UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

# DEVELOPMENT AND SIMULATION FOR A WALL IDENTIFICATION ALGORITHM BASED ON VELOCITY MEASUREMENTS

TESIS PARA OPTAR AL GRADO DE MAGISTER EN
CIENCIAS DE LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL MATEMÁTICO

FELIPE IGNACIO OLIVARES FERNÁNDEZ

PROFESOR GUÍA:
AXEL OSSES ALVARADO
PROFESOR CO-GUÍA :
JOAQUÍN MURA MARDONES

MIEMBROS DE LA COMISIÓN:
HANNE VAN DEN BOSCH
EDUARDO CERPA JERIA

Este trabajo ha sido parcialmente financiado por Proyecto Fondecyt 1191903 y CMM ANID BASAL FB210005

SANTIAGO DE CHILE
2022

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE
LA INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS,
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERIO CIVIL MATEMÁTICO
POR: FELIPE IGNACIO OLIVARES FERNÁNDEZ
FECHA: 2022
PROF. GUÍA: AXEL OSSES ALVARADO
PROF. CO-GUÍA: JOAQUÍN MURA MARDONES

# DESARROLLO Y SIMULACIÓN DE UN ALGORITMO PARA IDENTIFICACIÓN DE UNA PARED MÓVIL BASADA EN MEDICIONES DE VELOCIDAD

En el contexto de imágenes médicas, la recuperación no invasiva de la geometría de vasos sanguíneos a partir de mediciones de velocidad por MRI, es muy importante para el desarrollo de nuevas herramientas de análisis no invasivo para enfermedades cardíacas. Se trata de un problema inverso geométrico: recuperar parte de la frontera de un dominio móvil a partir de mediciones solamente de la velocidad del flujo y no de la presión.

Existen resultados parciales de la implementación práctica de este problema inverso en el caso de paredes estáticas y a partir solo de mediciones de flujo de salida por un borde. Sin embargo, falta desarrollar un algoritmo eficiente que utilice información parcial al interior del dominio y extienda los resultados al caso de una pared móvil. De este modo, el objetivo principal del presente trabajo es proponer e implementar un algoritmo para recuperar la forma de una pared arterial móvil que contiene a un fluido, a partir de mediciones de velocidad en un subdominio arbitrario.

En una primera parte, se propone una extensión del algoritmo existente de recuperación de una pared estática usando mediciones de borde al caso de mediciones de velocidad sobre cualquier subdominio distribuido espacialmente. Para ello es necesario resolver las ecuaciones de Navier-Stokes estacionario utilizando conceptos de derivación con respecto del dominio.

Enseguida, se propone un algoritmo para la recuperación de una pared móvil en el caso más complejo no-estacionario, lo que requiere una implementación de las ecuaciones de Navier-Stokes en interacción con estructuras en el formalismo Arbitrary Lagrangian-Euler (ALE). Para resolver el problema inverso, se plantea un problema de minimización del error de ajuste de la forma respecto a una velocidad de referencia. En la minimización, se utiliza la derivada de forma para la expresión del gradiente del funcional. El problema no estacionario se modela de hecho como un sistema quasi-estacionario en un algoritmo iterativo en el tiempo que permite resolver el caso de identificar una pared móvil.

Las simulaciones numéricas se realizan mediante la librería Fenics para la formulación de los problemas de Navier-Stokes, la cuál se realiza en dos dimensiones. En la minimización, se utiliza el algoritmo Broyden-Fletcher-Goldfarb-Shanno y se desarrolla en el lenguaje Python.

El análisis de esta tesis se puede extender a otros problemas de identificación de paredes móviles más complejos como por ejemplo el caso tridimensional o considerando una pared elástica. Este primer estudio más simple ha permitido contribuir al estudio de la estimación no invasiva de la deformación de vasos sanguíneos.

DEVELOPMENT AND SIMULATION FOR A WALL IDENTIFICATION ALGORITHM BASED ON VELOCITY MEASUREMENTS

In the context of medical imaging, the non-invasive recovery of the blood vessels geometry from velocity measurments by MRI is of the utmost importance for the development of new tools to non-invasively analize cardiovascular diseases. It's about a geometric inverse problem: the reconstruction of a mobile domain boundary based only on the flow velocity and not the pressure.

There are existing results that partially implement this inverse problem for the case of static walls and only with flow measurments at the outlet boundary. However, there is still a need to develop an efficient algorithm that could use partial information at the domain interior and could extend the results to the case of a mobile wall. Thus, the main objective of this work is to present and implement an algorithm to reconstruct the shape of a mobile wall containing a fluid, based on velocity measurments in an arbitrary subdomain.

Firstly, an extension is proposed for the existing reconstruction algorithm for static walls using boundary measurments to the case of velocity measurments over any subdomain. For this purpose, it is necessary to solve the stationary Navier-Stokes equations, applying shape derivative concepts.

Secondly, an algorithm is proposed to reconstruct a mobile wall for the more complex non-stationary case, which requires a Navier-Stokes implementation coupled with a structure in the Arbitrary Lagrangian Euler (ALE) formalism. To solve the inverse problem, a minimization problem is considered for the error adjustment with respect to a reference velocity. In the minimization, the shape derivative is used to express the functional gradient. The non-stationary problem is modeled in fact as a quasi-stationary system in an iterative algorithm over time, which allows to solve the identification problem for a mobile wall.

Numerical simulations are made using the Fenics library for the Navier-Stokes formulation of the problem, which is made in two dimensions. In the minimization, the Broyden-Fletcher-Goldfarb-Shanno algorithm is used and is implemented in the Python language.

The analysis of this thesis can be extended to more complex identification problems for mobile walls like the three-dimensional case or considering coupling with an elastic wall. This first simpler study has allowed to contribute to the study of non-invasive deformation estimation of blood vessels.

*A quién haya hecho de este viaje uno más ameno.*

# Acknowledgement

The first whom I have to thank are my parents, who have given everything for me and are the ones without whom I wouldn't be here today. Thanks to my aunt Aldy, who has always supported me inconditionally. To my brothers, for the good times in family. To Paula, who has been there when I've needed it.

I have to thank the friends I've met since school, Patricio Rivera, Alejandro Silva, Felipe Aedo, Sebastián Melo, Matías Frick. Thanks for all the parties and for your companionship.

From my time at the Department of Mathematics, the 435 office has been an amazing place full of amazing people. I have to thank everyone who made it so enjoyable, Diego Marchant, Felipe Matus, Kevin Contreras, Manuel Suil, Vicente Ocqueteau, Pablo Arratia, Francisco Sanhueza, Obed Ulloa, Tabita Catalán, Monserrat Morales, Tomás Ahumada, Camila Zárate and many more who made the place a home. Thanks for all the memes and UNO games.

Thanks to all the friends I made during my time in the Univerity, Diego Corvalán, Miguel Piña, Andy Area, Daniela Pollarolo, Camilo Campos, Ignacio Reyes, Jorge Vidal, Beatriz Latorre, Alex Millán, Carlos Meneses, Matías Moreno. Thank you for all the good times we've had and all the ones that will come.

Last but not least, thanks to my advisers Axel Osses and Joaquín Mura, their help has been invaluable. Thanks for all the patience throughout this process and for all the comments given and all the experiences shared.

# Contents

# List of Figures

# Introduction and bibliographical discussion

In the context of medical images, there exist unanswered questions that give rise to the use of mathematical modeling and numerical simulations as a tool for social benefit. These questions range from hemodynamic behavior, methods and technique development that allows higher resolution on magnetic resonance images, to biomarker identification based on magnetic resonance images for improving cardiac disease diagnosis. The last question is of particular interest for this work, magnetic resonance images are a non-invasive source of information inside blood vessels, so it is imperative to develop methods that make use of this information. One type of blood vessel, the arteries, deliver blood away from the heart to one or more parts of the body as part of the circulatory system and are made up of different layers, mostly collagen and elastic tissue. The elasticity of arteries play a fundamental role on diseases like artherosclerosis, which is the cause of almost 18 million deaths a year worldwide in 2016 (> 30% of total deaths) and near 28,400 deaths in Chile (approx. 25%) in 2019 [9].

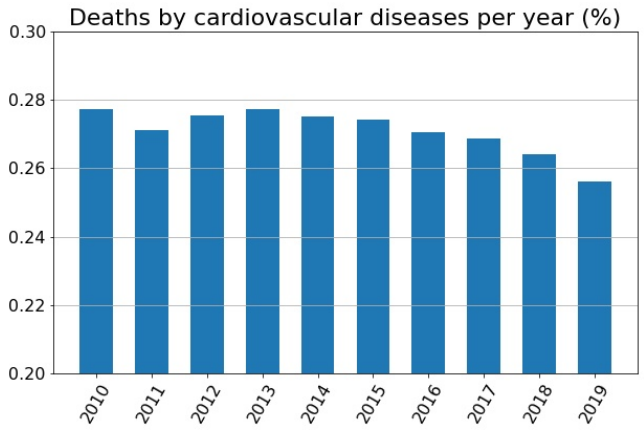


Figure 1: Percentage of deaths by cardiovascular disease compared to total deaths per year.

There are a variety of shape optimization techniques and all of them rely on the optimization of a given function which depends on the geometry or shape. The difference is how these shapes are parametrized, i.e, how its geometry and its variation can be described. The ways in which these shapes and variations can be parametrized are, for example, the vertices of a mesh representation and the vector map associated with the deformation [27], topologically through the insertion of holes in the domain and how this affect the function being optimized [24], using the contour of a level set given by a certain function [22], or by adding a permeability parameter to identify the shape based on penalization [3, 1, 2, 5]. Further details on optimization techniques applied to fluid dynamics can be found in [19, 20].

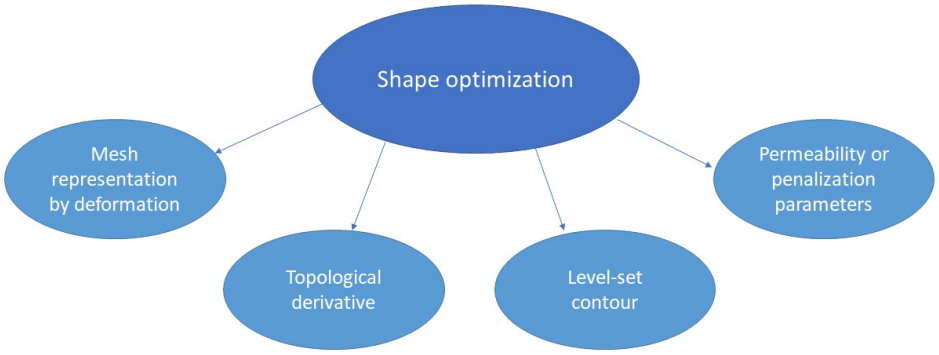


Figure 2: Diagram of methods used for shape optimization.

The implementation of these methods, i.e, the computations on the sensibility problem over changes in the domain, can be done through automatic differentiation, which requires understanding of the methods involved as a whole and expertise on the numerical resolution of the flow problem. On the other hand, the calculations can be done solving the minimization problem analytically, which involves mathematical tools to express the derivative of the optimized criterion (involving the adjoint problem) and then discretizing it for the numerical implementation, gaining the advantage of making more independent the process of differentiation and the numerical resolution.

Through the *4D Flow* sequence [26], a technique in which phase contrast is used on a magnetic resonance exam three-dimensionally in space, is possible to obtain the time evolution of blood flow inside blood vessels. This technique can be used in a variety of applications, including pressure detection [21, 8, 7], obstacle identification via penalization method [3, 1, 2, 5] and displacement computations as in elastography exams [16]. Although magnetic resonance images allow one to visualize blood vessels such as arteries, there's excessive noise around the contour, therefore, they are unsuitable by themselves to estimate biomarkers such as elasticity. There have been advancements in the development of tools that allow us to identify the shape of the arterial walls based on velocity measurements [27], but these measurements are restricted to the domain border and only consider static walls.

This work has two main objectives. The first one, is to extend the mathematical modeling made in [27] to being able to use velocity data measured not only on the domain border but inside of it, and allow the identification of the evolution of a mobile wall making it possible to describe the deformation of an arterial wall through time. The second objective consists in the development of an algorithm that efficiently, and via FEniCS library, allows the reconstruction of a mobile wall shape based on velocity data measured inside a domain.

From the modeling and numerical simulations point of view, it is worth mentioning that assumptions have to be made in order to reach satisfactory results in reasonable time, and for this reason there are compromises in the implemented model compared with an ideal one. For example, the

domain walls are considered as an imposed deformation which one wishes to recover, even though the ideal model would consider describing the arterial wall as an elastic border solving the coupled problem between flow and structure. Blood is modeled as an incompressible laminar fluid instead of a turbulent one for the simplicity it provides in comparison, and the problem is defined in two dimensions for the same reason. In reality, blood is composed of blood cells, mainly red blood cells, white blood cells and platelets suspended in plasma, but it is necessary to make assumptions nonetheless, taking into account the fidelity of the solution. Assuming a two-dimensional domain is reasonable (at least as a beginning) based on the axial symmetry [10] of an arterial section, and therefore if one can reconstruct the deformation in two dimensions, a next step and possible extension to what this work propose would be to reconstruct the deformation in three dimensions.

This thesis is organized as follows: **Chapter 1** describes the existing algorithm to reconstruct deformation and leads the way to understand the models used to describe the behavior of blood as a fluid. **Chapter 2** presents the first objective of this work, which extends the existing results to velocity data measured in a subdomain. **Chapter 3** describes the second of the main objectives, the extension to the mobile wall case. Lastly, **Chapter 4** presents results of the numerical simulations performed for each of the formerly described cases.

# Chapter 1

# A shape reconstruction algorithm

In this chapter, we will discuss the fluid model used in both the stationary and dynamic case, providing details on the boundary conditions used and the numerical implementation. Later, the algorithm in [27] will be summarized, as it is the basis on which this work was made. This algorithm consists of the reconstruction of a static wall shape based on a given velocity over a region of the boundary through iterations that minimize our function of interest:

$$J(\Omega) = \frac{1}{2}\int_{\Gamma^{out}} |u - u_{ref}|^2 \mathrm{d}s.$$

The descent direction is then obtained through the shape derivative in a process based on Newton iterations that will be described in this chapter.

## 1.1 Fluid model

Let us consider a bounded domain $\Omega \subset \mathbb{R}^N$ containing a Newtonian fluid with kinematic viscosity $\mu > 0$. The boundary $\partial\Omega$ is made of three disjoint regions: $\partial\Omega = \Gamma_{in} \cup \Gamma_{out} \cup \Gamma$. A velocity profile $u_{in}$ is imposed over $\Gamma_{in}$ (*inlet*), $\Gamma_{out}$ is free from surface forces, and $\Gamma$ is imposed with no slip boundary conditions. For this model, the incompressible Navier-Stokes equations are considered.

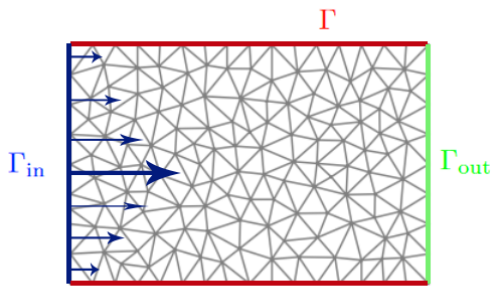


Figure 1.1: Domain mesh used and boundaries defined.

### 1.1.1 Stationary case

For the stationary Navier Stokes model, we describe the fluid as a pair $(u, p)$ where $u : \Omega \to \mathbb{R}^N$ is the velocity field and $p : \Omega \to \mathbb{R}$ the pressure. This pair solves the following equation:

$$\begin{aligned} -\mu\Delta u + (u \cdot \nabla)\, u + \nabla p &= 0 && \text{en } \Omega \\ \operatorname{div}(u) &= 0 && \text{en } \Omega \\ u &= u_{in} && \text{en } \Gamma_{in} \\ u &= 0 && \text{en } \Gamma \\ \sigma(u,p)n &= 0 && \text{en } \Gamma_{out}. \end{aligned} \tag{1.1}$$

In this equation, $n$ is the exterior normal vector and $\sigma(u, p)$ is the stress tensor defined as

$$\sigma(u, p) = 2\mu \mathrm{e}(u) - pI, \text{ where } \mathrm{e}(u) = \frac{1}{2}(\nabla u^T + \nabla u).$$

The conditions imposed in this problem are common in fluid models involving this type of domain geometry. Zero velocity on $\Gamma$ often called "no-slip condition" represents the friction against the wall. The inlet is imposed with a Dirichlet condition $u_{in}$ usually chosen as a parabolic function, which ensures continuity at the corners. Lastly, the Neumann condition at $\Gamma_{out}$ is known as free surface force condition, which is useful for modelling flow where the outlet can lead to different shapes.

Regarding the type of flow, in the adimensional problem ( 1.1) the Reynolds number $R\mathrm{e}$ is proportional to $1/\mu$ and can be used as an indicator for the type of flow. In this work, we consider a sufficiently low Reynolds number (under 2000) so the convective term is less prevalent and therefore the flow is laminar instead of turbulent.

Though it is well known that a general solution can not be guarantied for the Navier-Stokes equations, in this case the problem ( 1.1) is well-posed for sufficiently large values of viscosity $\mu$ [27, 25, 13, 14] (keeping the Reynolds number on laminar flow). For this system, we will consider the domain to be bounded with a Lipschitz-continuous boundary, and $u_{in}$ can be extended to a function $\tilde{u}_{in}$ in $H^{1/2}(\partial\Omega)^N$ to ensure this problem has a weak solution $(u, p) \in H^1(\Omega)^N \times L^2_0(\Omega)$, with $L^2_0(\Omega) := \{p \in L^2(\Omega), \int_\Omega p \mathrm{d}x = 0\}$ and $N \leq 3$ [27, 25, 13, 14]. Therefore, in the next pages we will consider this problem well-posed.

The numerical implementation of system ( 1.1) with finite element method is complicated firstly due to the nonlinearity of the convective term $(u \cdot \nabla)u$ and secondly because it is a saddle point problem. The first problem can be solved using a linearization method such as Newton's method, in this case it was used MUMPS (Multifrontal Massively Parallel sparse direct Solver), which is a linear solver dedicated to large sparse systems taking advantage of parallel computing. The second problem is solved by choosing appropriate finite element spaces for the velocity $u$ and pressure $p$. $\mathbb{P}2 - \mathbb{P}1$ Lagrangian elements (Taylor-Hood elements) are used for the velocity and pressure respectively ensuring the so-called *Brezzi inequality* holds, and therefore the system is invertible [27, 12, 11].

### 1.1.2 Dynamic case

In the non-stationary model, in addition to the domain $\Omega$ we consider a time interval $I = [0, T]$ and the velocity $u : \Omega \times I \to \mathbb{R}^N$ and pressure $p : \Omega \times I \to \mathbb{R}$ which solves the system

$$\begin{aligned} \frac{\partial u}{\partial t} - \mu \Delta u + (u \cdot \nabla)\, u + \nabla p &= 0 && \text{in } \Omega_\theta \times I \\ \operatorname{div}(u) &= 0 && \text{in } \Omega_\theta \times I \\ u &= u_{in} && \text{on } \Gamma_\theta^{in} \times I \\ u &= 0 && \text{on } \Gamma_\theta \times I \\ \sigma(u, p)n &= 0 && \text{on } \Gamma_\theta^{out} \times I \\ u(x, 0) &= u_0 && \text{on } \Omega_0. \end{aligned} \tag{1.2}$$

In addition to the boundary conditions described in the stationary case, we imposed an initial condition $u_0$, which for the purpose of the simulations required will be considered the zero function in $L^2(\Omega_0)$. The domain $\Omega \times I$ is still bounded and with Lipschitz-continuous boundary and for the inlet, $u_{in}$ can be extended to a function $\tilde{u}_{in} \in H^{1/2}(\partial\Omega) \times I$, thus ensuring there exists a unique solution $u \in L^2(I; H^1(\Omega_\theta)) \cap H^1(I; (H^1(\Omega_\theta))')$ at least for dimension $N = 2$ [25, 17].

It is worth noticing that, added to the non-stationary quality of the solution, the domain also depends on time. For notation convenience, we define it as $\Omega_\theta$ looking ahead for the optimization problem where the deformation is a function of time $\theta(t)$.

This system can be solved using the same finite element spaces ($\mathbb{P}2 - \mathbb{P}1$) and there are numerous methods to solve system ( 1.2), like *Stationary iterative methods*, *LU solver* or *Krylov subspace methods* better described in [11, 13, 14]. In this case, Newton-Krylov method was used, which is typically used for this system since it can be used in more general systems of linear equations. Inside the category of Krylov subspace methods there are different specific methods like *Lanzos iteration*, *MinRes*, *GMRES*, *BiCG method* and *BiCGSTAB method* among which we chose the *Generalized minimal residual method* (GMRES) solver by comparing them on this specific problem.

## 1.2 Summary for the stationary identification algorithm

This algorithm relies strongly on the concept of shape optimization, which consists of solving a minimization problem where the function is dependent on the shape of the domain. In a general case, the problem can be written as the minimization of a function $J$ over a set of admissible domains $\mathcal{O}_{\text{ad}}$,

$$\begin{aligned} &\underset{\Omega \in \mathcal{O}_{\text{ad}}}{\text{minimize}} && J(\Omega) \\ &\text{subject to} && E(\Omega) = 0. \end{aligned}$$

The main function of our interest is the quadratic difference between a reference velocity and a velocity obtained solving the Navier Stokes equations over the domain being evaluated.

$$J(\Omega) = \frac{1}{2}\int_{\Gamma^{out}} |u - u_{ref}|^2 \mathrm{d}s.$$

However, one first needs to establish the way this function will be minimized. There are various methods available, like "*Gradient Method*", "*Level-set Method*" and methods based on a permeability parameter.

Gradient method, which is the one used in this work, consists in obtaining the derivative of the function $J$ with respect to the shape of the domain $\Omega$ and apply a minimization algorithm such as steepest-descent or quasi-Newton methods. Either method require computation of the objective function's derivative. In general (just assuming $\Omega \subset \mathbb{R}^N$), this is difficult to compute, as there is no canonical space on which to apply the minimization algorithm. Therefore, in [15] the derivative is defined as the Fréchet derivative.

A convenient way to calculate the Fréchet derivative is to consider $\theta : \mathbb{R}^N \to \mathbb{R}^N$ as a vector field with norm close to zero and $\Omega_\theta$ defined based on Hadamard method [15] as $\Omega_\theta = (I + \theta)\Omega$.

$$J(\Omega_\theta) = J(\Omega) + J'(\Omega)(\theta) + o(\theta).$$

This allows us to define the derivative as $J'(\Omega)(\theta)$. Then, it is natural to define the following sets:

**Definición 1.1** *Deformation space*

$$\Theta = \{\theta \in W^{1,\infty}(\mathbb{R}^N) \mid \theta = 0 \text{ on } \Gamma^{in} \cup \Gamma^{out}\}.$$

**Definición 1.2** *Admissible domains*

$$O_{ad} = \{\Phi(\Omega) \subset \mathbb{R}^N \mid \Phi = (I + \theta),\ \theta \in \Theta\}.$$

Regarding functionals and their shape derivatives, an interesting first result proven to be useful is the volume and perimeter of a determined shape $\Omega$ as proven in [27, 15, 4].

**Theorem 1.3** *Let $\Omega$ be a piecewise $C^2$ shape. Then,*

1. *$Vol(\Omega)$ is shape differentiable and its derivative is:*

$$\forall \theta \in \Theta,\ Vol'(\Omega)(\theta) = \int_{\partial\Omega} \theta \cdot n \ \mathrm{d}s.$$

2. *$Per(\Omega)$ is shape differentiable and its derivative is:*

$$\forall \theta \in \Theta,\ Per'(\Omega)(\theta) = \int_{\partial\Omega} \kappa\theta \cdot n \ \mathrm{d}s,$$

*where $\kappa : \partial\Omega \to \mathbb{R}$ is the mean curvature of $\partial\Omega$.*

For the functional $J(\Omega)$ we are interested in, defined over the same domain from Chapter 1,

$$J(\Omega) = \int_{\Gamma^{out}} |u - u_{ref}|^2 \mathrm{d}s, \tag{1.3}$$

where $u_{ref} \in L^2(\Gamma^{out})$ is a reference velocity and $u$ is the solution for system ( 1.1). The following result proven in [27] holds,

**Theorem 1.4** *Let $\Omega \in O_{ad}$, $\mu$ sufficiently large kinematic viscosity and u solution of system ( 1.1). Then,*

$$J'(\Omega)(\theta) = \int_{\Gamma} 2\mu \mathrm{e}(u) : \mathrm{e}(v)\ \theta \cdot n \mathrm{d}s, \tag{1.4}$$

*where $(v, q)$ is solution of the following adjoint problem:*

$$\begin{aligned} -\mu\Delta v + (\nabla u)^T v - (\nabla v) u + \nabla q &= 0 && \text{in } \Omega \\ div(v) &= 0 && \text{in } \Omega \\ v &= 0 && \text{on } \Gamma \cup \Gamma^{in} \\ \sigma(v, q)n + (u \cdot n)v &= u - u_{ref} && \text{on } \Gamma^{out}. \end{aligned} \tag{1.5}$$

We are in position to describe the algorithm applied to system ( 1.1) to minimize $J$ in ( 1.3).

**Algorithm 1** Deformation identification

**initialization:**
    Fix Navier-Stokes and minimization parameters
    Open velocity reference file
    Initialize mesh for $\Omega_0$
**while** $||\theta|| > \varepsilon$ **do**
    Calculate solution $(u, p)$ to the Navier-Stokes system ( 1.1)
    Calculate solution $(v, q)$ to the Adjoint system ( 1.5)
    Compute shape gradient of $J$
    Infer a descent direction $\theta$ by solving ( 1.8)
    Find a descent step
    Update mesh to $(I + \theta)\Omega$
    Re-mesh if necessary
**end while**

It is worth mentioning that as explained in [27], the shape derivative of a general functional $F(\Omega)$ has the form:

$$F'(\Omega) = \int_{\Gamma} \phi\ \theta \cdot n \mathrm{d}s =: (\phi, \theta \cdot n)_{L^2(\Gamma)},$$

where the scalar function $\phi : \Gamma \to \mathbb{R}$ is the shape gradient of F; see [27, 15]. In particular, $F'(\Omega)$ only depends on the normal component $\theta \cdot n$ on the free boundary $\Gamma$.

To obtain $\theta$, in [27] a variational problem gives as a result a descent direction for $F(\Omega)$. This idea consists on defining an inner product $(\cdot,\cdot)_V$ over a Hilbert space $V$ and finding $\theta \in V$ such that

$$\forall \psi \in V,\ (\theta,\psi)_V = F'(\Omega)(\psi) = \int_\Gamma \phi\ \psi \cdot n\ \mathrm{d}s. \tag{1.6}$$

This ensures that $\theta$ is a descent direction,

$$F'(\Omega)(-\theta) = -(\theta,\theta)_V < 0.$$

Choosing in particular the following space and inner product,

$$V = \{v \in H^1(\Omega)^N,\ v|_{\Gamma^{in}\cup\Gamma^{out}} = 0\},$$

$$(\theta,\psi)_V = \int_\Omega A\mathrm{e}(\theta) : \mathrm{e}(\psi)\ \mathrm{d}x,\ A\mathrm{e} = 2\mu\mathrm{e} + \lambda tr(\mathrm{e}). \tag{1.7}$$

The solution $\theta$ solves the following system, which is common in fluid structure problems [23] to extend a deformation defined on a boundary into the interior, ensuring a certain elaticity is preserved.

$$\begin{aligned} -\mathrm{div}(A\mathrm{e}(\theta)) &= 0 && \text{in } \Omega \\ \theta &= 0 && \text{on } \Gamma_{in} \cup \Gamma_{out} \\ \sigma n &= \phi n && \text{on } \Gamma. \end{aligned} \tag{1.8}$$

It is worth mentioning that the algorithm can be applied to a general functional $F(\Omega)$, provided there is an expression for the shape gradient, which can be computed similarly to Theorem 1.4.

# Chapter 2

# Extension to subdomain measures

The algorithm previously described, even though it allows reconstructing the shape based on measurements on $\Gamma^{out}$, one would like to be able to work with more complex shapes, and eventually extend it to the temporal problem. For this purpose, it is necessary to be able to use information at the domain interior, so one can avoid convergence problems. Through MRI, and in particular through 4D Flow techniques, it is possible to obtain velocity data at the interior of vessels, therefore, the first goal in this work is to extend the existing results to measures of velocity at the domain interior over any subset $D$ of $\Omega$.

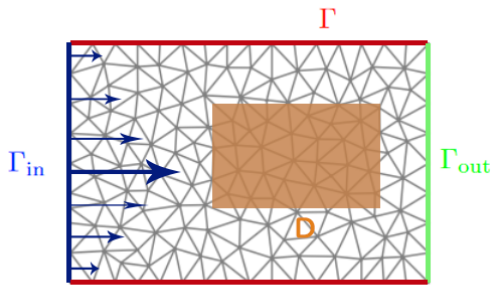


Figure 2.1: Domain mesh with subdomain D.

Let $D \subset \Omega$ be a subdomain, $u_{ref} : D \to \mathbb{R}^2$ a reference velocity and the functional

$$J(\Omega) = \frac{1}{2} \int_D |u - u_{ref}|^2 \mathrm{d}x. \tag{2.1}$$

In reality, the observation window is not necessarily a subset of the vessel domain, but for the purpose of this analysis we require it to be fully contained in the shape $\Omega$ in which we evaluate the functional. For this reason, we consider the open set $(\overline{D^{real} \cap \Omega})^{\circ} =: D$, where $D^{real}$ is the real observation window one could get from 4D Flow measures. Therefore, we are still writing the functional as dependent on $\Omega$.

## 2.1 Shape derivative

Following [27], one needs to obtain a derivate of the form $J'(\Omega)(\theta)$ using the adjoint problem associated to the functional $J(\Omega)$. This problem is calculated writing the Lagrangian related to $J(\Omega)$ and applying the variational principle to the system. In this case, the Lagrangian is written as follows:

$$\mathcal{L}(u,p,\theta,v,q) = \frac{1}{2}\int_D |u-u_{ref}|^2 \mathrm{d}x - \int_{\Omega_\theta} (2\mu \mathrm{e}(u) : \mathrm{e}(v) + (\nabla u u \cdot v) - p\mathrm{div}(v) - q\mathrm{div}(u))\,\mathrm{d}x,$$

$$D_u\mathcal{L}(u,p,\theta,v,q)h = \int_D (u-u_{\mathrm{ref}})h\mathrm{d}x - \int_{\Omega_\theta} 2\mu\mathrm{e}(h) : \mathrm{e}(v) - q\mathrm{div}(h) + (\nabla u)^T vh + (\nabla hu)\, v\, \mathrm{d}x = 0,$$

$$D_p\mathcal{L}(u,p,\theta,v,q)k = -\int_{\Omega_\theta} k\mathrm{div}(v)\mathrm{d}x = 0.$$

After some simple computations, the following adjoint problem arises:

$$\begin{aligned} -\mu\Delta v + (\nabla u)^T v - (\nabla v)\, u + \nabla q &= (u-u_{ref})\mathbb{1}_D && \text{in } \Omega \\ \mathrm{div}(v) &= 0 && \text{in } \Omega \\ v &= 0 && \text{on } \Gamma \cup \Gamma^{in} \\ \sigma(v,q)n + (u\cdot n)v &= 0 && \text{on } \Gamma^{out}. \end{aligned} \tag{2.2}$$

This system is in almost all of its terms identical to the one obtained for measurements over $\Gamma^{out}$ except for $(u-u_{\mathrm{ref}})\mathbb{1}_D$, which now plays a role in the first equation rather than a boundary condition.

The shape derivative of $J(\Omega)$ by chain rule can be written as

$$J'(\Omega)(\theta) = \int_D (u-u_{ref})\cdot u'\mathrm{d}x. \tag{2.3}$$

This expression is unsuitable to work with because of $u'$, which is the solution for the sensibility problem ( 2.4). It is difficult to compute without knowing the solution for the deformation, as is appears in the boundary condition on $\Gamma$. Thus, one would rather make use of the adjoint problem to rewrite the functional in a way that is easier to compute.

**Theorem 2.1** (Sensibility problem) *The derivative of the stationary Navier-Stokes equations ( 1.1)*

*with respect to changes in the domain is the solution to*

$$\begin{aligned} -\mu\Delta u' + (\nabla u)\, u' + (\nabla u')\, u + \nabla p' &= 0 && \text{in } \Omega \\ div(u') &= 0 && \text{in } \Omega \\ \sigma(u', p')n &= 0 && \text{on } \Gamma_{out} \\ u' &= 0 && \text{on } \Gamma_{in} \\ u' &= -\left(\frac{\partial u}{\partial n}\right)(\theta \cdot n) && \text{on } \Gamma. \end{aligned} \tag{2.4}$$

Proof. See [15, Theorem 5.33] □

The main result regarding the extension to measurements on the subdomain $D$ comes from Theorem 2.2, which rewrites the functional derivative in terms of the adjoint problem rather than the sensibility problem.

**Theorem 2.2** (Shape derivative) *Let $(u, p)$ and $(v, q)$ be solutions of ( 1.1) and ( 2.2) respectively, then the shape derivative of $J(\Omega) = \int_D |u - u_{ref}|^2 \mathrm{d}x$ can be written as:*

$$J'(\Omega)(\theta) = \int_\Gamma 2\mu \mathrm{e}(u) : \mathrm{e}(v)\theta \cdot n \mathrm{d}s.$$

Proof. Multiplying the first equation of ( 2.2) by $u'$ and integrating, we obtain

$$\begin{aligned} J'(\Omega)(\theta) &= \int_\Omega [-\mu\Delta v + (\nabla u)^T v - (\nabla v)u + \nabla q]u' \mathrm{d}x \\ &= \int_\Omega [2\mu \mathrm{e}(v) : \mathrm{e}(u') + (\nabla u)u' \cdot v - (\nabla v)u \cdot u']\mathrm{d}x - \int_{\partial\Omega} \sigma(v, q)n \cdot u' \mathrm{d}s. \end{aligned}$$

Multiplying the first equation of ( 2.4) by $v$ and then integrating, the following equation arises:

$$\int_\Omega [-\mu\Delta u' + (\nabla u)u' + (\nabla u')u + \nabla p']v \mathrm{d}x = 0.$$

Thus integrating by parts,

$$\int_\Omega [2\mu \mathrm{e}(u') : \mathrm{e}(v) + (\nabla u)u' \cdot v + (\nabla u')u \cdot v]\mathrm{d}x - \int_{\partial\Omega} \sigma(u', p')n \cdot v \mathrm{d}s = 0.$$

Then, we can rewrite the functional derivative as follows:

$$J'(\Omega)(\theta) = \int_\Omega -(\nabla v)u \cdot u' \mathrm{d}x - \int_{\partial\Omega} \sigma(v, q)n \cdot u' \mathrm{d}s - \int_\Omega (\nabla u')u \cdot v \mathrm{d}x + \int_{\partial\Omega} \sigma(u', p')n \cdot v \mathrm{d}s.$$

Integrating by parts one more time, this identity holds:

$$\int_\Omega -(\nabla v)u \cdot u' \mathrm{d}x - \int_\Omega (\nabla u')u \cdot v \mathrm{d}x = -\int_{\partial\Omega} (v \cdot u')(u \cdot n)\mathrm{d}s.$$

Therefore,

$$J'(\Omega)(\theta) = -\int_{\partial\Omega} (v \cdot u')(u \cdot n)\mathrm{d}s - \int_{\partial\Omega} \sigma(v,q)n \cdot u' \mathrm{d}s + \int_{\partial\Omega} \sigma(u',p')n \cdot v \mathrm{d}s.$$

By applying the boundary conditions $\sigma(v,q)n + (u \cdot n)v = 0$ on $\Gamma_{out}$, $\sigma(u',p')n = 0$ on $\Gamma_{out}$ and $v = 0$ on $\Gamma \cup \Gamma_{in}$,

$$J'(\Omega)(\theta) = \int_\Gamma \frac{\partial u}{\partial n}\sigma(v,q)n \cdot (\theta \cdot n)\mathrm{d}s = \int_\Gamma 2\mu \mathrm{e}(u) : \mathrm{e}(v)\theta \cdot n \mathrm{d}s.$$

Thus, the shape derivative in the case of subdomain measurements matches the expression one can obtain for the case of measurements only on the outlet border. □

## 2.2 Determination of the domain deformation field

In order to determine the domain deformation that represents the optimal shape, we make use of Theorem 2.2 to construct the gradient $\nabla J(\Omega)$. Once constructed, it is possible to minimize the functional with any method based on gradient available, and for this work it was decided to change the method used in [27] for a quasi-Newton method. The minimization algorithm used was BFGS or Broyden–Fletcher–Goldfarb–Shanno algorithm, which consists of a method that approximates the functional Hessian using the gradient and thus makes a better guess on which direction is a local minimum. After determining a descent direction, one needs to apply a linear search to find the appropriate step so the functional decreases significantly. To ensure this, the algorithm verifies two conditions named *Wolfe conditions*, which ensures significant decrease. We will see that this is a double edge sword, in the sense that in some cases no appropriate step can be found. For this reason, the Wolfe conditions were relaxed, sacrificing computation time for a more reliable algorithm.

The algorithm used consists in an adaptation of the method described in [27] to our case of interest, taking into account the extension to velocity measurements over a subdomain $D$. This algorithm can be summarized as shown in Algorithm 2.

**Algorithm 2** Deformation identification based on subdomain velocity measurements

**Initialization:**
    Fix Navier-Stokes and minimization parameters
    Open velocity reference file
    Initialize mesh for $\Omega_0$
**How to compute $J(\theta)$ :**
    Move mesh from $\Omega_0$ to $\Omega_\theta$
    Solve Navier-Stokes
    Compute $J(\Omega_\theta)$ based on equation ( 2.1)
    Move mesh from $\Omega_\theta$ to $\Omega_0$
    Return value of $J(\Omega_\theta)$
**How to compute $\nabla J(\Omega)(\theta)$ :**
    Move mesh from $\Omega_0$ to $\Omega_\theta$
    Solve Navier-Stokes
    Solve Adjoint problem
    Compute shape gradient as in Theorem ( 2.2)
    Evaluate $\nabla J'(\Omega)(\theta)$
    Move mesh from $\Omega_\theta$ to $\Omega_0$
    Return $\nabla J'(\Omega)(\theta)$
**Execute BFGS routine**

In order to lessen the computational cost of evaluating the functional $J$ and its gradient, one needs to represent the domain or shape $\Omega$ in a way that is easily codable. For this purpose, we introduce the following definitions:

**Definición 2.3** *Let $\Omega_0$ be a mesh for the rectangle $[0, 1.5] \times [0, 1]$.*

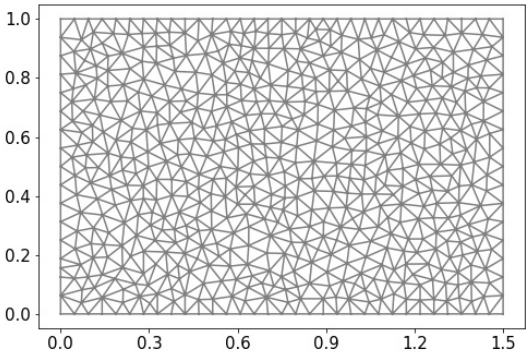


Figure 2.2: Reference mesh used for $\Omega_0$.

From now on, an arbitrary domain $\Omega$ will be represented based on the reference domain $\Omega_0$ as

$$\Omega = \Omega_\theta := (I + \theta)\Omega_0.$$

Given $u_{ref}$ defined over a subdomain $D$, the goal is to find $\theta$ that minimizes the functional

$$J(\Omega_\theta) = \frac{1}{2}\int_D |u - u_{ref}|^2 \mathrm{d}x.$$

Through Theorem 2.2, the shape derivative can be expressed as an integral over the free boundary,

$$J'(\Omega)(\theta) = \int_\Gamma 2\mu \mathrm{e}(u) : \mathrm{e}(v)(\theta \cdot n)\mathrm{d}s,$$

where $u$ is solution of the Navier-Stokes problem ( 1.1) in $\Omega$ and $v$ is solution for the adjoint problem

$$\begin{aligned} -\mu\Delta v + (\nabla u)^T v - (\nabla v)\, u + \nabla q &= (u - u_{ref})\mathbb{1}_D && \text{in } \Omega \\ \mathrm{div}(v) &= 0 && \text{in } \Omega \\ v &= 0 && \text{on } \Gamma \cup \Gamma^{in} \\ \sigma(v, q)n + (u \cdot n)v &= 0 && \text{on } \Gamma^{out}. \end{aligned}$$

From the shape derivative $J'(\Omega)(\theta)$, the gradient can be obtained evaluating in the standard basis of $\mathbb{R}^{N*V}$, where $V$ is the number of mesh vertices and $N = 2$, this way we obtain each gradient component. Computationally, the cost of evaluating the shape derivative is very similar to that of the functional, but the gradient requires computing the shape derivative for each component, i.e., $2V$ computations, thus encoding the deformation over the mesh. For the particular implementation of this method, we coded the domain $\Omega_\theta$ as a vector containing only the deformation over mesh vertices on the free boundary $\Gamma$, this allows to solve the minimization problem in a space of a much smaller dimension, improving the efficiency of the algorithm. Another practical implementation used in this particular case was modifying the BFGS algorithm, in particular the linear search section, so the previous step can be used when the Wolfe conditions are not satisfied. Without this modification the algorithm tends to halt, and no step is good enough to meet the Wolfe conditions, to allow the algorithm to use the previous step is a way of escaping from local minimums and potentially finding a better solution at the cost of time.

Once performed the minimization and obtained the deformation over the vertices on $\Gamma$, the next step is to extend this deformation into the interior. This extension in [27] is done implicitly through solving the pseudo-elastic problem:

$$\begin{aligned} -\mathrm{div}(A\mathrm{e}(\theta)) &= 0 && \text{in } \Omega \\ \theta &= 0 && \text{on } \Gamma_{in} \cup \Gamma_{out} \\ \sigma n &= \phi n && \text{on } \Gamma, \end{aligned}$$

where $A\mathrm{e} = 2\mu\mathrm{e} + \lambda tr(\mathrm{e})$, but in this case, we decided to extend the deformation using a Harmonic problem. This ensures that the final deformation is of class $C^2$ if the deformation on $\Gamma$ is $C^2$. Both methods are equally valid and both are used in ALE formulations for Fluid Structure problems [23]. Whichever method is used to extend the deformation into the interior, is important to ensure that the final deformation is regular enough to preserve the quality of the mesh. For this reason, all the systems are in the Eulerian framework, which simplifies the formulation.

## 2.3 Testing on symmetric domain

The method previously described was applied not only on the example given in [27], it is necessary to test it on a deformation that could easily extend to the next step, which is to consider time dependency. The domain to be reconstructed is a symmetric one, where the upcoming movement will easily represent a pulse through the vessel. This change in the domain resulted in the algorithm being unable to escape from a local minimum, far away from what can be called an "acceptable domain". For this reason, we incorporated two new terms to the functional to control its volume and perimeter, thus preventing the final shape to be unreasonable. The new terms correspond to the difference in $L^2$ norm with volume and perimeter of the reference domain, respectively.

$$
\begin{aligned}
J(\Omega) &= J_{main}(\Omega) + J_{vol}(\Omega) + J_{per}(\Omega),\\
J_{main}(\Omega) &= \int_D |u - u_{ref}|^2 \mathrm{d}x,\\
J_{vol}(\Omega) &= \frac{b}{2}\left(\int_\Omega \mathrm{d}x - \int_{\Omega_0} \mathrm{d}x\right)^2,\\
J_{per}(\Omega) &= \frac{c}{2}\left(\int_\Gamma \mathrm{d}s - \int_{\Gamma_0} \mathrm{d}x\right)^2.
\end{aligned}
$$

These terms, although they fulfill their goal of controlling the shape volume and perimeter, they raise the problem of adjusting two new parameters. Incorporating this modification into the algorithm previously described only requires the computation of the shape derivatives for each of these terms and adding them to the shape derivative obtained in Theorem 2.2, making use of lemmas 2.4 and 2.5.

**Lemma 2.4** (Volume shape derivative)

$$
J'_{vol}(\Omega)(\theta) = b\left(\int_\Omega \mathrm{d}x - \int_{\Omega_0} \mathrm{d}x\right)\int_\Gamma (\theta \cdot n)\mathrm{d}s.
$$

**Lemma 2.5** (Perimeter shape derivative)

$$
J'_{per}(\Omega)(\theta) = c\left(\int_\Gamma \mathrm{d}s - \int_{\Gamma_0} \mathrm{d}s\right)\int_\Gamma \kappa(\theta \cdot n)\mathrm{d}s.
$$

Then, coding the domain as previously done, i.e., a vector containing the deformation over the vertices on the free boundary, the problem can be solved numerically making use of the method described in this chapter.

# Chapter 3

# Extension to a mobile wall

The final goal is to identify the deformation of a mobile wall through a time interval. The natural and naive way of doing it is to apply the same method just described to the time-dependent Navier-Stokes equations, minimizing the functional

$$J(\Omega) = \frac{1}{2}\int_0^T \int_D |u(x,t) - u_{ref}(x,t)|^2 \mathrm{d}x\mathrm{d}t. \tag{3.1}$$

The difference in this case is that this method would try to solve globally in time, and this could sacrifice precision at some time steps. Besides, the nature of solving the problem globally implies that it is not possible to obtain partial results mid-computation. For this reason, we developed an alternative iterative method that optimize each time step individually and produces results as they are being calculated.

## 3.1 Direct method

The direct way to solve for the non-stationary case is to follow the algorithm previously described considering the functional integrated over the time interval and solve the Navier-Stokes equations in the dynamic formulation. The equations in the Eulearian framework are:

$$\begin{aligned}
\frac{\partial u}{\partial t} - \mu\Delta u + (u\cdot\nabla)\,u + \nabla p &= 0 && \text{in } \Omega_\theta \times I \\
\mathrm{div}(u) &= 0 && \text{in } \Omega_\theta \times I \\
u &= u_{\mathrm{in}} && \text{on } \Gamma_\theta^{in} \times I \\
u &= 0 && \text{on } \Gamma_\theta \times I \\
\sigma(u,p)n &= 0 && \text{on } \Gamma_\theta^{out} \times I \\
u(x,0) &= u_0 && \text{in } \Omega.
\end{aligned} \tag{3.2}$$

Let us consider the functional ( 3.1), then in the spirit of Theorem 2.2 proven for the extension to subdomain measurements, one can prove a similar result.

**Theorem 3.1** *The shape derivative for the functional ( 3.1) can be described in terms of the adjoint problem as*

$$J'(\Omega)(\theta) = \int_0^T \int_\Gamma 2\mu \mathrm{e}(u) : \mathrm{e}(v)(\theta \cdot n)\mathrm{d}s\mathrm{d}t,$$

*where* $(v, q)$ *is solution of the following system:.*

$$\begin{aligned}
-\frac{\partial v}{\partial t} - \mu\Delta v + (\nabla u)^T v - (\nabla v)(u - w) + \nabla q &= (u - u_{ref})\mathbb{1}_D && \text{in } \Omega_\theta \times I \\
div(v) &= 0 && \text{in } \Omega_\theta \times I \\
v &= 0 && \text{on } (\Gamma_\theta \cup \Gamma_\theta^{in}) \times I \\
\sigma(v, q)n + (u \cdot n)v &= 0 && \text{on } \Gamma_\theta^{out} \times I \\
v(x, T) &= 0 && \text{in } \Omega_\theta.
\end{aligned} \tag{3.3}$$

**Observation** Notice that the adjoint problem has a final condition instead of an initial one, i.e., reverse in time. Further details can be found in [23].

Proof. To express $J'(\Omega)(\theta)$ in terms of the adjoint problem, one needs to write the Lagrangian as was done in the stationary case. For this, we use the space-time variational formulation for the Navier-Stokes equations,

$$\begin{aligned}
\mathcal{L}(u, p, \theta, v, q) =& \frac{1}{2}\int_0^T \int_D |u(x,t) - u_{ref}(x,t)|^2 \mathrm{d}x\mathrm{d}t \\
&- \int_0^T \int_{\Omega_\theta} \left( \frac{\partial u}{\partial t} v + 2\mu \mathrm{e}(u) : \mathrm{e}(v) + (\nabla u u \cdot v) - p\mathrm{div}(v) - q\mathrm{div}(u) \right) \mathrm{d}x\mathrm{d}t \\
&- \int_{\Omega_\theta} u(x, 0)v(x, 0)\mathrm{d}x, \\
D_u\mathcal{L}(u, p, \theta, v, q)h =& \int_0^T \int_D (u - u_{ref})h\mathrm{d}x\mathrm{d}t \\
&- \int_0^T \int_{\Omega_\theta} \frac{\partial h}{\partial t} v + 2\mu \mathrm{e}(h) : \mathrm{e}(v) - q\mathrm{div}(h) + (\nabla u)^T vh + (\nabla hu)\, v \, \mathrm{d}x\mathrm{d}t = 0 \\
&- \int_{\Omega_\theta} h(x, 0)v(x, 0)\mathrm{d}x, \\
D_p\mathcal{L}(u, p, \theta, v, q)k =& - \int_0^T \int_{\Omega_\theta} k\mathrm{div}(v)\mathrm{d}x\mathrm{d}t = 0.
\end{aligned}$$

Integrating by parts the second equation we get

$$
\begin{aligned}
D_u\mathcal{L}(u,p,\theta,v,q)h = &\int_0^T\int_D (u-u_{ref})h\mathrm{d}x\mathrm{d}t \\
&-\int_0^T\int_{\Omega_\theta} -\frac{\partial v}{\partial t}h - \mu\Delta v h + \nabla q h + (\nabla u)^T v h - (\nabla v u)\, h\,\mathrm{d}x\mathrm{d}t \\
&-\int_0^T\int_{\partial\Omega_\theta} [\sigma(v,q)n + (u\cdot n)v]h \\
&-\left[\int_{\Omega_\theta} v(x,T)h(x,T)\mathrm{d}x - \int_{\Omega_\theta} v(x,0)h(x,0)\mathrm{d}x\right] - \int_{\Omega_\theta} h(x,0)v(x,0)\mathrm{d}x = 0.
\end{aligned}
$$

Then, the adjoint problem derived from this equation is the following:

$$
\begin{aligned}
-\frac{\partial v}{\partial t} - \mu\Delta v + (\nabla u)^T v - (\nabla v)\,(u-w) + \nabla q &= (u-u_{ref})\mathbb{1}_D && \text{en } \Omega_\theta \times I \\
\mathrm{div}(v) &= 0 && \text{en } \Omega_\theta \times I \\
v &= 0 && \text{en } (\Gamma_\theta \cup \Gamma_\theta^{in}) \times I \\
\sigma(v,q)n + (u\cdot n)v &= 0 && \text{en } \Gamma_\theta^{out} \times I \\
v(x,T) &= 0 && \text{en } \Omega_\theta.
\end{aligned}
$$

Now we can write the formulation as finite elements in space and finite differences in time. A monolithic scheme will be used as in [6], and will be solved in the ALE formalism over a reference domain. The formulation is as follows:

$$
\begin{aligned}
&\int_{\Omega_0}\left[J\frac{\partial v}{\partial t}h + J2\mu\mathrm{e}(v) : \mathrm{e}(h) - q\mathrm{div}(JHh) + k\mathrm{div}(JHv) + J(\nabla u H)^T v h + J(\nabla h H)(u-w)v\right]\mathrm{d}x \\
&-\int_D (u-u_{ref})h\mathrm{d}x = 0.
\end{aligned}
$$

Having solved the adjoint problem, we can differentiate the functional $J(\Omega)$ with respect to shape and replace the term $u'$,

$$
J'(\Omega)(\theta) = \int_0^T\int_D (u(x,t) - u_{ref}(x,t))u'(x,t)\mathrm{d}x\mathrm{d}t,
$$

where $u'$ is the solution of the Navier-Stokes equations derived with respect to shape or sensibility problem, i.e.,

$$
\begin{aligned}
\frac{\mathrm{d}u'}{\mathrm{d}t} - \mu\Delta u' + (\nabla u)\, u' + (\nabla u')\, u + \nabla p' &= 0 && \text{in } \Omega \\
\operatorname{div}(u') &= 0 && \text{in } \Omega \\
\sigma(u', p')n &= 0 && \text{on } \Gamma^{out} \\
u' &= 0 && \text{on } \Gamma^{in} \\
u' &= -\left(\tfrac{\partial u}{\partial n}\right)(w \cdot n) && \text{on } \Gamma.
\end{aligned}
\tag{3.4}
$$

Before going further, notice that at this point we can change the Navier-Stokes model by adding the kinematic condition $u = w$ on $\Gamma$, where $w$ is the wall velocity. This, in the Lagrangian framework, means that the fluid velocity on the free boundary matches the wall velocity or deformation velocity. However, because the simulations are made following the Eulerian framework, there is no deformation velocity in the equations. Instead, that information is contained in the domain as we deform it along with the iterations. Therefore, when writing the equations in the Eulerian framework, $w = 0$.

$$
\begin{aligned}
\frac{\partial u}{\partial t} - \mu\Delta u + ((u - w) \cdot \nabla)\, u + \nabla p &= 0 && \text{in } \Omega \\
\operatorname{div}(u) &= 0 && \text{in } \Omega \\
u &= u_{in} && \text{on } \Gamma^{in} \\
u &= w && \text{on } \Gamma \\
\sigma(u, p)n &= 0 && \text{on } \Gamma^{out} \\
u(x, 0) &= u_0 && \text{in } \Omega.
\end{aligned}
\tag{3.5}
$$

This new system gives the same adjoint problem ( 3.3) we got in Theorem 3.1 so there is no need to recalculate. And for the solution $(u', p')$, the sensibility problem is the same as ( 3.4) except for the kinematic condition, which can be calculated independently. Let $\Phi(x, t)$ be a function that returns the new position of a point in the reference mesh at a certain time. This function is defined as

$$\Phi(x, t) = x + \theta(x, t) = x + tw(x).$$

Then we do some computations similar to what is done for the Laplace equation in [15],

$$u_t(\Phi(x, t)) = \frac{\mathrm{d}\Phi(x, t)}{\mathrm{d}t} = \frac{\mathrm{d}\theta(x, t)}{\mathrm{d}t} = w(x), \quad \forall x \in \Gamma_{ref},$$

$$u' + \nabla u \cdot \frac{\mathrm{d}\Phi(x, 0)}{\mathrm{d}t} = 0, \quad \forall x \in \Gamma_{ref},$$

i.e.,

$$u' + \nabla u \cdot w = 0, \quad \forall x \in \Gamma_{ref}.$$

Now that we got both the adjoint problem and the sensibility problem for the new system considering the kinematic condition, we calculate the shape derivative replacing $u'$ with the adjoint solution in the same way done for the stationary case.

$$J'(\Omega)(\theta) = \int_0^T \int_D (u - u_{ref})u' \mathrm{d}x,$$

$$J'(\Omega)(\theta) = \int_0^T \int_\Omega \left[ -\frac{\partial v}{\partial t} - \mu\Delta v + (\nabla u)^T v - (\nabla v)\, u + \nabla q \right] u' \mathrm{d}x\mathrm{d}t.$$

The first equation in system ( 3.4) says that

$$\int_\Omega \frac{\partial u'}{\partial t} - \mu\Delta u' + (\nabla u)u' + (\nabla u')u + \nabla p' \mathrm{d}x = 0,$$

i.e., multiplying by $\mu$ and integrating by parts we obtain

$$\begin{aligned} \int_0^T \int_\Omega \frac{\partial u'}{\partial t} v + 2\mu \mathrm{e}(u') : \mathrm{e}(v) + (\nabla u)u'v + (\nabla u')uv - p' \mathrm{div}(v) \mathrm{d}x\mathrm{d}t \\ - \int_0^T \int_{\partial\Omega} \sigma(u', p')n \cdot v \mathrm{d}s\mathrm{d}t = 0. \end{aligned} \tag{3.6}$$

Then, integrating by parts and subtracting the variational formulation ( 3.6) we get

$$\begin{aligned} J'(\Omega)(\theta) = & \int_0^T \int_\Omega \left[ -\frac{\partial v}{\partial t} - \mu\Delta v + (\nabla u)^T v - (\nabla v)\, u + \nabla q \right] u' \mathrm{d}x\mathrm{d}t \\ = & \int_0^T \int_\Omega -\frac{\partial v}{\partial t} u' + 2\mu \mathrm{e}(v) : \mathrm{e}(u') + (\nabla u)^T v u' - (\nabla v)\, uu' \mathrm{d}x\mathrm{d}t - \int_0^T \int_\Omega \sigma(v, q)n \cdot u' \mathrm{d}s\mathrm{d}t \\ = & \int_0^T \int_\Omega \left[ -(\nabla v)uu' - (\nabla u')uv \mathrm{d}x - \int_{\partial\Omega} \sigma(v, q)nu' \mathrm{d}s + \int_{\partial\Omega} \sigma(u', p')n \cdot v \mathrm{d}s \right] \mathrm{d}t \\ & + \int_\Omega v(x, T)u'(x, T) \mathrm{d}x - \int_\Omega v(x, 0)u'(x, 0) \mathrm{d}x. \end{aligned}$$

Using the initial and final conditions $v(x, T) = 0$, $u'(x, 0) = 0$ and the boundary conditions, we get the following expression for $J'$:

$$J'(\Omega)(\theta) = \int_0^T \int_\Gamma 2\mu \mathrm{e}(u) : \mathrm{e}(v)(\theta \cdot n) \mathrm{d}s\mathrm{d}t.$$

□

Although this method is the direct translation of the algorithm in Chapter 2, one can only see the solution at the end, as all time steps are being solved simultaneously. Moreover, this method does not guarantee a good approximation in each time step individually, as it solves the integral over the time interval. For these reasons, we propose an alternative method that pretends to solve these problems.

## 3.2 Quasi-stationary method

This method consists of solving a stationary problem on each time step, adding to the equations an inertial force product of the fluid's movement. This way of solving iteratively in time allows one to see partial results as they are being generated, and to minimize the error on each individual time step instead of the aggregate.

Let $(u, p) := (u^n, p^n)$ and $\Omega := \Omega^n$ be the solution of the following system:

$$\begin{aligned} -\mu\Delta u + ((u - w^n)\cdot\nabla)\,u + \nabla p &= -\tfrac{u-u_{ref}^{n-1}}{\tau} && \text{in } \Omega \\ \operatorname{div}(u) &= 0 && \text{in } \Omega \\ u &= u_{in} && \text{on } \Gamma^{in} \\ u &= w^n && \text{on } \Gamma \\ \sigma(u,p)n &= 0 && \text{on } \Gamma^{out}. \end{aligned} \tag{3.7}$$

The term $-\frac{u-u_{ref}^{n-1}}{\tau}$ is a finite difference approximation of the acceleration produced by the fluid's movement, and because it was considered the normalized equations ($\rho = 1$), this term represents the inertial force of the fluid. Solving each time step individually implies the objective function must depend on time, thus the need to solve for the minimization problem involving the functional defined in the stationary case now dependent on a time step.

$$J^n(\Omega) = \frac{1}{2}\int_D |u(\Omega)^n - u_{ref}^n|^2 \mathrm{d}x.$$

Then the following theorem holds:

**Theorem 3.2** *Given $(u, p)$ solution of ( 3.7) and $(v, q)$ solution of ( 3.8), the shape derivative of $J^n(\Omega)$ can be written as*

$$(J^n)'(\Omega)(\theta) = \int_\Gamma 2\mu \mathrm{e}(u^n) : \mathrm{e}(v^n)(\theta\cdot n)\mathrm{d}s.$$

Proof. Since we are using the stationary Navier-Stokes equations, both the adjoint and sensibility problems are the same we previously calculated, i.e., $(v, q) := (v^n, q^n)$ and $(u', p') := (u'^n, p'^n)$ solutions of the following systems respectively:

$$
\begin{aligned}
-\mu\Delta v + (\nabla u)^T v - (\nabla v)\,(u^n - w^n) + \nabla q &= (u^n - u_{ref})\mathbb{1}_D && \text{in } \Omega \\
\operatorname{div}(v) &= 0 && \text{in } \Omega \\
v &= 0 && \text{on } \Gamma \cup \Gamma^{in} \\
\sigma(v,q)n + (u^n \cdot n)v &= 0 && \text{on } \Gamma^{out},
\end{aligned}
\tag{3.8}
$$

$$
\begin{aligned}
-\mu\Delta u' + (\nabla u)\,u' + (\nabla u')\,u + \nabla p' &= 0 && \text{in } \Omega \\
\operatorname{div}(u') &= 0 && \text{in } \Omega \\
\sigma(u',p')n &= 0 && \text{on } \Gamma^{out} \\
u' &= 0 && \text{on } \Gamma^{in} \\
u' &= -\left(\tfrac{\partial u}{\partial n}\right)(w^n \cdot n) && \text{on } \Gamma.
\end{aligned}
\tag{3.9}
$$

Then, the directional derivative of the functional can be written in terms of the adjoint solution as

$$
(J^n)'(\Omega)(\theta) = \int_D (u - u_{ref})u' \mathrm{d}x = \int_\Gamma 2\mu \mathrm{e}(u^n) : \mathrm{e}(v^n)(\theta \cdot n)\mathrm{d}s.
$$

□

# Chapter 4

# Numerical Results

This chapter will present the numerical simulations done for the two cases described in Chapter 2, where the objective function considers a subdomain $D$ containing information of the reference velocity,

$$J(\Omega) = \frac{1}{2}\int_D |u - u_{ref}|^2 \mathrm{d}x,$$

and the quasi-stationary simulations from the Chapter 3 method, where the functional depends on the time iteration and the Navier-Stokes problem considers an inertial force produced by the fluid's movement.

$$J^n(\Omega) = \frac{1}{2}\int_D |u(\Omega)^n - u^n_{ref}|^2 \mathrm{d}x.$$

The reference domain is modeled as a rectangular mesh, which will be modified on each iteration, thus solving the systems in the Eulerian framework. This allows in the dynamic case to simplify the equations because of the term of domain velocity when considering a fixed domain. On the other hand, this means one needs to modify the mesh mid-execution, which implies re-meshing regularly to ensure the quality of the solution.

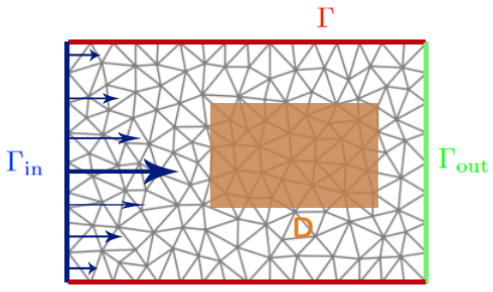


Figure 4.1: Reference mesh.

## 4.1 Extension to subdomain measurements

The first example being implemented is the one proposed in [27] as a way to test the algorithm considering subdomain measurements. This simulation aims to validate the method and compare against the original, for this reason, the mesh used in this case for the reference domain and the initial condition for the algorithm is the rectangle $[0, 1.5] \times [0, 1]$, with a mesh made of 1127 cells by the FEniCS library. For this example, consider a parabolic inlet

$$u_{\text{in}}(x, y) = (y(1 - y), 0), \tag{4.1}$$

imposed on $\Gamma^{in}$ boundary as shown in figure 4.1. Since the algorithm from Chapter 2 requires data inside the domain, one needs to solve the Navier-Stokes equations on the deformed domain to get the velocity data. Thus, it was necessary to recreate the original method in [27] and solve it using the following reference velocity in $\Gamma^{out}$:

$$u_{ref}(x, y) = (2y^2(1 - y), 0). \tag{4.2}$$

This gives as result the "optimal" deformation, and solving the direct Navier-Stokes problem ( 1.1) in the deformed mesh results in the velocity inside the domain. This velocity data is the one to be used in the algorithm proposed in Chapter 2.

Figure ( 4.2) shows the initial and final deformation using the method described in Chapter 2 without considering volume or perimeter penalization. It takes 38 BFGS iterations in approximately 47.8 minutes to reach a solution close enough to the optimal shape.

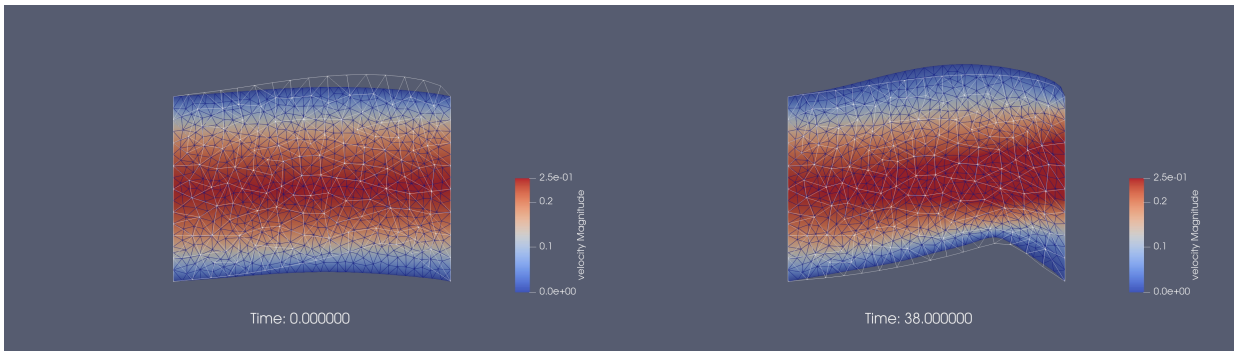


Figure 4.2: Deformation given by Algorithm 2 on iterations t=0 y t=38, initial and final states respectivly on the example proposed in [27].

Figure ( 4.3) shows the simulation results for the same example considering penalization terms for the volume and perimeter. This result shows a better approximation to the optimal shape, taking 38 iterations and approximately 33.8 minutes.

For the second example, considering a symmetrical shape motivated by the deformation of a vessel at the time of a pulse. Considering both the volume and perimeter penalization terms, Figure ( 4.4) shows the obtained results. For this case, it was defined an imposed deformation, the exponential function

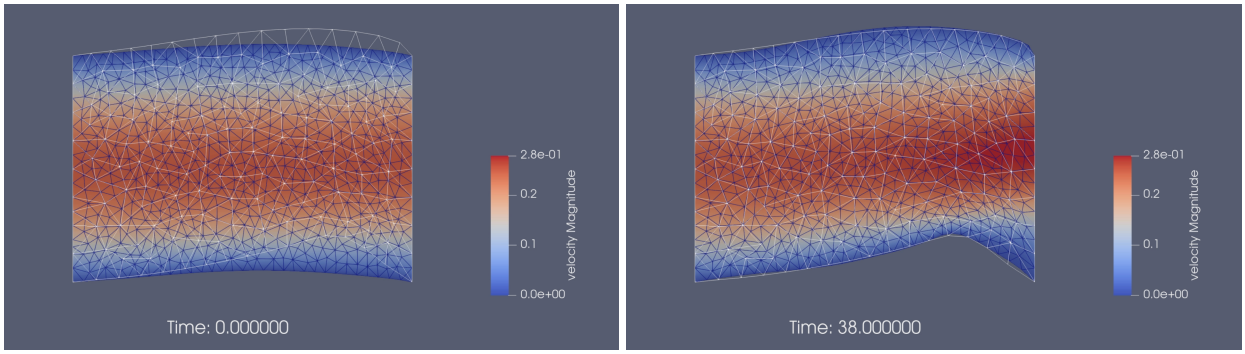


Figure 4.3: Deformation given by Algorithm 2 on iterations t=0 y t=38, initial and final states respectivly on the example proposed in [27].

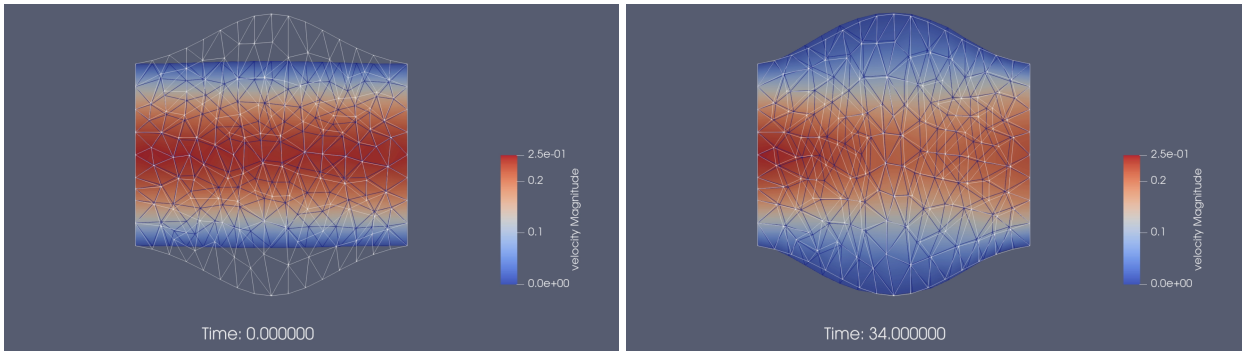


Figure 4.4: Stationary case, symmetrical example.

$$9(y-\tfrac{1}{2})\left(\tfrac{1}{\mathrm{d}}\mathrm{e}^{\frac{-(x-3/4)^2}{2\sigma^2}}-m\right), \tag{4.3}$$

where $\mathrm{d}=6\sqrt{2\pi}$, $\sigma=1/3$ and $m=\frac{1}{\mathrm{d}}\mathrm{e}^{\frac{-(3/4)^2}{2\sigma^2}}$. This function over the $\Gamma$ boundary was design to represent a blood vessel expanding by the blood movement. This function also allows to easily extend to the time dependant example by moving on the X axis by the following deformation:

$$\mathrm{Sym}(x,y)=\begin{bmatrix}0\\ 9(y-\frac{1}{2})\left(\frac{1}{\mathrm{d}}\mathrm{e}^{\frac{-(x-3/4)^2}{2\sigma^2}}-m\right)\end{bmatrix}. \tag{4.4}$$

## 4.2 Extension to a mobile wall

First, we need to redefine the domain and mesh to work with for this extension, as the previous one is too small in length to allow the peak to move significantly. The new domain is the rectangle $[0,6]\times[0,1]$ with a structured mesh as we want to minimize the computations and a structured mesh can be stored in less memory and gives a more symmetrical solution (as should happen according to the shape gradient).

For this extension, velocity data over time is needed. This means solving the non-stationary Navier-Stokes equations over a domain dependent on time. The deformation follows the exponential

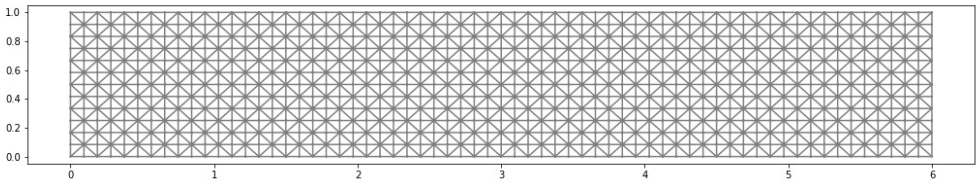

Figure 4.5: Reference mesh

equation

$$\mathrm{Def}(x,y) = \begin{bmatrix} 0 \\ 6(y-\frac{1}{2})\left(\frac{1}{\mathrm{d}}\mathrm{e}^{\frac{-(x-3/2-\ell)^2}{2\sigma^2}} - m\right) \end{bmatrix}, \tag{4.5}$$

where $\ell$ is the displacement of this function. For the purpose of these simulations, let us consider $\ell \in [0,3]$, and the following growth to avoid discontinuities in the velocity,

$$\ell(t) = \frac{1}{t^*}\frac{V_{max}}{2}t^2, \tag{4.6}$$

with $V_{max} = 5$, $t^* = \frac{2(V_{max}-3)}{V_{max}}$. $\ell(t)$ being a function that remains constant from a certain time $t^*$ onwards, initial velocity zero and quadratic acceleration before $t = t^*$ as shown in figure ( 4.6)

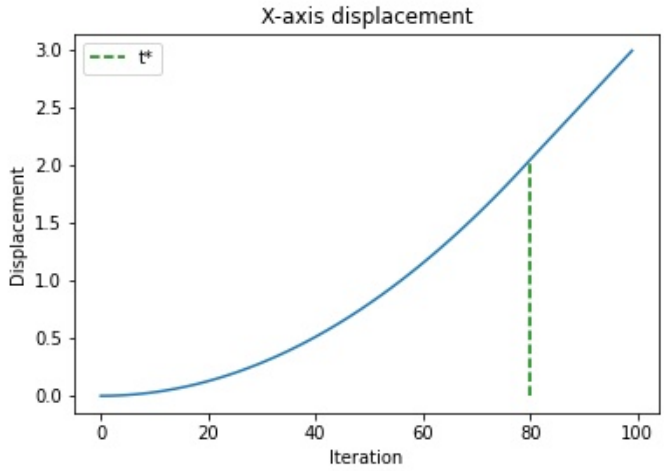


Figure 4.6: Peak displacement over time, quadratic and linear regime.

For the purpose of ensuring that the velocity does not change abruptly, the Navier-Stokes equations were solved over an interval $[0.3]$ consisting of two segments, the first one do not move the domain, thus reaching a steady flow. Then we apply the displacement over the time interval $[2,3]$. This interval is discretized with a step $\tau = 0.1$ (this is why figure ( 4.6) was made with 100 time steps). Solving the non-stationary Navier-Stokes equations from system ( 1.2) gives as a result the velocity in the domain interior through the time interval $[0,3]$. Since the segment of interest is the one in which the deformation moves, let us rename this segment as the time interval $[0,1]$ and forget about the previous segment onwards.

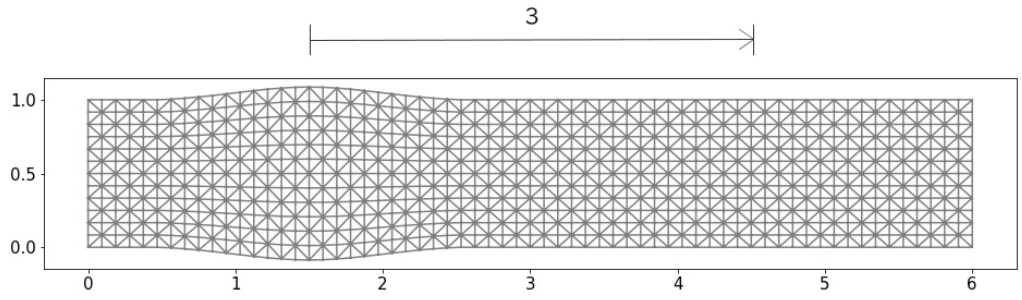


Figure 4.7: Reference for mobile domain.

Using this velocity data as reference, the deformation identification algorithm described in Chapter 3 was used to reconstruct the shape of the domain. In order to measure the velocity of the reconstructed domain, the last 50 iterations of the reference velocity and deformation were used, this gives a deformation that moves at a more easily comparable speed early in the iteration process, and therefore allowed to make corrections to the parameters earlier.

Unfortunately, the results did not come out as expected, as the deformation obtained kept increasing with each iteration. For this reason, to try to control the deformation, three nodes were fixed to zero deformation close to the inlet and outlet boundaries on following examples. In reality, there was an error in the visualization, which accumulated the deformations making it seem like an over increment in the volume, but not knowing that at the moment, it was believed that the fact that the penalization parameters are the same for every time step could have been the reason why the deformation over increased, meaning that they were not appropriate for every time step and needed adjusting between iterations, making them dynamic to the changes in the deformation. Later on, the error was corrected for the dynamic parameters case and also for the fixed parameters case.

This dynamic parameter technique is accomplished by defining the penalization parameters to follow the same proportion that worked for the previous case where we had the same shape but stationary. It is designed to not overshadow the main term of the objective function, as it keeps the same proportion for every time step. Figure ( 4.8) shows the peak displacement obtained compared to the reference movement for the algorithm incorporating the dynamic parameters. The results were able to capture the movement of the peak through the x-axis, although there are some errors at the beginning which can be attributed to the algorithm solving the problem in each time step independently except for the inertial force considered in the fluid motion equations.

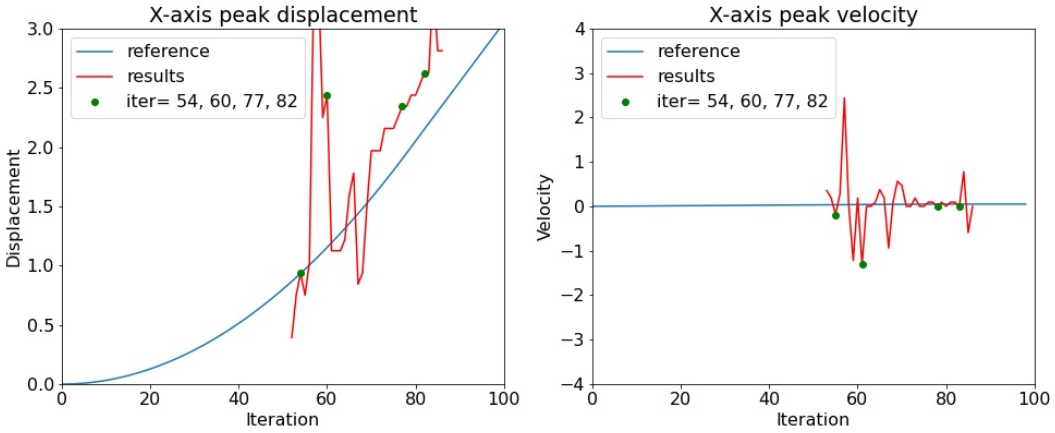


Figure 4.8: Peak displacement and velocity with dynamic parameters, indicating key times iter=54, iter=60, iter=77, iter=82 (corresponding to times 2, 8, 25, 30).

The velocity graph in figure( 4.8) confirms that the solution presents oscillations at first and then stabilizes. Although, at the end there are still some errors product of the instabilities. Figure ( 4.9) shows the deformation at key times, which allows seeing that it starts on the right foot and at some times (time=8 for example) the displacement is off because there is no visible deformation and the peak or maximum even escapes from the range it should be. At last, times 25 and 30 shows the peak moving ahead of the reference as can be seen in the displacement graph in figure ( 4.8)

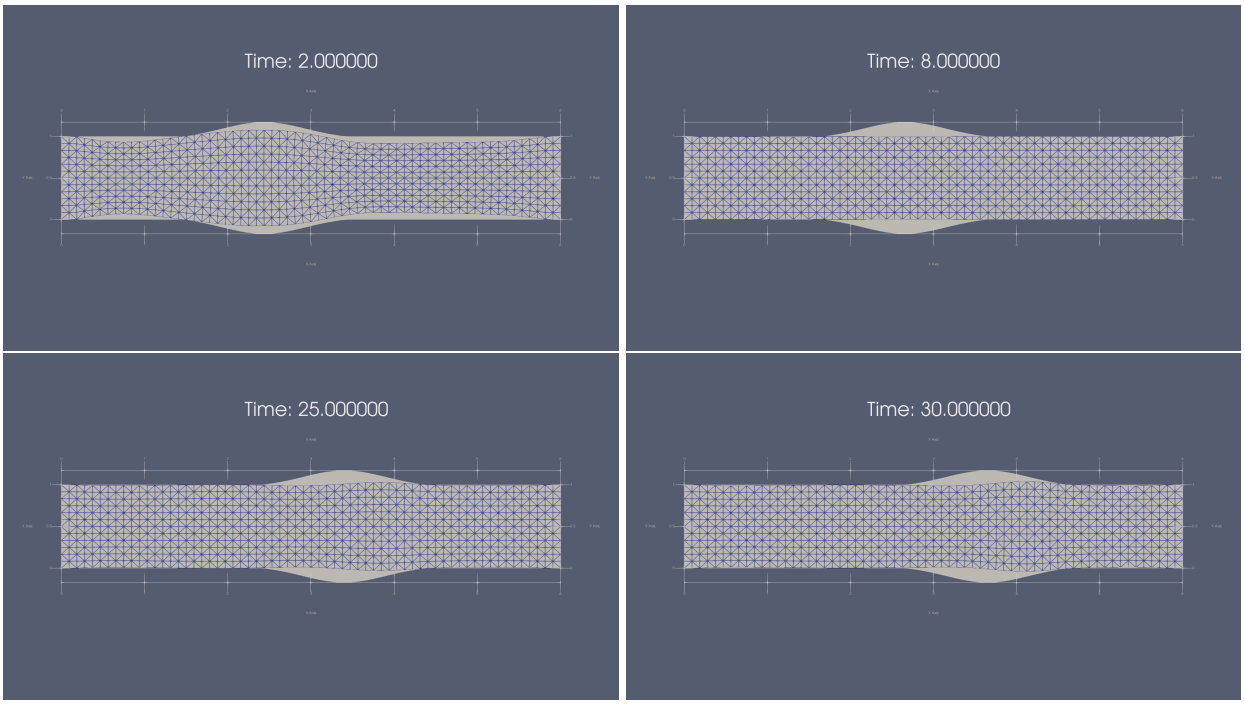


Figure 4.9: Comparison between the results and the reference at key times.

Lastly, the fixed parameters case was tested again as explained before correcting the error, which gave as result the displacement and velocity graph in figure ( 4.10). The displacement at the beginning presents a similar problem to the result obtained in the dynamic parameters case, and

a similar stabilization at the end. One can notice that the final segment stabilizes, although the mean velocity seems to correspond with the reference velocity, the results are shifted ahead of the reference. This phenomenon is yet to have an answer, but a possible explanation could be the reference velocity increasing in magnitude as the peak moves.

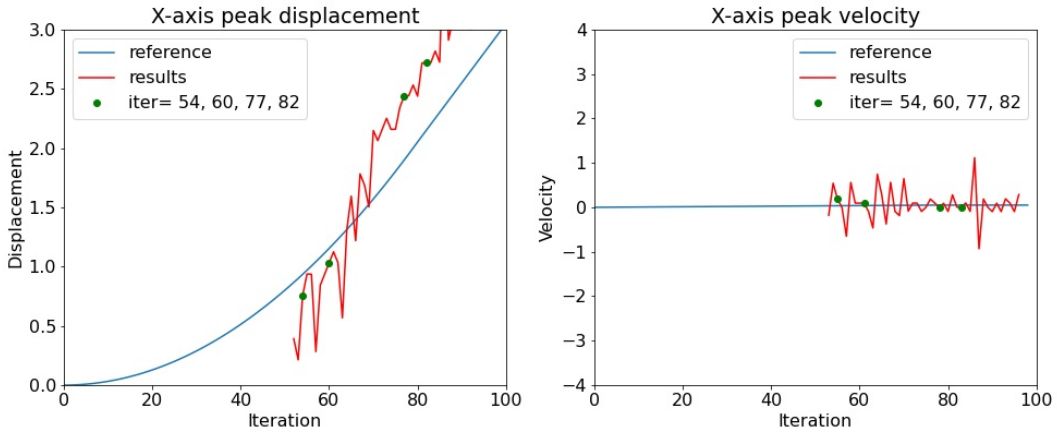


Figure 4.10: Peak displacement and velocity with fixed parameters, indicating key times iter=54, iter=60, iter=77, iter=82 (corresponding to times 2, 8, 25, 30).

# Conclusion and discussion

The main objectives of this work were partially accomplished, since it was possible to make an analysis to give an easily computable expression to the shape gradient for a wall identification problem based on velocity measurements for both the stationary and non-stationary cases, and an algorithm capable of reconstructing the boundary shape for the stationary case and partially capable of identifying the movement of a mobile wall. However, the results obtained show oscillations and instabilities for a problem that does not consider noise in the measurements, which could be solved imposing known or estimated information into the problem.

This work contributed in the first place to the analysis of the shape gradient for shape identification problems based on velocity data inside the domain over any subset. This allows one to be flexible, and work with more information when available or to restrict the subdomain to where the velocity is known. Secondly, the algorithm implemented on Python based on the methodology proposed in [27] applied to the subdomain measurements extension. This implementation is open source and extensible, since the PDE solvers are modular.

In the third place, the extension to a time dependent problem considering a mobile wall with an imposed deformation. This extension reduces the time dependent problem to the stationary case defining a new model for the fluid considering the movement as an inertial force, which makes it possible to obtain partial results mid-iteration.

Finally, the algorithm implementation for the non-stationary case, which makes use of a modified version of the BFGS algorithm and a methodology of self-adjusting parameters throughout the iteration process.

This work has contributed to the study of non-invasive deformation estimation of blood vessels, but the analysis made can be extended to more complex identification problems for mobile walls like the three-dimensional case or considering coupling with an elastic wall, which could be done considering a fluid-structure formulation instead of the Navier-Stokes system. The methodology and algorithms made in this study could also be extended or modified to be applied to other models where the shape is the unknown and one has information on the interior of the domain.

# Bibliography


[1] Jorge Aguayo, Cristóbal Bertoglio, and Axel Osses. A distributed resistance inverse method for flow obstacle identification from internal velocity measurements. *Inverse problems*, 37(2), January 2021.

[2] Jorge Aguayo and Hugo Carrillo. Analysis of obstacles immersed in viscous fluids using brinkman's law for steady stokes and navier-stokes equations. *ArXiv*, abs/2012.08635, 2020.

[3] Jorge Aguayo and Axel Osses. A stability result for the identification of a permeability parameter on navier–stokes equations. *Inverse Problems*, 38(7):075001, may 2022.

[4] Grégoire Allaire. *Conception Optimale de Structures*. 01 2007.

[5] Philippe Angot, Charles-Henri Bruneau, and Pierre Fabrie. A penalization method to take into account obstacles in viscous flows. *Numerische Mathematik*, 81:497–520, 02 1999.

[6] Reidmen Aróstica and Cristóbal Bertoglio. On monolithic and chorin-temam schemes for incompressible flows in moving domains. *Applied Mathematics Letters*, 112:106830, 02 2021.

[7] Cristóbal Bertoglio, Rodolfo Núñez, Felipe Galarce, David Nordsletten, and Axel Osses. Relative pressure estimation from velocity measurements in blood flows: State-of-the-art and new approaches. *International journal for numerical methods in biomedical engineering*, 34, 09 2017.

[8] Hugo Carrillo Lincopi, Axel Osses, Sergio Uribe, and Cristóbal Bertoglio. Optimal dual-venc (odv) unwrapping in phase-contrast mri. *IEEE Transactions on Medical Imaging*, PP:1–1, 11 2018.

[9] Instituto Nacional de Estadística de Chile (INE). Estadísticas vitales, informe anual, 2010-2019.

[10] Simone Deparis. Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation. 01 2004.

[11] Jean Donea and Antonio Huerta. Finite element methods for flow problems. 2003.

[12] Alexandre Ern and Jean-Luc Guermond. Theory and practice of finite elements. 2004.

[13] Vivette Girault and P. A. Raviart. Finite element approximation of the navier-stokes equations. 1979.

[14] Vivette Girault and P. A. Raviart. Finite element methods for navier-stokes equations - theory and algorithms. In *Springer Series in Computational Mathematics*, 1986.

[15] Antoine Henrot and Michel Pierre. Shape variation and optimization: A geometrical analysis. Zürich, Switzerland: European Mathematical Society Publishing House, 2018.

[16] Helge Herthum, Hugo Carrillo, Axel Osses, Sergio Uribe, Ingolf Sack, and Cristobal Bertoglio. Optimal multiple motion encoding in Phase-Contrast MRI. December 2020.

[17] Olga A. Ladyzhenskaya, Richard A. Silverman, Jacob T. Schwartz, and Jacques E. Romain. The mathematical theory of viscous incompressible flow. 1972.

[18] Anders Logg. *Automated solution of differential equations by the finite element method : the FEniCS book*. Springer, Berlin New York, 2012.

[19] B. Mohammadi and O. Pironneau. *Applied Shape Optimization for Fluids*. Oxford University Press, 2001.

[20] Bijan Mohammadi and Olivier Pironneau. Shape optimization in fluid mechanics. *Annual Review of Fluid Mechanics*, 36(1):255–279, 2004.

[21] David Nolte, Jesús Urbina, Julio Sotelo, Leo Sok, Cristian Montalba, Israel Valverde, Axel Osses, Sergio Uribe, and Cristóbal Bertoglio. Validation of 4d flow based relative pressure maps in aortic flows. *Medical Image Analysis*, 74:102195, 08 2021.

[22] Stanley Osher and R. Fedkiw. *The Level Set Methods and Dynamic Implicit Surfaces*, volume 57, pages xiv+273. 05 2004.

[23] Thomas Richter. *Fluid-structure Interactions*. Springer Cham, 2017.

[24] Jan Sokolowski and Antoni Zochowski. On Topological Derivative in Shape Optimization. Research Report RR-3170, INRIA, 1997.

[25] Roger Temam. Navier-stokes equations: Theory and numerical analysis. 1979.

[26] Michael Markl y Alex Frydrychowicz y Sebastian Kozerke y Mike Hope y Oliver Wieben. 4d flow mri. *JMRI*, 36(5):1015–1036, octubre 2012.

[27] Dapogny C. y Frey P. y Omnès F. et al. Geometrical shape optimization in fluid mechanics using freefem++. *Struct Multidisc Optim*, 58(6):2761–2788, Diciembre 2018.