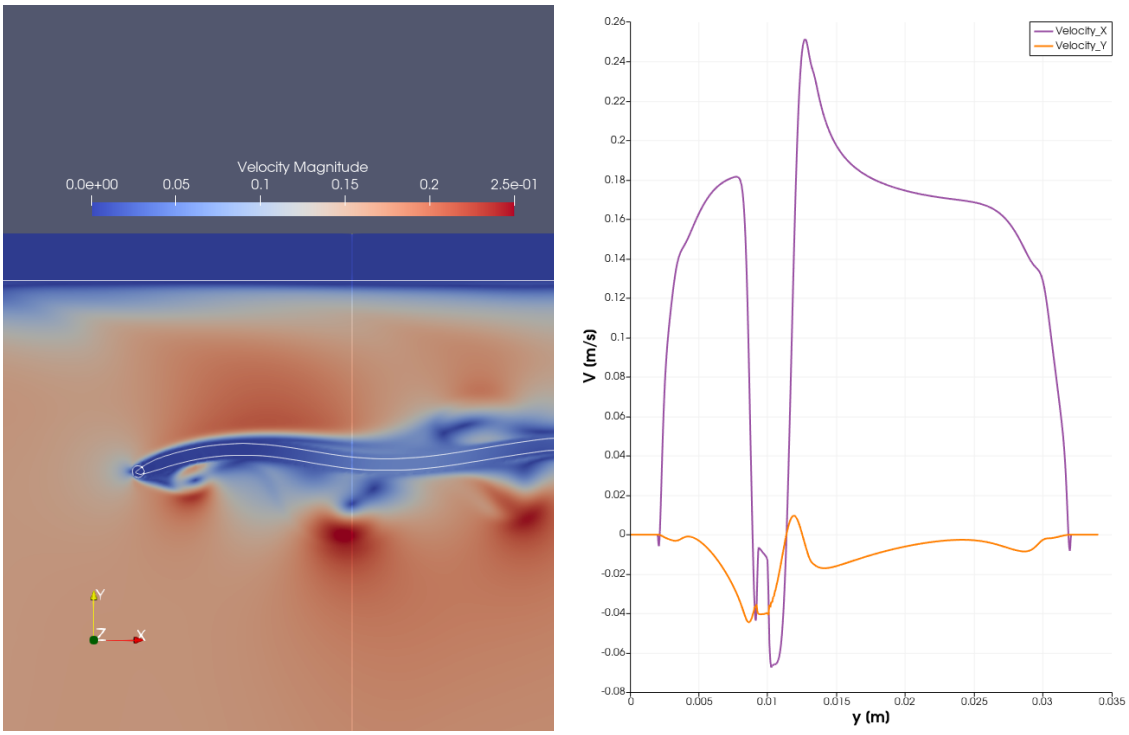
$$\bar{Q}_{2D,in} = \frac{1}{T - t_0} \int_{t_0}^{T} Q_{2D,in}(t)\, dt, \quad \text{and} \quad \bar{Q}_{2D,out} = \frac{1}{T - t_0} \int_{t_0}^{T} Q_{2D,out}(t)\, dt$$

Clearly, those curves should progressively tend to coincide perfectly as the time runs, but evidently the performed simulation was not long enough to reach that condition. Probably, if the pumping heart was stopped, the released blood flow would lead the two curves to overlap quickly. Then, in figure 5.31 some snapshots of the solution are provided, as usual, showing the interesting configuration assumed by the elastic walls in time.

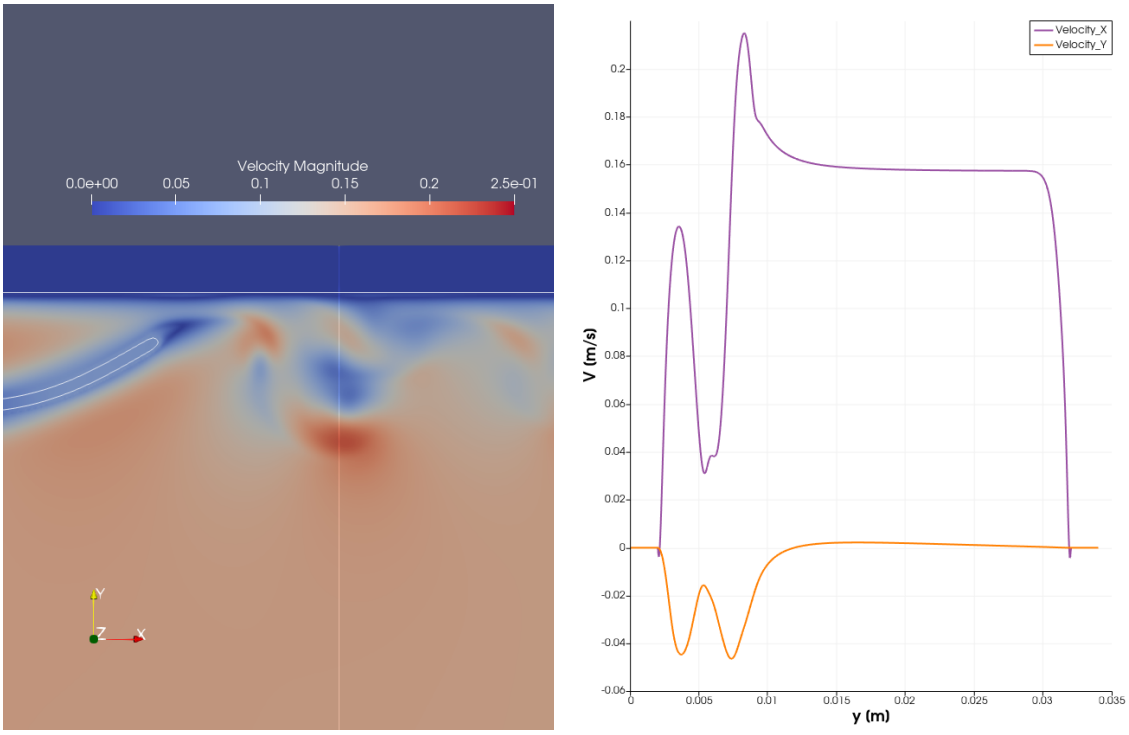
Floating Membrane & Compliant Elastic Walls:

This last paragraph puts together what has been shown within the previous two ones, leading to a solution where both elastic compliant aortic walls and the floating *tunica intima* membrane are present. There is not much more to say in addition to what has already been observed about the features of such simulations, since the same approaches

are adopted regarding the coupling between the 0D and the 2D models, the expectation for the time averaged flow-rates to coincide, once the initial transitory regime has been overcome, or even the possibility to appreciate the formation of vortexes surrounding the most disturbed zones where the entry and the exit tear are represented. One last remark that could be worth making is the fact that, differently to what has been done with respect to the previous case where only the elastic walls were involved, letting alone the floating membrane, here the statistically stationary condition is far from being reached, since the computational effort would have been too high if 10-11 beats had been simulated. This is due some instabilities which could arise on the inflow section when a too high domain partitioning is performed, with the aim of parallelizing the problem. Thus, a reduced number of nodes needed to be employed, making it difficult to obtain considerable results within a reasonable amount of time. In figure 5.32 the flow-rate and pressure waveshapes are reported, even for this last case, whereas in figures 5.33 and 5.34, respectively, some snapshots of the solution together with two more detailed situations including vortexes formation are provided.

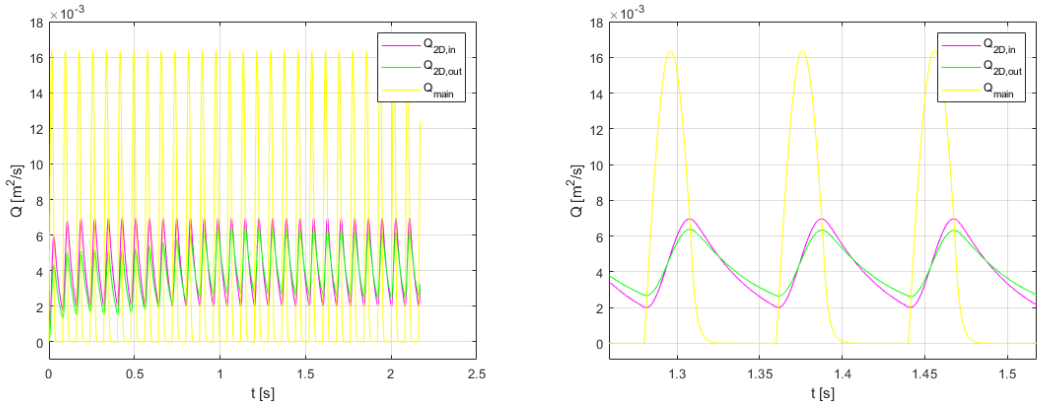


(a) *Entry tear vortexes.*

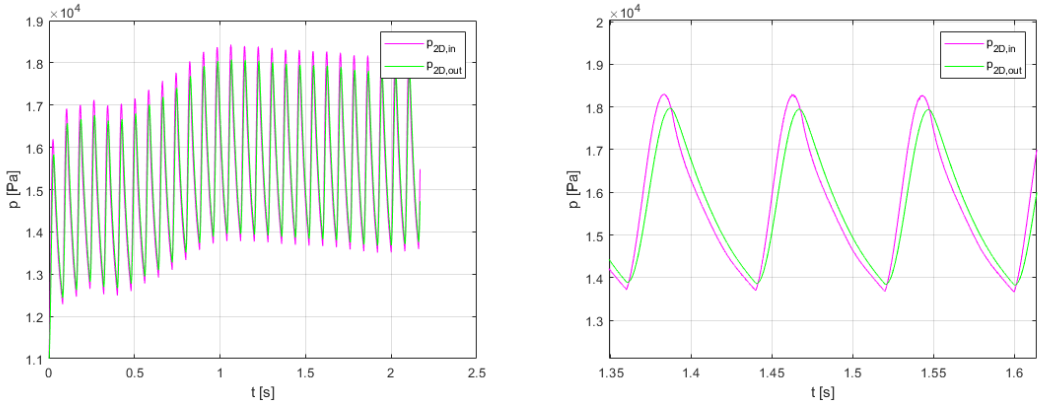


(b) *Exit tear vortexes.*

Figure 5.28: Velocity profile extracted at two different x positions.



(a) *Flow-rate waveshape (zoomed on the right).*



(b) *Pressure waveshape (zoomed on the right)*

Figure 5.29: Time waveshapes of mass-flow and pressure for the moving walls case without floating membrane, closed-loop configuration.

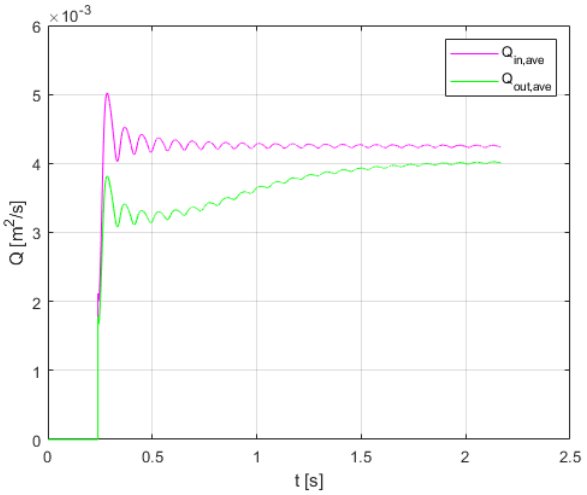
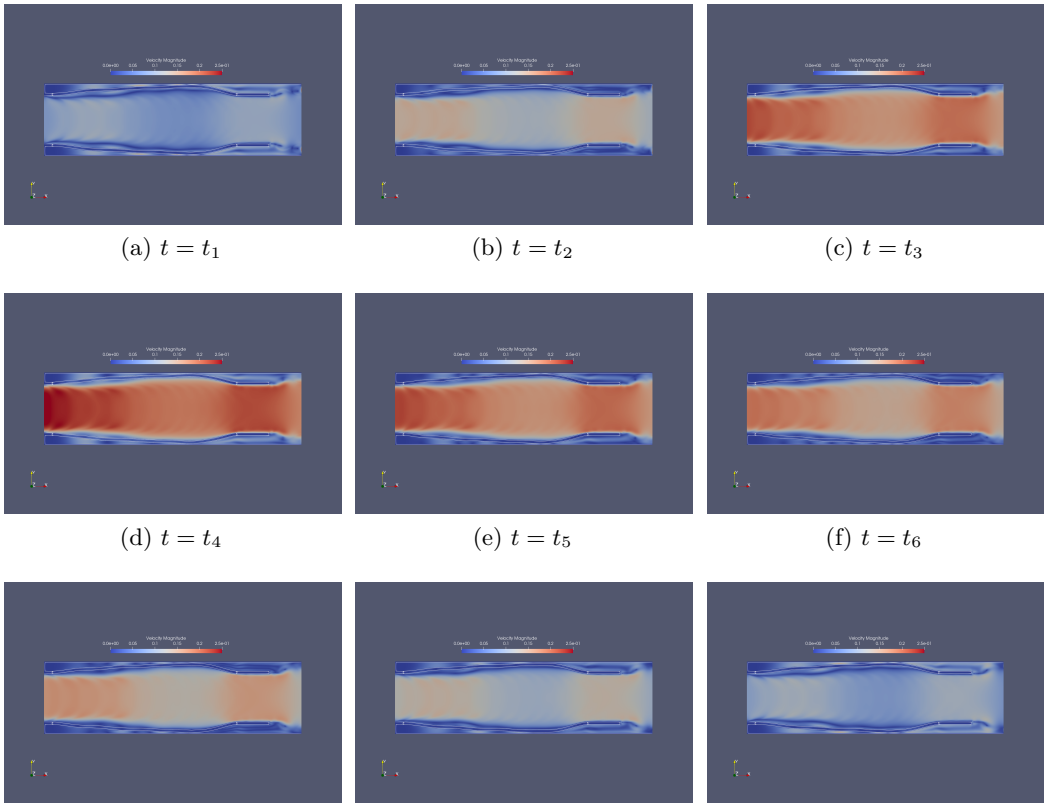


Figure 5.30: Time averages of inflowing (purple) and outflowing (green) 2D mass-rates

.

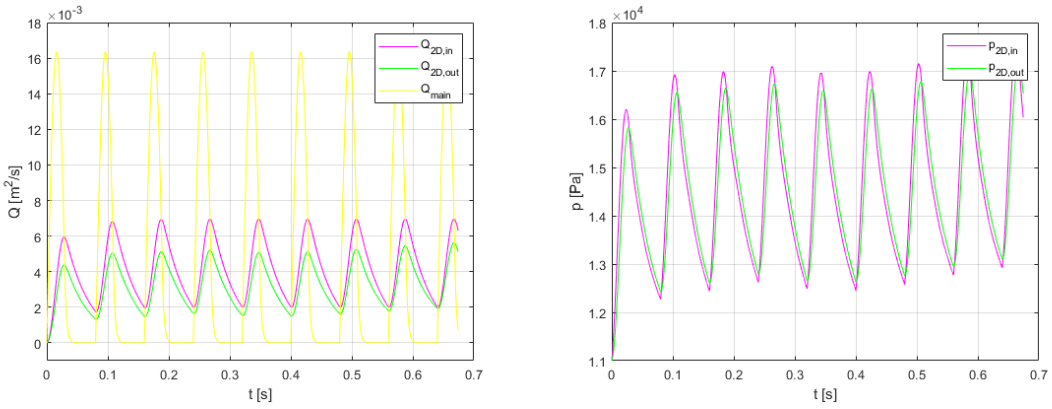


(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$

(d) $t = t_4$ (e) $t = t_5$ (f) $t = t_6$

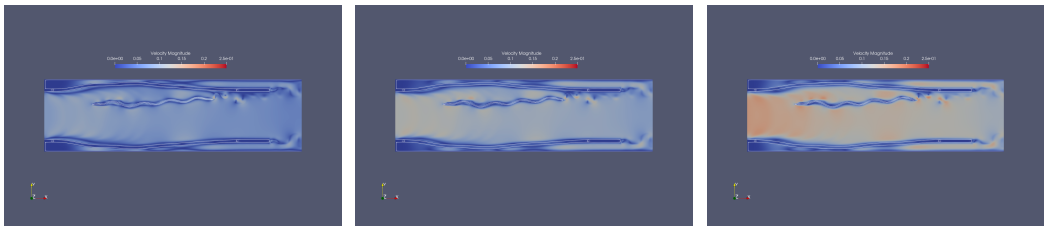
(g) $t = t_7$ (h) $t = t_8$ (i) $t = t_9$

Figure 5.31: Snapshots of the solution involving elastic aortic walls without floating membrane, closed-loop configuration.

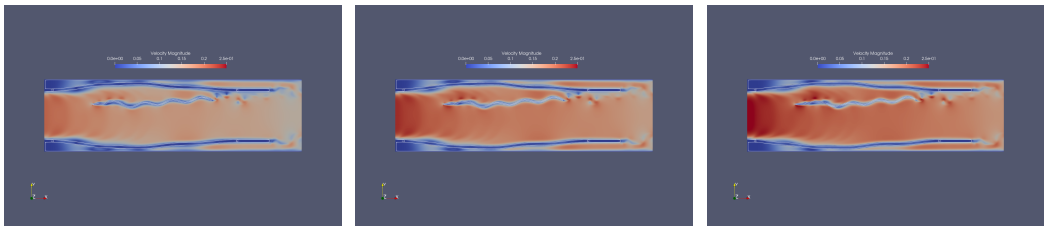


(a) *Flow-rate waveshapes.* (b) *Pressure waveshapes.*

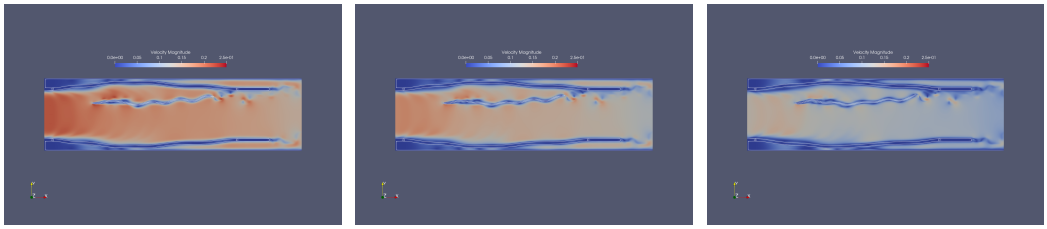
Figure 5.32: Time waveshapes of mass-flow and pressure for the moving walls case with floating membrane, closed-loop configuration.



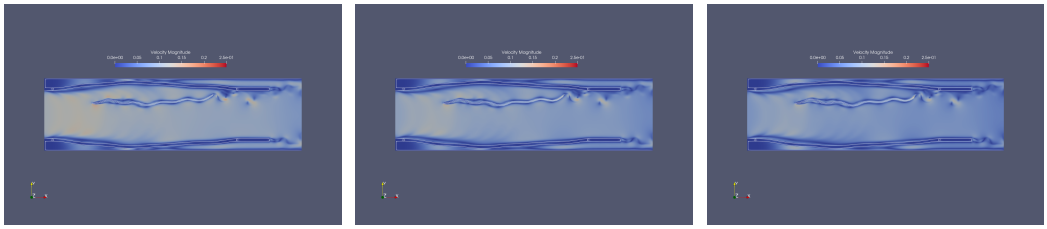
(a) $t = t_1$ (b) $t = t_2$ (c) $t = t_3$



(d) $t = t_4$ (e) $t = t_5$ (f) $t = t_6$

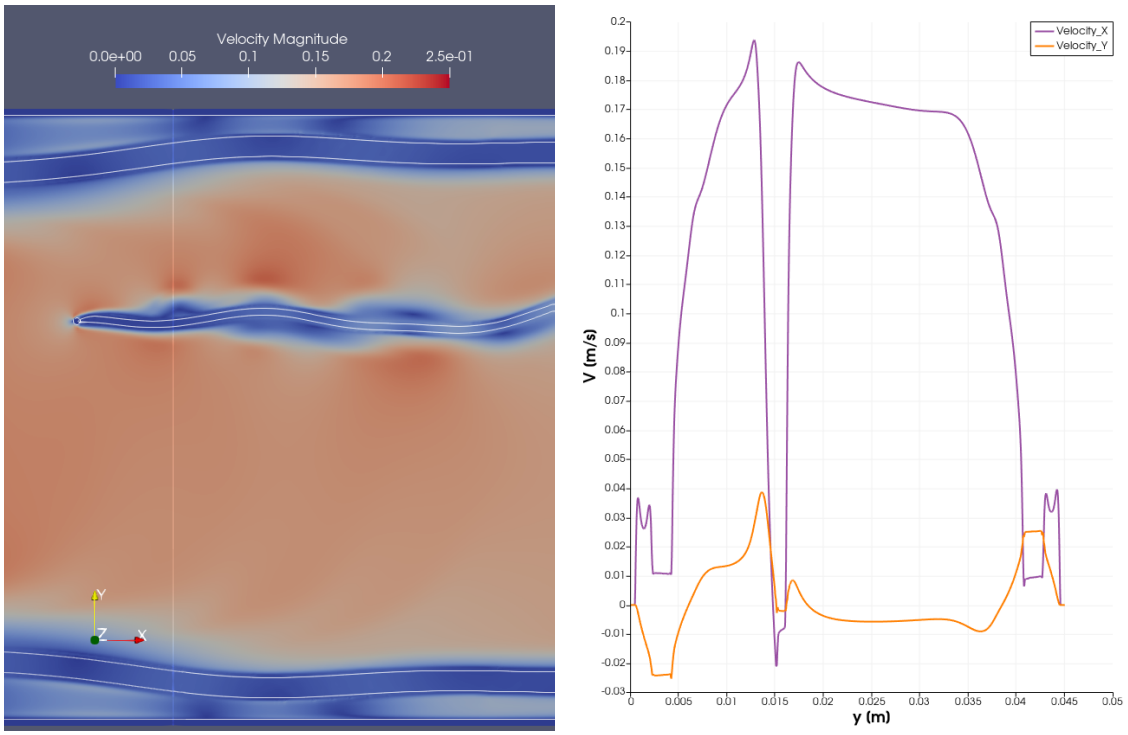


(g) $t = t_7$ (h) $t = t_8$ (i) $t = t_9$

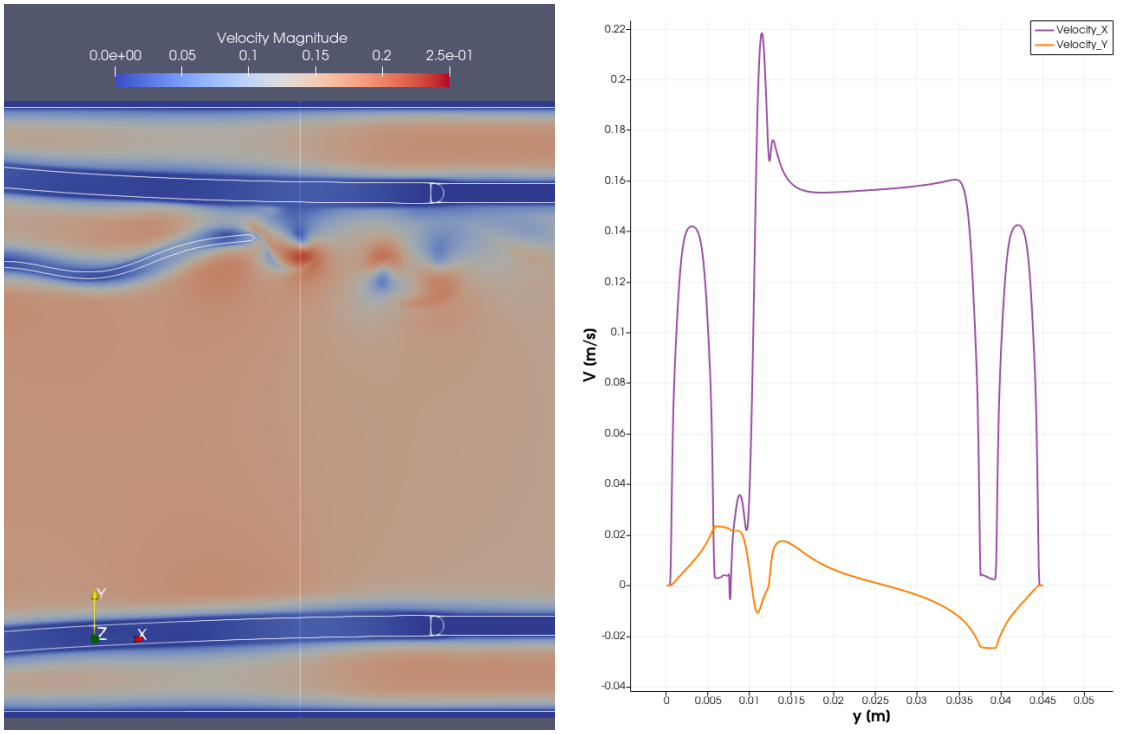


(j) $t = t_{10}$ (k) $t = t_{11}$ (l) $t = t_{12}$

Figure 5.33: Snapshots of the solution involving elastic aortic walls and floating membrane, closed-loop configuration.



(a) *Entry tear vortexes.*



(b) *Exit tear vortexes.*

Figure 5.34: Velocity profile extracted at two different x positions.

Conclusions

To conclude this Master's thesis work, we try now to resume all main investigated concepts and most relevant results carried out throughout the whole previous discussion. To begin with, surely, we recall what was meant to be the primary objective of this entire work: the realization of a 2D simulation of a type B aortic dissection occurring within the descending aortic channel of a human's cardiovascular system, and consisting in the formation of an entry tear on the inner aortic layer, the *tunica intima*, thus allowing for the blood to flow out from the main vessel giving rise to a second *false* one, therefore called the *false lumen*, extending right beside the original *true lumen*, as long as a second exit tear appears downstream linking back together them two.

Clearly, such a simulation, due to its 2D geometrical configuration, is far from being a realistic case of study of an actual aortic dissection, which is undoubtedly a non-symmetric and three-dimensional phenomenon. Rather than representing a clinical monitoring or predictive tool, thus, the results shown in this thesis work highlight the powerful method devised and adopted in order to perform an effective and efficient numerical simulation of a fluid structure interaction problem. These sorts of problems, in fact, are really challenging from a numerical and mathematical point of view, since they are said to be non-conservative problems, meaning that the action exerted by the fluid on the containing structure, i.e. the aortic vessel, leads to the displacement and deformation of the latter, which in turn modifies the fluid field solution, and thus its mechanical action all over again. Additionally, it is well known the fact that computing domain deformation represents always a tough task to perform, being associated with the need for the computational grid to be either continuously regenerated or moved (e.g. following an advecting approach), as long as a lagrangian description is adopted.

However, by changing the point of view from which we look at the problem, thus meaning by focusing on a newly devised eulerian description, all these challenging difficulties can be overcome easily, since a fixed eulerian grid is employed, with no need to perform its deformation in time, and all moving immersed bodies and interfaces are traced by exploiting an auxiliary function advected by the current itself in an eulerian manner. Furthermore, even boundary conditions are managed the same fashion, that is by enforcing a specific velocity law on moving/fixed objects immersed in the fluid domain, and as far as the interaction between blood and deforming structures is concerned, when this is the case evidently, the latter is performed by considering the entire domain as constituted by a unique monolithic material, characterized by different porosity depending on the considered region within itself, and thanks to which elastic properties take relevance when immersed bodies are encountered, i.e. when a low porosity region is crossed, whereas the

traditional viscous fluid equations are considered when regions with high porosity are the case.

With regards to the adopted numerical method universally associated with numerical simulations of incompressible flows, then, namely the fractional step projection method, one has to admit the fact that in the literature the issue concerning boundary conditions conception and imposition is oftentimes a problem tackled not exhaustively enough, leaving the reader without all needed information and not capable of reproducing and well understanding the mathematical and physical meaning and behaviour of all involved quantities. This is particularly the case for the pressure Poisson's problem associated with the projection step of the method, both in its corrective formulation and not, which is represented by a second order elliptic equation in space, thus needing additional conditions for the pressure variable on the boundaries, or, equivalently, for the corrective projector commonly named as the corrective pressure field. What we have done throughout our work, which is probably one of its most important contributions given to the numerical fluid mechanics community, is to provide a detailed and exhaustive discussion over boundary conditions imposition and problem posedness, leading to a solid and theoretically based approach for their conception, both in the case of stationary and time-dependent flow problems, always supported by numerical results, examples and test cases. Therefore, the main take-away of this thesis work is, undoubtedly, this deep investigation conducted over boundary conditions definition for incompressible flows, capable of ensuring the well-posedness of the problem.

As main developments which are possible to be made concerning this topic, we primarily address the improvement of model reliability and faithfulness to reality, by setting up a more geometrically specific domain provided with suitable boundary conditions capable of ensuring the well-posedness of the problem, as well as the fidelity of its fluid mechanical behaviour. Concerning boundary conditions, then, one additional effort that can be done is to carry out a parametric study aimed at investigating the influence of the capacitor employed in the closed-loop configuration on the 2D computational domain, and in turn on the floating membrane behaviour in time, in order to reach again more realistic dissection mechanics which will then be able to be considered as first simple examples for future clinical exploitations.

Appendix A

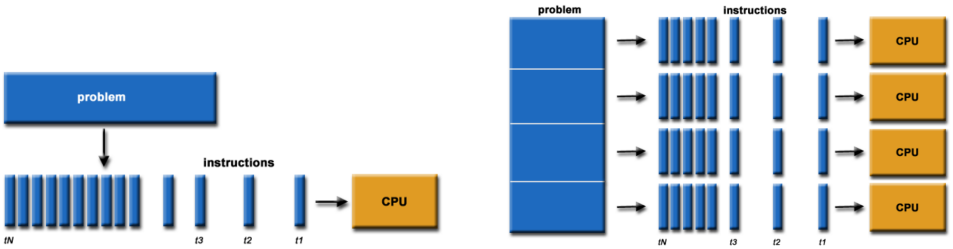
Parallel Computing

Within this appendix we aim to present the fundamentals of parallel computing, since it represents an extremely powerful tools oftentimes employed in several research fields, where large amounts of data and huge computational efforts are involved, required or need to be treated. In fact, this is exactly the case concerning this master's thesis work, whose simulations were all performed exploiting the computational resources of MCIA's cluster, *Curta*, and thus employing those concepts related to the wide parallel computing world. This is why, then, such a section has been dedicated to this topic, due to its importance to the purposes of the present work. All the following information can be found pretty easily on the web, or consulting whatever manual regarding an introduction to parallel computing. Here, we cite in particular [35, 57, 11, 30, 27], just to give some examples. The section is then organized as follows: first, a brief introduction to parallel computing is provided, making comparisons with the traditional sequential computing machines, much more commonly spread so far; subsequently, some formalisms concerning machines' performance assessment are presented, followed by an overview of the most popular architectures and computing/programming models; Before concluding, a deeper investigation on MPI standard interface is proposed, and then, lastly, a list of the most common strategies of design is addressed, to summarize.

A.1 Introduction to Parallel Computing

Before introducing the actual definition of parallel computing, it is worth recalling that, originally, all softwares were thought and written in order to be launched in a serial fashion, accordingly to a sequential computing approach. This means, in other words, that they had to be launched on a single computer, and then executed by a single CPU as well (Central Processing Unit). To this aim, the programme (and so the problem itself) was split into several subsequential instructions, thus executed one after each other, and compulsorily one by one, following a precise order, and leading then to a unique final result (figure A.1). Such machines were initially all based on a so-called Von-Neumann architecture, according to which a unique CPU had to perform reading and writing sequential operations on a connected memory, where the programme itself, together with its data and the outcoming results were all saved and stored.

Accordingly to its simplest definition, parallel computing basically consists of the simultaneous employment of several computational resources, that is, of multiple CPUs, which could be included within a single multi-processor computer, associated with a given number of machines belonging to a common network, or a combination of them two, with the primary goal of solving some huge, complex problem. Such problems, actually, being too computationally expensive to be faced by traditional sequential machines, are more and more common within all scientific fields, involving large amounts of data, complex algorithms, need for wide memory spaces, etc. Similarly to what has been stated concerning sequential computing, the basic idea of parallel computing is that of splitting the main problem into a number of sub-problems, each of which is then assigned to a different CPU, responsible for its proper treatment, fundamentally carried out through a traditional sequential fashion (figure A.1). Formally, considering a problem p_D of



(a) *Sequential computing.* (b) *Parallel computing.*

Figure A.1: Comparison between sequential and parallel computing basic ideas.

dimension (size) n, defined within a domain D , then, if p_D can be said as *parallelizable*, D can be properly decomposed as:

$$D = \sum_{i=1}^{N} d_i = d_1 + d_2 + \cdots + d_N \tag{A.1}$$

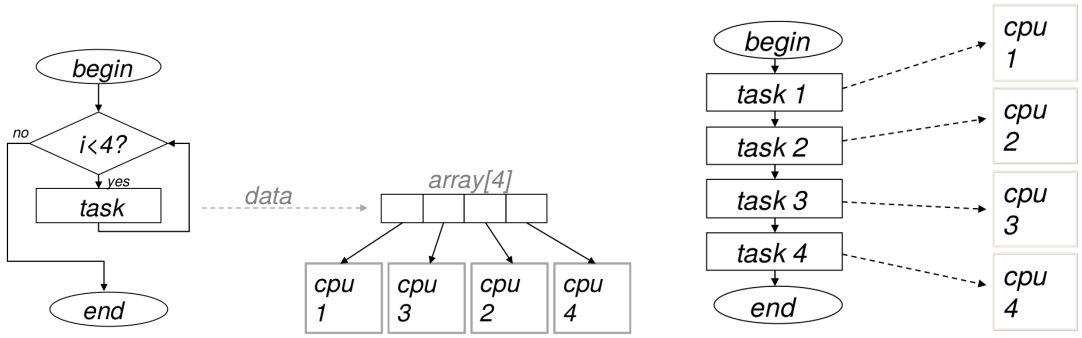
Evidently, not all problems are parallelizable, or at least not completely parallelizable, due to many different reasons, among which, for instance, data dependencies in non-linear problems (Newton-Rhapson methods, etc.). According to this definition, in order to decompose the problem P_D in N sub-problems, one can substantially distinguish between two fundamental parallelization strategies (see figure A.2 for an illustration):

- p_D being *data-parallel*, if D can be decomposed into N different *data-elements*, and the solution of which implies the application of a given function f to each of them, that is:

$$f(D) = \sum_{i=1}^{N} f(d_i) = f(d_1) + f(d_2) + \cdots + f(d_N) \tag{A.2}$$

- p_D being *task-parallel*, if D can be decomposed into N different functions d_i (called *task-elements*, equivalently), and whose solution requires the application of each function element to a common stream of data S, so that:

$$D(S) = \sum_{i=1}^{N} d_i(S) = d_1(S) + d_2(S) + \cdots + d_N(S) \tag{A.3}$$



(a) *Data-parallelism.* (b) *Task-parallelism.*

Figure A.2: Parallelization strategies.

At this point, to make the following discussion clearer, a list of keywords is provided here below, together with a brief description of such adopted terminology, all strictly linked to the parallel computing world.

Task: a sequence of logical instructions, a sort of programme, or a sub-programme, devoted to reach a desired result, and executed by a given CPU. It literally consists of an assignment, and the main programme will therefore be divided into a certain number of tasks.

Serial Execution: it is a matter of executing a given programme in a sequential manner, one instruction by one. Usually, not the whole algorithm can be parallelized, apart from some rare cases, and there will always be, thus, a portion of it that will be executed in such a way. Additionally, each CPU works independently in a serial way on its data stream.

Parallel Execution: trivially, the programme is simultaneously executed by several CPU, which are assigned a different task each. The latter, actually, could be the same for all processors, or, conversely, a different task could even be assigned to each of them, depending on the selected strategy of parallelization. There exists a substantial difference, in fact, between parallelism and concurrency, which is worth underlining at this point: the former refers to the simultaneous execution of different tasks by different CPUs, whereas the latter rather indicates those condition in which, within a given algorithm in execution, multiple tasks could be considered as in progression simultaneously, which is deeply different. To sum up, parallelism requires concurrency, but the contrary is no longer true. For instance, one could consider the Operative System, which is the concurrent multi-tasking programme by definition (but almost never in the parallel sense).

Shared Memory: from a physical point of view, it refers to a unique central memory architecture to which all processors can access freely, in order to do their job; from a programming standpoint, instead, it consists of a model where all tasks can access the

same memory location as if they were linked to a unique central shared memory, even if this is not the actual physical case.

Distributed Memory: as hardware, it represents an architecture where multiple and different memories are distributed and assigned to all processors, forming a network; as software, it refers to a parallel model according to which all tasks can only access their own local memory location, and need to communicate one each other in order to exchange information, even if the physical configuration could be a shared memory type.

Communications: crucial feature of any parallel machine, responsible for information exchange between processors and processes (tasks), performed differently according to specific adopted protocols.

Synchronization: ensures task coordination, allowing for the parallel machine to achieve its objectives by leading all processes to the common end, step by step. Practically, it often represents the conclusion of communication events, through the definition of appropriate checkpoints that must be reached by all tasks before proceeding, thus making operative time increase.

Speedup: index representing the ratio between the execution time of a given serial algorithm and the associated parallel one.

Overhead: total amount of wasted time due to all those operations necessary for the parallelization to be made effective, therefore including communications, synchronizations, tasks startup and termination, and so on.

Granularity: index representing the ratio between the effective computation time and that associated with communication operations. It could be *coarse*, in case the first was greater than the second one, or *fine* on the contrary.

Scalability: capacity of a given parallel machine to increase its performance proportionally with the increasing number of processors composing it, which is not oftentimes a trivial issue, due to larger overheads and hardware installation difficulties.

Portability: associated with the compatibility of a given parallel algorithm with different programming languages.

A.2 Performance Assessment

It is now possible to quantitatively assess a parallel algorithm's performance, by appositely define a proper metric, built up ad-hoc. To this aim, new variables need to be added to the traditional space and time ones, sufficient to describe sequential algorithms only, such a, to begin with, the already encountered concept of speedup, the parallel algorithm

efficiency, its memory bandwidth and its *flops* (floating point operations per second). Let us consider, then, the same parallelizable problem p_D as before, of size n, with domain D. If we aim to parallelize such problem by means of a number p of processors, we can define the so-called *running-time* of the algorithm as the time $T(n,p)$ employed for its execution, under the above mentioned conditions. First, we define two important metrics based on the definition of running-time just given:

- Work: total amount of time necessary to execute a given algorithm, still associated with the problem of size n, disposing of a unique processor, that is $T(n,1)$;
- Span: total amount of time necessary to execute the same algorithm disposing instead of an infinite number of processors, thus $T(n,\infty)$.

Therefore, two distinct lower bounds for the running-time $T(n,p)$ can be defined starting from these last two metrics, Which are:

- the *work law*: the running-time of a given algorithm, parallelized by means of p processors, cannot be lower than at least $\frac{1}{p}$ of its *work*, thus:

$$T(n,p) \geq \frac{T(n,1)}{p}$$

- the *span law*: simply, the running-time of a given algorithm, parallelized by means of p processors, cannot be lower than the running-time of the same algorithm parallelized with an infinite number of processors, i.e. its *span*, that is:

$$T(n,p) \geq T(n,p)$$

Finally, the speedup of a given algorithm can now be defined, by considering the ratio of running-time associated with the best serial one devoted to the solution of a parallelizable problem p_D, denoted with $T_s(n,1)$ (the best obtainable work), with respect to that related to the actual parallelized algorithm, disposing of p processors, say, $T(n,p)$, then:

$$S_p = \frac{T_s(n,1)}{T(n,p)}$$

With, in general, $T_s(n,1) \leq T(n,1)$. Clearly, resuming the work law previously introduced, one has that $S_p \leq p$, meaning that even upper bound exists for the speedup so defined. Therefore, in S_p goes linearly with the number of processors p, we can speak about *linear speedup*, but if we had that, in the best case, $S_p = p$ exactly, then we could even speak about *perfectly linear speedup*, meaning that $T(n,p) = \frac{T_s(n,1)}{p}$, and it would be the maximum speedup reachable for a certain algorithm associated with a problem p_D of a given fixed size n. Actually, this last condition is practically impossible to achieve, because of overheads and other delaying factors, thus leading almost always to sub-linear relationships. Rarely, one could hear about superlinear speedup, which is commonly considered as an unrealistic situation in which the parallel machine would literally be faster than the sum of its composing parts. These situations are reported in figure A.3.

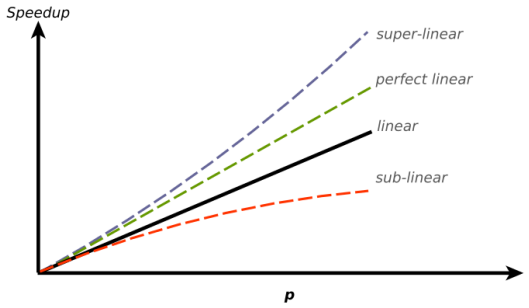


Figure A.3: Parallel machines' speedup typologies.

In case a given algorithm was not completely parallelizable, one can resort to a differently devised expression for the speedup, based on the so-called *Amdahl's law*, which states that:

$$S(p) = \frac{1}{(1-c) + \frac{c}{p}}$$

Where c denotes the fraction of parallelized code, whereas $1 - c = s$ would therefore be the remaining sequential one. Furthermore, one can notice that if the number of employed processors increases towards (ideally) infinity:

$$S(p \to \infty) = \frac{1}{1-c}$$

In this first case of scalability, the size of the problem has not been changed yet, and we therefore speak about *strong scalability*, according to which only the number of processors is increased in order to increase the algorithm's speedup. As an example, consider a machine composed of a high number of processors ($p \to \infty$), exploited to solve a problem whose parallelizable portion is 80%, consequently one has that the maximum theoretical reachable speedup is of 5x only. See figure A.4 as an illustration of such scalability relation. Proceeding with the issue of scalability of a problem, the case in which also the size of

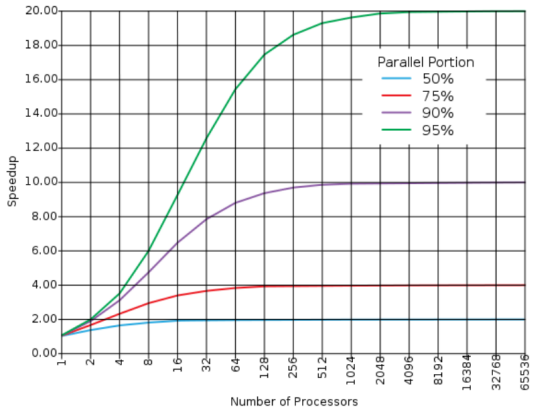


Figure A.4: Example of Amdahl's law application to different parallelizable problems.

p_D can be changed, say, n, needs to be considered. We would deal, then, with a *weak scalability* case, but some additional constraint has to be defined: either the work per processor or the problem size fraction per processor has to be kept constant. Let us consider the first case, where the work per processor of the parallelized problem is kept unchanged by increasing both p and n, in a weak scalability context, then the so-called *Gustavson's law* could turn out to be useful to our purpose. According to such relationship, by considering the running-time of an algorithm executed with only one processor, and composed of a sequential portion s and a parallel one c, equal for all p processors (thus cp will now be the total serial running-time when only one processor is working), we have that the former can be written as $T(n,1) = s + cp$. Therefore, as stated by Gustavson's law, the speedup can newly be expressed as:

$$S_p(p) = \frac{T(n,1)}{T(n,p)} = \frac{s + cp}{s + c}$$

Holding for a given n. Since in many applications it is evident that increasing the size of the problem could lead the component fraction c to become much more important than the sequential one s, this last adopted approach of choosing higher problem sizes n could therefore turn out to be advantageous, see figure A.5 as an illustration. Eventually, one

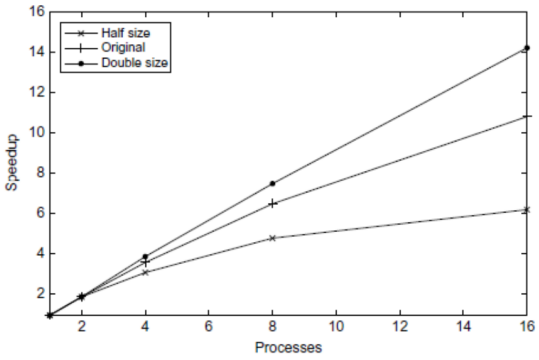


Figure A.5: Example of weak scaling for different sized parallelizable problems.

could also define a sort of *efficiency* of a given algorithm, by taking the ratio between the speedup associated with the n-sized problem p_D and the number of employed processors p, thus:

$$E_p = \frac{S_p}{p} = \frac{T_s(n,1)}{pT(n,p)} \leq 1$$

Where the inequality holds because of the work law previously introduced. This index can tell how much the computational resources are well exploited, and it decreases with p due to greater overheads and time delays, whereas it improves with the problem size n, as shown in figure A.6.

There exist other different parameters that can be introduced with the aim of evaluating the performance of a certain parallel algorithm, such as the already cited flops and bandwidth, which are more experimental assessments linked to the actual implemented programme. The former, i.e. the number of floating-point operations performed each

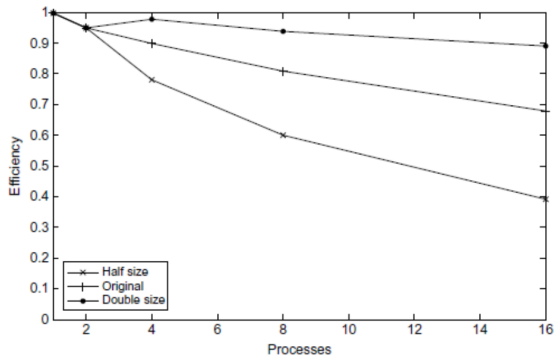


Figure A.6: Example of efficiency scaling for different sized parallelizable problems.

second by the so built machine, is defined as the ratio between the actual flops performed by the machine under evaluation, composed of its actual hardware and software, and those associated with the best machine currently known (see figure A.7 to have an idea of the most spread operative machines in the world), thus:

$$f_{\#} = \frac{F_{actual}}{F_{ref}}$$

Then, as it regards data bandwidth instead, it represents the actual data flux registered

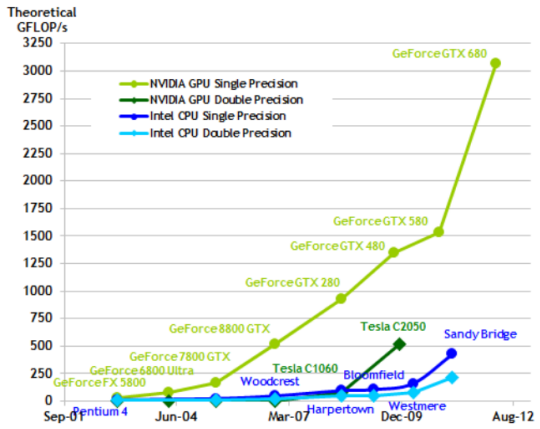


Figure A.7: Example of efficiency scaling for different sized parallelizable problems.

between all processors and the central memory, or vice versa, it is commonly measured in Gb/s, and again expressed through an index defined as the ratio between the actual bandwidth of the considered machine and that of the best known one, that is:

$$B_{Gb/s} = \frac{B_{actual}}{B_{ref}}$$

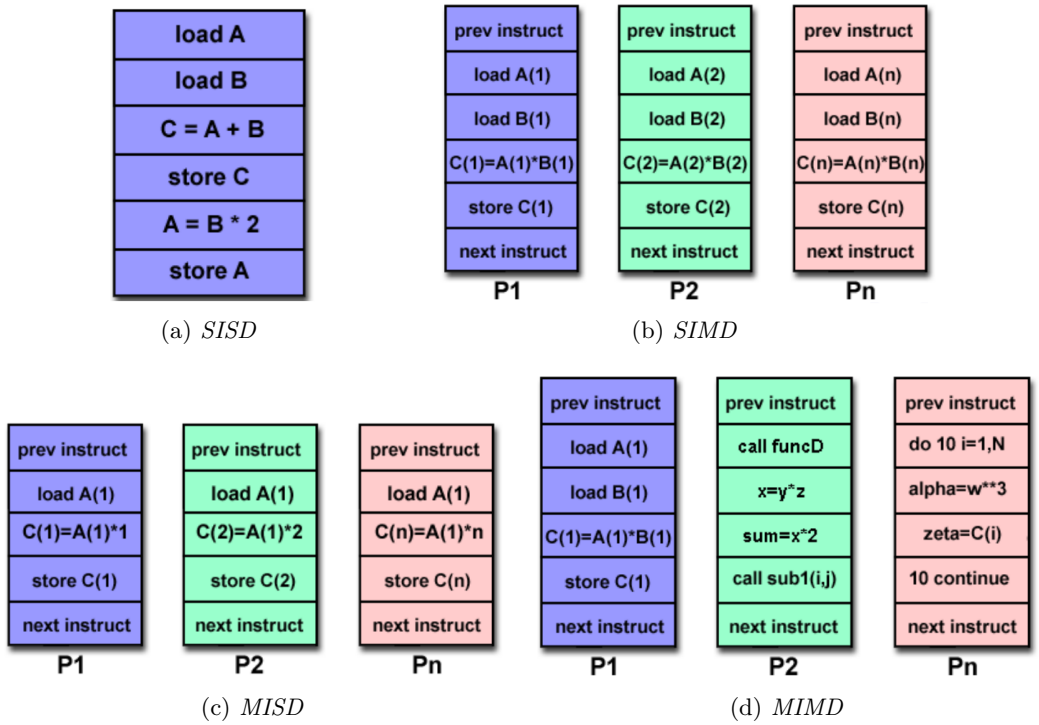
Finally, another factor which is sometimes taken into consideration is the *performance per Watt*, with the aim of maximizing the speedup with the least heat production, in order to reduce environmental impacts.

A.3 Common Architectures

The architecture associated with a parallel machine is at least as important as the software component is, as well. It is responsible for how processors are linked one another and, therefore, the way they are supposed to work together, other than defining the memory organization within the system itself, in addition. The architecture of a parallel machine has to be designed in such a way to allow for the best performance and for the highest computation capacity, considering that serial architectures have already reached their optimum, due to overheating constraints on their maximum clock frequency. Generally, one could resort to the so-called *Flynn's taxonomy* in order to classify architectures (both parallel and serial), with respect to the data and instructions fluxes management, according to which, basically, we could deal with four different kinds of hardware configurations (see figure A.8 as an illustration of what follows):

- SISD (Single Instruction Single Data): traditional serial machines (Von Neumann) with no parallelism, which is currently the most common architecture among users nowadays;
- SIMD (Single Instruction Multiple Data): example of data parallelism, allowing for the execution of a single instruction, but which could be applied to several streams of data, a very common and useful solution when scientific computation is the case, for instance;
- MISD (Multiple Instruction Single Data): not very common architectures, they allow for the application of different tasks to a unique stream of data, consisting thus in an example of task parallelism;
- MIMD (Multiple Instruction Multiple Data): the most spread and versatile parallel architecture, since it is able to deal with different tasks, assigned to the available processors, and which can even be applied to different streams of data, offering the most desired flexibility in that sense. Obviously, it represents also the most complex configuration among those presented here. Furthermore, a MIMD machine, from the programming point of view, could in turn be divided into two more sub-groups: SPMD (single programme multiple data) and MPMD (multiple programme multiple data). The former, differently from the above mentioned SIMD, denotes a software capable of executing a unique programme on all processors, but, as a matter of fact, they could work simultaneously on different portion of it, thus resulting in different tasks, whereas the latter defines different programmes for all involved processors.

Actually, the most important factor of a parallel machine's architecture is its memory organization, and in all those procedures according to which processors can access it. Basically, we could deal with two fundamental types of memory configurations: shared or distributed architectures. As already anticipated in the introduction to this appendix, shared memory structures allow all processors to access a unique, central, common memory location, globally visible, where all modifications made on stored variables affect all linked CPUs, and through which the latter can communicate one another. According to the access protocol, then, those types of architectures can in turn be distinguished in:



(a) *SISD* (b) *SIMD*

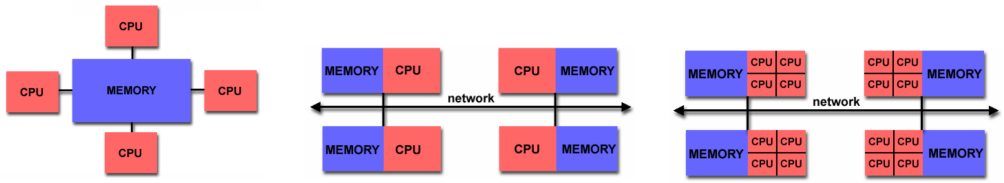
(c) *MISD* (d) *MIMD*

Figure A.8: Flynn's taxonomy illustration of machine's architectures.

- UMA (Uniform Memory Access): all processors are given the same memory access capacity, as well as the same access time (also known as SMP, Symmetric Multi-Processor machines);
- NUMA (Non-Uniform Memory Access): usually composed of several SMP machines, based on different access time and procedures, due to the relative physical positioning of the processors with respect to the needed memory location;

As an advantage, shared memory architectures are surely simpler than distributed memory ones, as well as their enhanced memory access velocity, even though, on the other hand, they present scarce scalability, disposing of an unique, and oftentimes too busy, central address, leading to high overheads. Within distributed memory architectures, instead, each processor works together with its own memory location, independently from all the others, but with which it is connected throughout a network whose topology is specifically defined depending on the machine we want to build up (bus, star, ring, grid, etc.).In order to cooperate one each other, then, proper protocols need to be established, so that processor communications, as well as data exchanges, could be performed. Also suitable synchronization approaches need to be devised, thus. Among their most valuable advantages, one can appreciate their great scalability properties, associated with reduced overheads, to begin with. Clearly, these machines will always be classified as NUMA, following the definition given above. Beside the couple of introduced architectures for parallel machines, then, one can even find some hybrid solutions, mainly focused on

the combination of the advantages shown by both configurations. In figure A.9 the last described categories schemes are reported.



(a) *Shared memory configuration.* (b) *Distributed memory configuration.* (c) *Hybrid memory configuration.*

Figure A.9: Flynn's taxonomy illustration of machine's architectures.

A.4 Computing and Programming Models

Parallel Computing Models

Having introduced the most common machine architectures within the previous section of this appendix, we now aim to analyse more deeply how such systems work, in the presence of a given algorithm, by describing what in the literature are known as *parallel computing models*. It is substantially a matter of formally provide, following a mathematical approach, those tools necessary to assess once again the skills of our parallel machine, by defining apposite models whose describing parameters are able to continuously provide information about its functionality. The following one is meant to be only a brief introduction to this subject, and more interested readers could refer to those works cited at the beginning of this appendix, in particular to [57]. We therefore list here below the most commonly adopted parallel computing models:

- PRAM (Parallel Random Access Machine): model based on the concept of computers' RAM (Rabdom Access Machine), and constituted by the ensemble of all processors which can communicate with an unlimited central memory, properly synchronized. More specifically, depending on the adopted communication protocols, one could deal with PRAM models of type EREW (Exclusive Read, Exclusive Write), CREW (Concurrent Read, Exclusive Write), ERCW (Exclusive Read, Concurrent Write) or CRCW (Concurrent Read, Concurrent Write), depending on whether processors can read/write information from/on an exclusive memory segment or from/on a shared one, respectively;

- PMH (PParallel Memory Hierarchy): it consists of a more realistic evolution of PRAM models, since it also accounts for the different hierarchical levels into which the parallel machine is organized, organizing them on a memory modules hierarchical tree. The tree's leaves correspond to each single processor, and proceeding towards the central trunk one encounters more and more extended memory segments and slowly accessible;

- BSP (Bulk Synchronous Parallel): computing model based on communications, and consequently on synchronization between processors, naturally organized in a distributed manner, and thus representing a cost for the functionality of the entire machine. In such a model, each algorithm's step is partitioned into three distinct sub-steps, i.e. a first local computation phase associated with each processor, a following spread of data to all processors of the network, and finally the definition of a synchronization barrier up to which all processors have to arrive before proceeding with the next step. The overall cost will therefore be the sum of all sub-step overheads;

- LogP: similar to the previous one, this last computing model is more suitable in those cases where synchronization barriers are defined between each pair of processors, within a communication event, rather than being global. In addition, communication sub-steps are enriched considering even latencies, the time needed in order for a message to be transmitted, gaps, the elapsed time between two consecutive messages sent by a processor, and both sending and receiving overheads.

The above cited models are very useful when one has to assess a parallel algorithm performance, which is already implemented on a certain architecture, as a function of the problem size n and the number of employed processors p, but the actual working logic of each algorithm, more specifically, is the primary objective of *parallel programming models*.

Parallel Programming Models

Within this paragraph we aim to briefly introduce those considered the most spread programming models for parallel systems currently implemented by the parallel computing community. Subsequently, we will mainly focus on the one of most interest for our purposes, that is the message-passing model, since it represents the base model for the standard MPI (Message Passing Interface) libraries. Programming models are designed to describe how processors communicate one each other, as well as how they are substantially programmed, thus representing the actual parallel algorithm implementation on the machine. At this point, it is worth underlining that one has not to confuse these models with the machine architecture previously introduced, since they have almost nothing to do one each other, meaning that a shared memory-type architecture could be able to work in combination with the already mentioned message-passing programming logic, which could be intuitively associated with a distributed memory configuration. Indeed, it is the communication protocol to establish how all processors will work together, communicating and sending information one another in order to reach the aim of the whole system, whatever is the actual physical configuration of the machine. Here below we provide a list of the most popular programming models currently adopted:

- Shared Memory: all tasks work as if they could communicate within a common shared memory, whose access priority is managed through the proper definition of lockers and semaphores. From the programming point of view, one has not to deal with the direct communication between each pair of CPUs, since all writing and reading operations on the common memory will be immediately available for

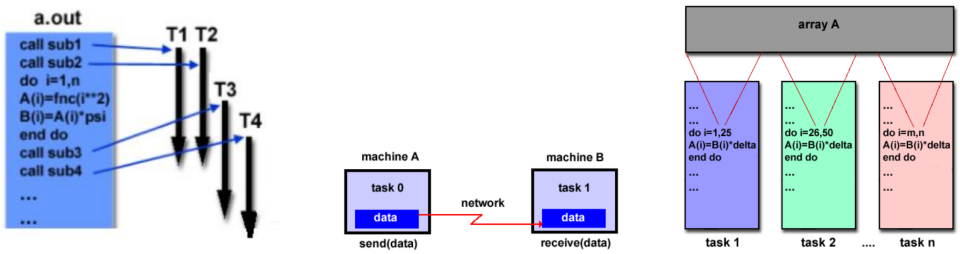
all linked processors. Clearly, such models are thought to work very efficiently in combination with shared memory architectures, and can be very well described by PRAM computing models, but this is not compulsory, as stated above;

- Threads/Tasks: it basically consists of a particular case of shared memory models, according to which the main programme is launched, it executes part of its work in a sequential manner and, when necessary, it will create specific tasks (or threads) that will be run in parallel to the main one, analogously to a subroutine. Each thread works with local data, but it can access the common memory associated with the main programme, through suitable synchronization protocols. A very popular standard of such model is the well-known *OpenMP*, whose Fortran version was released in 1997, whereas the C/C++ one in 1998, generally consisting of a multi-platform, well portable, simple and effective solution for parallel computing;

- Message-Passing: naturally more suitable to be combined with distributed memory architectures, and to be described through a BSP or LogP computing model, this programming strategy is based on the distribution of tasks among all processors linked within a given network, which will then work on a local memory (ideally, although a shared memory architecture is employed), and which are subsequently supposed to exchange data and information by sending and receiving messages according to specific communication protocols. Rules characterising communication events are generally strict and specific, to cite one, the needing for a clear matching between sender and receiver, which must always be established. Among most popular message-passing releases, available since 1980, one can refer to the already cited MPI community, first opened in 1992, and which has always represented the most widely accepted message-passing standard up to the present days;

- Data Parallel: it consists of a suitable model for both shared and distributed memory architectures, based on the simple concept of data parallelism. Tasks assigned to all different processors are basically the same, and they are supposed to work on different streams of data, oftentimes on a portion of a common decomposed domain, for instance. To cite some important implementations of such models, one can refer to Fortran 90 and 95, as well as to the HPF (High Performance Fortran).

In figure A.10 some illustrations are reported concerning these common parallel programming models. Additionally, beside those here presented, there even exist hybrid models, oftentimes resulting from the combination of message-passing with threads models or shared memory ones, or of data parallel and message passing. Lastly, the issue concerning automatic versus manual parallelization is worth being mentioned. This solution could be very useful in order to reduce programmers' total amount of work, since it can be performed through proper algorithms capable of automatically parallelizing a source code. On the other hand, though, this approach can be adopted only for those cases where a very simple and straightforward parallelization is required, for example when cycle operations or image processing are the case, thus representing an extremely limited solution, as a matter of fact.



(a) *Threads model.* (b) *Message-passing model.* (c) *Data parallel model.*

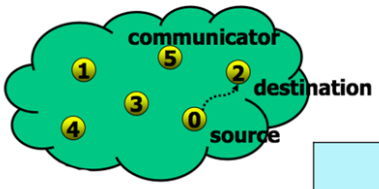
Figure A.10: Some illustrated examples of parallel programming models.

A.5 Standard Interface MPI

The present section has been introduced with the aim of providing a more focused investigation around the already mentioned MPI interface, the most commonly adopted standard in order to implement the previously discussed message-passing programming model. In particular, through the MPI interface, it is possible to create and manage the group of all involved processes, to allow for their message exchanges, to organize work among all different tasks, and so on. To begin with, all programmes that are meant to be parallelized must include the MPI library (module or header file) within their source file, that is *mpi.h* in C, or *mpif.h* in Fortran. Then , the whole body of a parallelized code must be included within the two functions *MPI_Init* and *MPI_Finalize*. This way, the MPI default communicator is finally initialized, commonly named *MPI_COMM_WORLD*, which collects all processes involved in the parallelization and assigns them a unique identification number, named *rank*. The total number of processes, or processors, included within a communicator represents the *size* of the latter.

The most elementary communication pattern between two processors is called *point-to-point* communication (or even P2P), consisting of a *sender* process, in charge of envoying a certain message, and a *receiver* process, meant to collect it. Clearly, messages are standardized as well, since they are generally composed of an *envelope*, made up of information concerning source, destination, communicator and tag of the message, and of a *body*, in turn composed of buffer, datatype and count. See figure A.11 to have a representation of a point-to-point communication within a global communicator, and to have an idea of a standard message configuration. In order to correctly send/receive a message, then, both processes must call the relative function *MPI_Send* or *MPI_Recv*, respectively, specifying their correspondent rank. A common issue concerning P2P communications is that of *deadlocks*, a particular condition in which the two involved processes remain blocked while waiting respectively the other one to respond. Usually, a solution to this problem could be represented by the function *MPI_Sendrecv*. Furthermore, P2P communications can also be distinguished as *blocking* or *non-blocking*, whether they imply the proceeding of the job assigned to the sender process, after its message sending, even if the communication event has not been completed yet, or not, meaning that both processes cannot proceed until the communication event has been correctly accomplished. With "correctly accomplished" one means that, in case of blocking communications, the receiver process must confirm to have successfully collected the message sent by its associated sender, whereas, conversely,

in the case of non-blocking communications, the messaging completeness check will be effectuated only later, therefore requiring more programming complexity, but allowing for a better reduction o latencies and deadlocks. Obviously, until the communication event is complete in case of non-blocking communication, that memory area is no longer accessible by the sender process, and the message contents are not exploitable by the receiver process.



Envelope				Body		
source	destination	communicator	tag	buffer	count	datatype

Figure A.11: Point-to-point communication and standard message format.

A deeply different group of communication patterns, then, is represented by the so-called *collective communications.* Beside P2P communications, the latter allows all processes within a communicator (e.g. the MPI_COMM_WORLD) to join the communication event, according to several different schemes. In the following, we list some of the most employed collective communication functions, which could even be grouped into three main categories: *all-to-one* communications, *one-to-all* communications and *all-to-all* communications, which are pretty self-explaining. Thus, there follow some of those most popular collective communication functions (some corresponding illustrations are shown in figure A.12):

- Reduce: belonging to the all-to-one class, it collects the contribution coming from each process, it reduces them into a single resulting value through some mathematical operator (maximum, minimum, sum, product, etc.) and return it to the calling process, named *root*;

- Broadcast: belonging to the one-to-all class, it copies data coming from the root process (sender) onto all processes present within the communicator, included itself;

- Scatter: one-to-all class, it splits a datum into several, distinct and equal portions, and then it sends one of them to each process within the communicator, included the sender itself, following a rank-order;

- Gather: all-to-one class, it makes each process envoy the content of its send-buffer to the root process, contrarily to what was done by the scatter function;

- Allgather: it consists of the ordinate combination of gather and broadcast functions, since each process collects the content of the send-buffer of all the others within the communicator, thus belonging to the all-yo-all class;

- Allreduce: equivalent to the execution of a reduce function, followed by a broadcast function, thus again all-to-all, and allowing to compute a reduced result which will then be copied on each process within the communicator;

- Barrier: it creates a barrier for all communicator processes, which are then paused until it is reached by all of them;

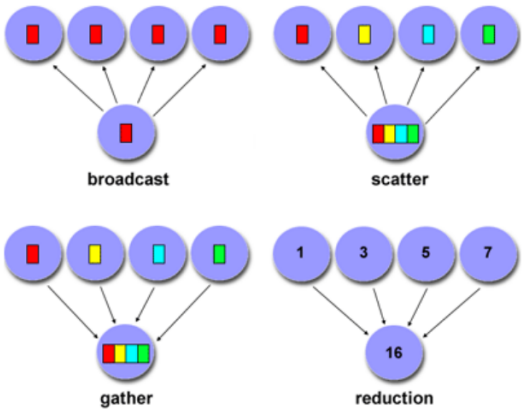


Figure A.12: Examples of collective communication patterns.

To conclude, with MPI it is even possible to define additional communicators, creating then sub-groups of processes organized according to a given topology, and capable of communicating with processes belonging to different communicators, beside the MPI_COMM_WORLD main one.

Once the programme is properly parallelized according to MPI standardization, it can be compiled and launched through *mpirun* or *mpiexec* by specifying also the number of processes into which we intend to divide it.

A.6 Common Strategies of Design

There is no general rule to follow when designing a parallel programme, although it is possible to summarize what has been stated within this appendix in order to devise a sort of common strategy.

First, one needs to know if the programme we have to deal with, starting from its serial stage, can be parallelized or not, and in what measure with respect of its parallelizable fraction. In order to obtain a good and efficient parallelization, one needs to identify the *hotspots* of the algorithm, that is its most computationally expensive and hard-working parts (e.g. long cycling operations). Then, a good idea would be that of individuate the *bottlenecks*, which are the slowest portion of the programme, even if parallelized, such as input/output operations, and which have thus to be limited, as well as data dependencies and other parallalization inhibitors. The next step in parallelizing a problem is its decomposition, or partitioning, which could consist of either a domain decomposition,

or a functional decomposition, or even both, following the already mentioned strategies of data and task parallelism.

Subsequently, communication is an issue of the most importance to consider when designing a parallel algorithm. Communications lead to an increasement of overheads, therefore the programmer is responsible for the proper managing of communication events, by avoiding numerous communication acts, grouping them together whenever it is possible, and thus reducing latencies. As a general rule, asynchronous and non-blocking communications should be preferred to those synchronous and blocking ones, even though the latter are simpler and safer in comparison, because of all related features listed in the previous sections. Also data dependencies should be minimized, as they represent the so-called parallelization inhibitors.

In addition, another characteristic that must be taken into consideration, already cited within the introduction to this appendix, is the granularity of the parallelized algorithm, i.e. the ratio between computation and communication events. Having a fine granularity would surely lead to high overheads, due to the excessive number of communication events with respect to the entire algorithm running, but, on the other hand, it could also improve another important feature of parallelized programmes: the *load balancing*. This last approach consists of assigning all processes an equivalent amount of work, so that each task will reach the subsequent synchronization point almost together with the others. Therefore, by disposing of frequent communication events one can equally distribute assignments among all involved tasks in a more straightforward way.

Finally, in order to further reduce parallelization inhibitors, one should minimize also input/output operations (bottlenecks), even though some parallelized implementations exist also for these kind of operations. Acquiring data from one process and then distributing them to all the others is still the most preferred way employed so far.

All what has been discussed within this last section can be in turn summarized accordingly to the so-called four *four-steps strategy* formulated by I. Foster in 1995 (see [57]), whose flowchart scheme is depicted in figure A.13

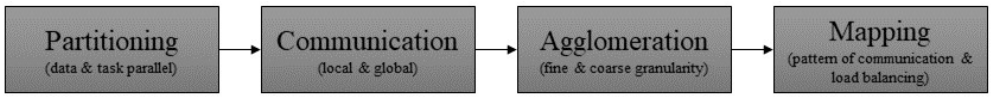


Figure A.13: Scheme of 4-steps Foster's parallelization strategy.

Appendix B

Linear Solvers

B.1 A Brief Survey

We decided to add this appendix in order to provide readers with a very brief introduction on linear solvers, with special care for those employed to the aim of this thesis, that is, Krylov's space linear solvers. As suggested by their name, it is a matter of solving linear systems, typically characterised by large and sparse matrixes, whose solution may be easier when carried out following an iterative procedure. On the other hand, if one had to deal with dense matrixes, direct methods could be more suitable. As a reference for what it follows, one can consider the contents of [42], from which the present introduction is completely extracted. Pursuing our primary goal, it is worth remarking that oftentimes problems governed by systems of PDEs, such as it happens for most physical problems, are associated with large and sparse linear systems when their discretization is introduced with the aim of numerically find an approximate solution to them.

Direct solvers are mainly based on the Gauss elimination algorithm, or some sort of its derivation, usually concerned with the definition of permutation factors, proper decomposition matrixes, depending on the specific faced problem-case, devised in order to ensure the stability of the associated solving algorithm, as well as its occasional parallelization, when required, if huge and massive problems involving a very large amount of data are faced. These methods would be more appropriate when dense matrix systems are the case, conversely, if systems whose coefficient matrixes presented a sparse pattern, iterative solvers should be preferred. Among them, one can encounter the traditional Jacobi or Gauss-Seidel schemes, or the much more employed *preconditioned Krylov space solvers*, to cite some. The latter, thus, are basically preconditioning methods, also known as *semi-iterative* methods or *polynomial acceleration* ones, aiming at building a given sequence of matrix-vector products, as explained in the following.

B.2 Krylov's Space Methods

In order to reach the normally exploited formulation of Krylov's methods, let us start from the simplest iterative one, that is, the Jacobi's scheme. By considering the following

linear system:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b} \tag{B.1}$$

Where $\boldsymbol{A} \in \mathbb{R}^{n,n}$, whereas $\boldsymbol{x}, \boldsymbol{b} \in \mathbb{R}^n$, and where all products have to be intended in the matrix-vector sense. We can introduce the following splitting of the coefficient matrix $\boldsymbol{A}$, so that:

$$(\boldsymbol{A} - \boldsymbol{D})\boldsymbol{x} + \boldsymbol{D}\boldsymbol{x} = \boldsymbol{b}$$

Thus, it follows that:

$$\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{x} + \boldsymbol{D}^{-1}\boldsymbol{b}$$

And, by defining $\hat{\boldsymbol{B}} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{A}$ and $\hat{\boldsymbol{b}} = \boldsymbol{D}^{-1}\boldsymbol{b}$, one finally obtains the classic (Jacobi) fixed-point problem associated with all iterative methods:

$$\boldsymbol{x} = \hat{\boldsymbol{B}}\boldsymbol{x} + \hat{\boldsymbol{b}} \quad \longrightarrow \quad \boldsymbol{x}^{k+1} = \hat{\boldsymbol{B}}\boldsymbol{x}^k + \hat{\boldsymbol{b}} \tag{B.2}$$

Obviously, in order to ensure convergence, it must hold that:

$$\boldsymbol{x}^k \to \bar{\boldsymbol{x}}, \quad k \to \infty$$

Where with $\bar{\boldsymbol{x}}$ it has been denoted the exact solution to B.1. The latter is guaranteed as long as:

$$\rho(\hat{\boldsymbol{B}}) < 1$$

Where the operator $\rho(.)$ refers to the spectral radius of its argument. Unfortunately, this approach suggests to know a priori the exact solution $\bar{\boldsymbol{x}} = \boldsymbol{A}^{-1}\boldsymbol{b}$ to the problem represented by B.1. Alternatively, one could consider the residual associated with each single iterative step:

$$\boldsymbol{r}^k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k \neq 0 \tag{B.3}$$

Which, as indicated, is different from zero until the exact solution is reached. Furthermore, one can notice that:

$$\boldsymbol{r}^k = \boldsymbol{A}\bar{\boldsymbol{x}} - \boldsymbol{A}\boldsymbol{x}^k = \boldsymbol{A}(\bar{\boldsymbol{x}} - \boldsymbol{x}^k)$$

Where the connection between the current residual and the related absolute error with respect to the exact solution is shown. Then, by assuming $\boldsymbol{D} = \boldsymbol{I}$, which means that $\hat{\boldsymbol{B}} = \boldsymbol{I} - \boldsymbol{A} = \boldsymbol{B}$ and $\hat{\boldsymbol{b}} = \boldsymbol{b}$, one has that:

$$\boldsymbol{r}^k = \boldsymbol{b} - \boldsymbol{x}^k + \boldsymbol{B}\boldsymbol{x}^k = \boldsymbol{x}^{k+1} - \boldsymbol{x}^k$$

Being now simply $\boldsymbol{x}^{k+1} = \boldsymbol{B}\boldsymbol{x}^k - \boldsymbol{b}$. Consequently, by multiplying $\boldsymbol{A}$ to both the right and the left hand side of the previous expression, we have that:

$$\boldsymbol{A}\boldsymbol{x}^{k+1} = \boldsymbol{A}\boldsymbol{x}^k + \boldsymbol{A}\boldsymbol{r}^k$$

Recalling B.3, according to which $\boldsymbol{r}^k = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}^k$, we obtain:

$$\boldsymbol{r}^{k+1} - \boldsymbol{b} = \boldsymbol{r}^k - \boldsymbol{b} - \boldsymbol{A}\boldsymbol{r}^k \quad \longrightarrow \quad \boldsymbol{r}^{k+1} = (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{r}^k = \boldsymbol{B}\boldsymbol{r}^k \tag{B.4}$$

Which is now written in the form of a recursive expression for the residual $\boldsymbol{r}$. The latter leads to the following iterative problem:

$$\boldsymbol{r}^k = p_k(\boldsymbol{A})\boldsymbol{r}^0 \tag{B.5}$$

Where $\boldsymbol{r}^k \in span\{\boldsymbol{r}^0, \boldsymbol{A}\boldsymbol{r}^0, \ldots, \boldsymbol{A}^k\boldsymbol{r}^0\}$, whereas the polynomial of k-th degree is generally defined as $p_k(\xi) = (1-\xi)^k$. Finally, by reminding that $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{r}^k$, we obtain the same recursive problem in terms of the solution vector $\boldsymbol{x}$:

$$\boldsymbol{x}^k = \boldsymbol{x}^0 + \boldsymbol{r}^0 + \cdots + \boldsymbol{r}^{k-1} = \boldsymbol{x}^0 + q_{k-1}(\boldsymbol{A})\boldsymbol{r}^0 \tag{B.6}$$

Actually, the two vectors resulting from B.5 and B.6 lie in the same vectorial space defined above, introducing a shifting of $\boldsymbol{x}^0$, that is $span\{\boldsymbol{r}^0, \boldsymbol{A}\boldsymbol{r}^0, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}^0\}$, which is right what we will call from now on a *Krylov's space* (or subspace), $\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r})$, thus defined as:

$$\mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}) = span\{\boldsymbol{r}, \boldsymbol{A}\boldsymbol{r}, \ldots, \boldsymbol{A}^{k-1}\boldsymbol{r}\} \tag{B.7}$$

Therefore, the main idea is to build up a sequence of approximate solutions $\boldsymbol{x}^k \in \boldsymbol{x}^0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}^0)$ tending to the exact one $\bar{\boldsymbol{x}} \in \boldsymbol{x}^0 + \mathcal{K}_k(\boldsymbol{A}, \boldsymbol{r}^0)$, included in the same Krylov's space. This means also that the sequence of residual vectors $\boldsymbol{r}^k \in \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{r}^0)$ must tend to zero. Practically, Krylov solvers are always very slowly convergent, if at least converging, thus making it necessary to resort to suitable preconditioners, so that problem B.1 can be written in different forms.

B.3 An Application: the Conjugate Gradient Method

The most popular Krylov's space solver method is, probably, the so-called *conjugate-gradient method* (CG), which represents substantially an orthogonal projection method aimed at minimizing the error associated with the research for the solution, with respect to the exact one, according to the so-called energy norm. Being such error of the form $\boldsymbol{e}^k = \boldsymbol{x}^k - \bar{\boldsymbol{x}}$, one aims at studying the minimization problem associated with the functional $\Psi(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^t\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}^t\boldsymbol{x} + c$, whose gradient can be expressed as $\nabla\Psi = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = -\boldsymbol{r}$, and where then, obviously, $\nabla\Psi(\bar{\boldsymbol{x}}) = 0$, being thus $\bar{\boldsymbol{x}}$ its minimizer. In fact, it can be proved that minimizing the functional Ψ, which corresponds to find the exact solution to the linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, say, $\bar{\boldsymbol{x}}$, is equivalent to minimize the energy norm of the error $\boldsymbol{e} = \boldsymbol{x} - \bar{\boldsymbol{x}}$, as stated above.

The minimum of the functional can be found by following the steepest descent way, which means throughout a curved line. From a numerical standpoint, one can perform only discrete steps (iterations), whose direction could be chosen as coincident with the gradient of the functional itself, and denoted with $\boldsymbol{v}_0$. By chosing, in addition, a second direction $\boldsymbol{v}_1$ which is orthogonal to $\boldsymbol{A}\boldsymbol{v}_0$ (or $\boldsymbol{A}$-orthogonal), one can reach the minimum in at most two steps. In multi-dimensions:

$$\boldsymbol{v}_k^t\boldsymbol{A}\boldsymbol{v}_q = 0, \quad q = 0, \ldots, k-1$$

Thus, we have that:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{v}_k\omega_k \tag{B.8}$$

From which it follows that, as a consequence:

$$\boldsymbol{r}^{k+1} = \boldsymbol{r}^k - \boldsymbol{A}\boldsymbol{v}_k\omega_k \tag{B.9}$$

Where the weight ω_k is chosen so that the A-norm (energy norm) of the error $\boldsymbol{e}$ is once again minimized along the direction $\boldsymbol{x}^k + \boldsymbol{v}_k\omega$, for some ω_k minimizer. In order for this method to be a Krylov's space one, there must hold that:

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^0 + \boldsymbol{v}_0\omega_0 + \cdots + \boldsymbol{v}_0\omega_0 \in \boldsymbol{x}^0 + span\{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n\}$$

Where, indeed, $span\{\boldsymbol{v}_0, \ldots, \boldsymbol{v}_n\} = \mathcal{K}_{k+1}(\boldsymbol{A}, \boldsymbol{r}^0)$. Notice that this method is applicable only in case of symmetric and positive definite matrixes $\boldsymbol{A}$. Nevertheless, if that is not the case, one could resort to the more complicated *generalized CG method* or the *biconjugate gradient method*, among the others.

Bibliography

[1] https://ctsurgerypatients.org/adult-heart-disease/aortic-dissection.

[2] https://en.wikipedia.org/wiki/Hyperelastic-material.

[3] https://en.wikipedia.org/wiki/Finite-strain-theory.

[4] https://en.wikipedia.org/wiki/Strain-energy-density-function.

[5] https://en.wikipedia.org/wiki/Mooney-Rivlin-solid.

[6] https://www.geeksforgeeks.org/runge-kutta-4th-order-method.

[7] https://en.wikipedia.org/wiki/Projection-method-fluid-dynamics.

[8] D. Adalsteinsson and J. A. Sethian, *A fast level set method for propagating interfaces*, J. Comput. Phys, 118 (1994).

[9] P. Angot, C.-H. Bruneau, and P. Fabrie, *A penalization method to take into account obstacles in incompressible viscous flows*, Numerische Mathematik, 81 (1999), pp. 497–520.

[10] G. B. Arfken and H. J. Weber, *Mathematical methods for physicists*, 1999.

[11] B. Barney et al., *Introduction to parallel computing*, Lawrence Livermore National Laboratory, 6 (2010), p. 10.

[12] A. Benim, F. Gul, A. Nahavandi, A. Assmann, P. Feindt, and F. Joos, *Cfd analysis of blood flow in human aorta with experimental validation*, in Proceedings of the 7th International Conference on Computational Heat and Mass Transfer, Istanbul, Turkey, 2011.

[13] A. Benim, A. Nahavandi, A. Assmann, D. Schubert, P. Feindt, and S. Suh, *Simulation of blood flow in human aorta with emphasis on outlet boundary conditions*, Applied Mathematical Modelling, 35 (2011), pp. 3175–3188.

[14] M. Bergmann, J. Hovnanian, and A. Iollo, *An accurate cartesian method for incompressible flows with moving boundaries*, Communications in Computational Physics, 15 (2014), pp. 1266–1290.

[15] M. Bergmann and A. Iollo, *Modeling and simulation of fish-like swimming*, Journal of Computational Physics, 230 (2011), pp. 329–348.

[16] C. Bertoglio, A. Caiazzo, and M. A. Fernández, *Fractional-step schemes for the coupling of distributed and lumped models in hemodynamics*, SIAM Journal on Scientific Computing, 35 (2013), pp. B551–B575.

[17] M. Bonfanti, S. Balabani, J. P. Greenwood, S. Puppala, S. Homer-Vanniasinkam, and V. Díaz-Zuccarini, *Computational tools for clinical support: a multi-scale compliant model for haemodynamic simulations in an aortic dissection based on multi-modal imaging data*, Journal of The Royal Society Interface, 14 (2017), p. 20170632.

[18] M. Bonfanti, G. Franzetti, G. Maritati, S. Homer-Vanniasinkam, S. Balabani, and V. Díaz-Zuccarini, *Patient-specific haemodynamic simulations of complex aortic dissections informed by commonly available clinical datasets*, Medical Engineering & Physics, (2019).

[19] M. Braack, P. B. Mucha, and W. M. Zajaczkowski, *Directional do-nothing condition for the navier-stokes equations*, J. Comput. Math, 32 (2014), pp. 507–521.

[20] M. Catanho, M. Sinha, and V. Vijayan, *Model of aortic blood flow using the windkessel effect*, University of California of San Diago, San Diago, (2012).

[21] X. Chen, *A nonlinear viscoelastic mooney-rivlin thin wall model for unsteady flow in stenosis arteries*, (2003).

[22] Z. Cheng, C. Juli, N. Wood, R. Gibbs, and X. Xu, *Predicting flow in aortic dissection: comparison of computational model with pc-mri velocity measurements*, Medical engineering & physics, 36 (2014), pp. 1176–1184.

[23] Z. Cheng, C. Riga, J. Chan, M. Hamady, N. B. Wood, N. J. Cheshire, Y. Xu, and R. G. Gibbs, *Initial findings and potential applicability of computational simulation of the aorta in acute type b dissection*, Journal of vascular surgery, 57 (2013), pp. 35S–43S.

[24] A. J. Chorin, *A numerical method for solving incompressible viscous flow problems*, Journal of computational physics, 2 (1967), pp. 12–26.

[25] A. J. Chorin, *Numerical solution of the navier-stokes equations*, Mathematics of computation, 22 (1968), pp. 745–762.

[26] A. J. Chorin, J. E. Marsden, and J. E. Marsden, *A mathematical introduction to fluid mechanics*, vol. 3, Springer, 1990.

[27] M. Cremonesi, P. Dagna, and P. Ramieri, *Patc course: Introduction to parallel computing with mpi and openmp (2015)*, (2015).

[28] D. D'Ambrosio, *Fluidodinamica computazionale (slides)*, Polito, (2019).

[29] A. De Brauer, A. Iollo, and T. Milcent, *A cartesian scheme for compressible multimaterial models in 3d*, Journal of Computational Physics, 313 (2016), pp. 121–143.

[30] V. Eijkhout, *Introduction to parallel computing*, 2011.

[31] R. P. Fedkiw and X.-D. Liu, *The ghost fluid method for viscous flows*, in Innovative methods for numerical solution of partial differential equations, World Scientific, 2002, pp. 111–143.

[32] G. Fekken et al., *Numerical simulation of free-surface flow with moving rigid bodies*, PhD thesis, Rijksuniversiteit Groningen, 2004.

[33] W. W. Feng and J. O. Hallquist, *On mooney-rivlin constants for elastomers*, stress (force per unit undeformed area), 1 (2017), p. 5.

[34] S. E. Francis, *Continuous estimation of cardiac output and arterial resistance from arterial blood pressure using a third-order windkessel model*, PhD thesis, Massachusetts Institute of Technology, 2007.

[35] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, et al., *Open mpi: Goals, concept, and design of a next generation mpi implementation*, in European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting,

Springer, 2004, pp. 97–104.

[36] P. Gabriel and L. M. Tine, *High-order weno scheme for polymerization-type equations*, in ESAIM: Proceedings, vol. 30, EDP Sciences, 2010, pp. 53–69.

[37] D. Gallo, G. De Santis, F. Negri, D. Tresoldi, R. Ponzini, D. Massai, M. Deriu, P. Segers, B. Verhegghe, G. Rizzo, et al., *On the use of in vivo measured flow rates as boundary conditions for image-based hemodynamic models of the human aorta: implications for indicators of abnormal flow*, Annals of biomedical engineering, 40 (2012), pp. 729–741.

[38] J. Gawinecka, F. Schönrath, and A. von Eckardstein, *Acute aortic dissection: pathogenesis, risk factors and diagnosis*, Swiss medical weekly, 147 (2017).

[39] H. Gharahi, B. A. Zambrano, D. C. Zhu, J. K. DeMarco, and S. Baek, *Computational fluid dynamic simulation of human carotid artery bifurcation based on anatomy and volumetric blood flow rate measured with magnetic resonance imaging*, International journal of advances in engineering sciences and applied mathematics, 8 (2016), pp. 46–60.

[40] F. Gibou, R. Fedkiw, and S. Osher, *A review of level-set methods and some recent applications*, Journal of Computational Physics, 353 (2018), pp. 82–109.

[41] J.-L. Guermond, P. Minev, and J. Shen, *An overview of projection methods for incompressible flows*, Computer methods in applied mechanics and engineering, 195 (2006), pp. 6011–6045.

[42] M. H. Gutknecht, *A brief introduction to krylov space methods for solving linear systems*, in Frontiers of Computational Science, Springer, 2007, pp. 53–62.

[43] F. Ham and Y.-N. Young, *A cartesian adaptive level set method for two-phase flows*, (2003).

[44] G.-S. Jiang and D. Peng, *Weighted eno schemes for hamilton–jacobi equations*, SIAM Journal on Scientific computing, 21 (2000), pp. 2126–2143.

[45] N. Johansson, *Implementation of a standard level set method for incompressible two-phase flow simulations*, 2011.

[46] V. John, *Finite element methods for incompressible flow problems*, Springer, 2016.

[47] I. Kokalari, T. Karaja, and M. Guerrisi, *Review on lumped parameter method for modeling the blood flow in systemic arteries*, (2013).

[48] N. Kumar and V. V. Rao, *Hyperelastic mooney-rivlin model: determination and physical interpretation of material constants*, Parameters, 2 (2016), p. 01.

[49] E. Maitre, *Review of numerical methods for free interfaces*, Les Houches, 27 (2006), p. 31.

[50] G. Marckmann and E. Verron, *Comparison of hyperelastic models for rubber-like materials*, Rubber chemistry and technology, 79 (2006), pp. 835–858.

[51] L. A. Mihai, L. Chin, P. A. Janmey, and A. Goriely, *A comparison of hyperelastic constitutive models applicable to brain and fat tissues*, Journal of The Royal Society Interface, 12 (2015), p. 20150486.

[52] S. Mirjalili, S. S. Jain, and M. Dodd, *Interface-capturing methods for two-phase flows: An overview and recent developments*, Center for Turbulence Research Annual Research Briefs, 2017 (2017), pp. 117–135.

[53] R. Mittal and G. Iaccarino, *Immersed boundary methods*, Annu. Rev. Fluid Mech., 37 (2005), pp. 239–261.

[54] M. E. Moghadam, I. E. Vignon-Clementel, R. Figliola, A. L. Marsden, M. of Congenital Hearts Alliance (MOCHA) Investigators, et al., *A modular numerical method for implicit 0d/3d coupling in cardiovascular finite element simulations*, Journal of Computational Physics, 244 (2013), pp. 63–79.

[55] J. Y. Moon, D. C. Suh, Y. S. Lee, Y. W. Kim, and J. S. Lee, *Considerations of blood properties, outlet boundary conditions and energy loss approaches in computational fluid dynamics modeling*, Neurointervention, 9 (2014), p. 1.

[56] M. Mooney, *A theory of large elastic deformation*, Journal of applied physics, 11 (1940), pp. 582–592.

[57] C. A. Navarro, N. Hitschfeld-Kahler, and L. Mateu, *A survey on parallel computing and its applications in data-parallel problems using gpu architectures*, Communications in Computational Physics, 15 (2014), pp. 285–329.

[58] S. Nobakht, *A computational model of aneurysmal flow at the abdominal aorta*, Master's thesis, Graduate Studies, 2013.

[59] S. Osher and J. A. Sethian, *Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations*, Journal of computational physics, 79 (1988), pp. 12–49.

[60] N. Peña Pérez, *Windkessel modeling of the human arterial system*, B.S. thesis, 2016.

[61] P.-O. Persson, *The level set method*, Lecture notes, MIT, 16 (2005), pp. 1–7.

[62] C. S. Peskin, *Fluid dynamics of the heart and its valves*, in APS Meeting Abstracts, 1997.

[63] S. Pieraccini and S. Berrone, *Metodi numerici e calcolo scientifico (slides)*, Polito, (2018).

[64] S. Pirola, Z. Cheng, O. Jarral, D. O'Regan, J. Pepper, T. Athanasiou, and X. Xu, *On the choice of outlet boundary conditions for patient-specific analysis of aortic flow using computational fluid dynamics*, Journal of biomechanics, 60 (2017), pp. 15–21.

[65] L. Quartapelle, *Numerical solution of the incompressible Navier-Stokes equations*, vol. 113, Birkhäuser, 2013.

[66] A. Quarteroni and S. Quarteroni, *Numerical models for differential problems*, vol. 2, Springer, 2009.

[67] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical mathematics*, vol. 37, Springer Science & Business Media, 2010.

[68] D. Rempfer, *On boundary conditions for incompressible navier-stokes problems*, (2006).

[69] S. Ringgaard, S. Oyre, H. Stødkilde-Jørgensen, and E. Pedersen, *Velocity profile fitting methods in physiological pulsatile flow: Implications for mr phase contrast based on wall shear stress estimates*, in 7th Scientific Meeting, Philadelphia, Pennsylvania, 1999.

[70] R. Rivlin, *Large elastic deformations of isotropic materials. i. fundamental concepts*, Philosophical Transactions of the Royal Society of London. Series A, Mathematical

and Physical Sciences, 240 (1948), pp. 459–490.

[71] S. Scarsoglio, *Fluidodinamica nel volo spaziale - biofluidodinamica spaziale (slides)*, Polito, (2019).

[72] K. Schneider, *Immersed boundary methods for numerical simulation of confined fluid and plasma turbulence in complex geometries: a review*, Journal of Plasma Physics, 81 (2015).

[73] J. Sethian, *Advancing interfaces: level set and fast marching methods*, in Proceedings of the international conference on industrial and applied mathematics: plenary lectures, 1999.

[74] J. A. Sethian, *Theory, algorithms, and applications of level set methods for propagating interfaces*, Acta numerica, 5 (1996), pp. 309–395.

[75] E. K. Shang, D. P. Nathan, R. M. Fairman, J. E. Bavaria, R. C. Gorman, J. H. Gorman III, and B. M. Jackson, *Use of computational fluid dynamics studies in predicting aneurysmal degeneration of acute type b aortic dissections*, Journal of vascular surgery, 62 (2015), pp. 279–284.

[76] Y. Shi, P. Lawford, and R. Hose, *Review of zero-d and 1-d models of blood flow in the cardiovascular system*, Biomedical engineering online, 10 (2011), p. 33.

[77] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in Advanced numerical approximation of nonlinear hyperbolic equations, Springer, 1998, pp. 325–432.

[78] C.-W. Shu, *High order weighted essentially nonoscillatory schemes for convection dominated problems*, SIAM review, 51 (2009), pp. 82–126.

[79] R. Temam, *Remark on the pressure boundary condition for the projection method*, Theoretical and Computational Fluid Dynamics, 3 (1991), pp. 181–184.

[80] F. Van de Vosse and M. Van Dongen, *Cardiovascular fluid mechanics-lecture notes*, Faculty of Applied Physics, Faculty of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, Netherlands, (1998).

[81] T. Van Kruchten, *Cfd modeling of abdominal aortic aneurysms*, (2015).

[82] N. Westerhof, J.-W. Lankhaar, and B. E. Westerhof, *The arterial windkessel*, Medical & biological engineering & computing, 47 (2009), pp. 131–141.

[83] P. Youssefi, A. Gomez, C. Arthurs, R. Sharma, M. Jahangiri, and C. Alberto Figueroa, *Impact of patient-specific inflow velocity profile on hemodynamics of the thoracic aorta*, Journal of biomechanical engineering, 140 (2018).