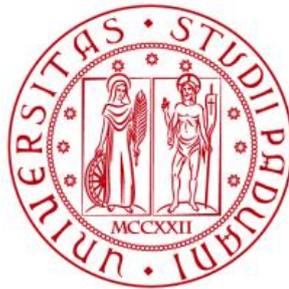


# UNIVERSITY OF PADOVA

DIPARTIMENTO DI INGEGNERIA CIVILE, EDILE E AMBIENTALE

*Department Of Civil, Environmental and Architectural Engineering*

Master's degree in Mathematical Engineering



UNIVERSITÀ  
DEGLI STUDI  
DI PADOVA

MASTER'S THESIS

**3D Numerical Model of Navier-Stokes Flows to solve  
Topology Optimization by including scalar Transport  
Problems**

SUPERVISOR:

Prof. PAOLO PERUZZO

CANDIDATE:

ANDREA GORGI

2010658

CO-SUPERVISOR:

Char.mo Prof. STEFANO LANZONI

Academic year 2022/2023



# Contents

|          |  |           |
|----------|--|-----------|
| <b>1</b> | <b>Introduction</b>  | <b>19</b> |
| 1.1      | The Topology Optimization, an overview . . . . .             | 20        |
| 1.2      | Motivation and project guidelines . . . . .                  | 21        |
| 1.3      | State of the Art . . . . .                                   | 25        |
| <b>2</b> | <b>Numerical models: background</b>                          | <b>29</b> |
| 2.1      | Topology Optimization: general description . . . . .         | 29        |
| 2.2      | Topology Optimization for fluid flow . . . . .               | 32        |
| 2.2.1    | Objective functionals . . . . .                              | 35        |
| 2.2.2    | Constraints . . . . .  | 38        |
| 2.3      | The Topology optimization for Transport of Species . . . . . | 38        |
| 2.3.1    | Objective functionals and constraints . . . . .              | 40        |
| 2.4      | FEM formulation . . . . .                                    | 42        |
| 2.4.1    | Navier-Stokes . . . . .                                      | 42        |

---

|          |   |           |
|----------|---|-----------|
| 2.4.2    | Advection-Diffusion-Reaction . . . . .                                      | 46        |
| 2.5      | Solver . . . . .  | 50        |
| 2.5.1    | Navier-Stokes . . . . .   | 50        |
| 2.5.2    | Advection-Diffusion-Reaction . . . . .                                      | 57        |
| <b>3</b> | <b>Materials and Methods</b>  | <b>59</b> |
| 3.1      | Adjoint derivation . . . . .  | 59        |
| 3.1.1    | The Adjoint derivation for the Navier-Stokes equations                      | 59        |
| 3.1.2    | The Adjoint system for the Advection-Diffusion-Reaction<br>module . . . . . | 74        |
| 3.2      | The Optimization schemes . . . . .  | 78        |
| 3.2.1    | The Optimality Criteria (OC) . . . . .                                      | 78        |
| 3.2.2    | The Generalized Optimality Criteria (GOC) . . . . .                         | 80        |
| 3.2.3    | The Method of Moving Asymptotes (MMA) . . . . .                             | 83        |
| 3.3      | SIMPLE preconditioning and shared-memory multiprocessing                    | 88        |
| 3.3.1    | The SIMPLE scheme . . . . .   | 88        |
| 3.3.2    | The flexible inner preconditioned GMRES . . . . .                           | 89        |

|          |   |            |
|----------|---|------------|
| 3.3.3    | The Shared memory multiprocessing . . . . .                     | 96         |
| 3.4      | The Topology Optimization: the code structure . . . . .         | 97         |
| 3.4.1    | The Topology Optimization algorithm for fluid flow . . . . .    | 97         |
| 3.4.2    | The TO for species transport . . . . .                          | 100        |
| 3.5      | Validation procedure and testing for the TO algorithm . . . . . | 101        |
| 3.5.1    | TO for fluid flow . . . . .                                     | 101        |
| 3.5.2    | TO for species transport . . . . .                              | 111        |
| <b>4</b> | <b>Results and Discussion</b>                                   | <b>115</b> |
| 4.1      | TO for fluid flows . . . . .                                    | 115        |
| 4.1.1    | 2D testing . . . . .  | 115        |
| 4.1.2    | 3D validation . . . . .   | 131        |
| 4.1.3    | Some final considerations: Checkboard problem . . . . .         | 135        |
| 4.1.4    | Optimizer schemes comparison . . . . .                          | 136        |
| 4.2      | TO for fluid flow and scalar transport problems . . . . .       | 137        |
| 4.2.1    | Testing: Bend Channel . . . . .                                 | 137        |
| 4.2.2    | Instabilities and checkboard effect . . . . .                   | 151        |

|          |  |            |
|----------|--|------------|
| 4.2.3    | Optimization schemes: final considerations . . . . . | 153        |
| <b>5</b> | <b>Conclusions and future developments</b>           | <b>157</b> |
| <b>6</b> | <b>Appendices</b>                                    | <b>161</b> |
| 6.1      | Appendix A: Boundary conditions . . . . .            | 161        |
| 6.2      | Appendix B: the elements choice . . . . .            | 167        |
| 6.3      | Appendix C: Solver efficiency . . . . .              | 169        |
| 6.4      | Appendix D: Validation . . . . .                     | 172        |
| 6.4.1    | Advection Diffusion Reaction . . . . .               | 172        |
| 6.4.2    | Navier-Stokes . . . . .                              | 175        |

# List of Figures

|     |   |     |
|-----|---|-----|
| 1.1 | Size optimization, shape optimization and topology optimization of a cantilever with minimizing compliance . . . . .  | 21  |
| 1.2 | <i>Number of papers on TO problems published per year and total accumulated publications over time.</i> [8] . . . . .   | 24  |
| 1.3 | Distribution of papers in overall problem type (left). Distribution of papers for fluid model type (right). [8] . . . . .   | 25  |
| 2.1 | $\mathcal{P}_1 - iso \mathcal{P}_2 / \mathcal{P}_1$ stable elements. Pressure $\mathcal{T}_h^P$ (left) and velocity $\mathcal{T}_h^V$ (right) discretization. . . . . | 45  |
| 3.1 | TO algorithm flowchart . . . . .  | 98  |
| 3.2 | Navier-Stokes solver scheme . . . . .   | 99  |
| 3.3 | TO algorithm flowchart . . . . .  | 100 |
| 3.4 | The "Double pipe" geometry, by [32]. Grey area represents the analyzed domain due to symmetry. . . . .  | 102 |
| 3.5 | Unstable double pipe geometry, by [16]. . . . .   | 103 |
| 3.6 | "4 terminal device" geometry, by [16]. . . . .  | 104 |

|      |   |     |
|------|---|-----|
| 3.7  | "Bend channel" geometry, by [16]. . . . .   | 105 |
| 3.8  | Three-dimensional "bend channel" geometry, by [43]. . . . .   | 107 |
| 3.9  | "6 terminal device" geometry, by [43]. . . . .  | 108 |
| 3.10 | "bend channel" geometry. . . . .  | 112 |
| 4.1  | Optimal layouts for $L = 1.0\text{m}$ (left) and $L = 1.5\text{m}$ (right).<br>From Okubo Jr. [32]. . . . .   | 115 |
| 4.2  | Snapshots of optimization procedure for the steady double<br>pipe example in Figure (3.4) for $L = 1\text{m}$ . From the left, con-<br>figurations at iteration 2, 5, and final result . . . . .                | 117 |
| 4.3  | Snapshots of flow field at the optimization procedure for the<br>steady double pipe example in Figure (3.4) for $L = 1\text{m}$ . From<br>the left, configurations at iteration 2, 5, and final result. . . . . | 117 |
| 4.4  | Performance comparison of the optimizer schemes. . . . .  | 118 |
| 4.5  | Initial material configuration. . . . .   | 119 |
| 4.6  | Snapshots of optimization procedure for the steady double<br>pipe example in Figure (3.4) for initial condition of Figure(4.5).<br>From the left, configurations at iteration 2, 5, and final result. . . . .   | 119 |
| 4.7  | Snapshots of optimization procedure for the steady double<br>pipe example in Figure (3.4) for $L = 1.5\text{m}$ . From the left,<br>configurations at iteration 5 and final result. . . . .                     | 120 |

|      |   |     |
|------|---|-----|
| 4.8  | Snapshots of flow field at the optimization procedure for the steady double pipe example in Figure (3.4) for $L = 1.5\text{m}$ . From the left, configurations at iteration 5 and final result. . . . | 120 |
| 4.9  | Performance comparison of the optimizer schemes. . . . .  | 121 |
| 4.10 | Optimal layout, from [32]. . . . .  | 122 |
| 4.11 | Snapshots of flow field the initial configuration for the unsteady double pipe example in Figure (6.14). . . . .  | 123 |
| 4.12 | Snapshots of optimization procedure for the unsteady double pipe example in Figure (6.14). From the left, configurations at iteration 2, 5 and final result. . . . .                                  | 123 |
| 4.13 | Performance comparison of the optimizer schemes. . . . .  | 124 |
| 4.14 | Snapshots of flow field at the final MMA configuration of Figure(4.12) for the unsteady double pipe example in Figure (6.14).124  |     |
| 4.15 | Optimal layout for the 4 terminal device [16]. . . . .  | 125 |
| 4.16 | Snapshots of optimization procedure for the 4 device example in Figure (3.6). From the left, configurations at iteration 5 and final result. . . . .  | 126 |
| 4.17 | Snapshots of flow field at the optimization procedure for the 4 device example in Figure (3.6). From the left, configurations at iteration 5 and final result. . . . .                                | 126 |
| 4.18 | Performance comparison of the optimizer schemes. . . . .  | 127 |

|   |     |
|---|-----|
| 4.19 Optimal design at different Reynolds numbers [16], with an objective . . . . .   | 128 |
| 4.20 Optimal layouts for the bend channel example in Figure (3.7) at $Re = 1$ . From the left, configurations at iteration 5 and final result. . . . .  | 128 |
| 4.21 Optimal layouts for the bend channel example in Figure (3.7) at $Re = 300$ . . . . .   | 129 |
| 4.22 Optimal layout, from [45]. . . . .   | 130 |
| 4.23 Optimal layouts for the 4 terminal device under gravity (4.22). . . . .  | 130 |
| 4.24 Optimal layout of the three-dimensional bend channel [43]. . . . .   | 131 |
| 4.25 Snapshots of optimization procedure for the three-dimensional bend channel device example in Figure (3.8) at $Re = 1$ . From the left, configurations at iteration 2, 5, and final result. . . . . | 132 |
| 4.26 Snapshots of optimization procedure for the 6 device example in Figure (3.8) at $Re = 50$ . From the left, configurations at iteration 2, 5, and final result. . . . .                             | 132 |
| 4.27 Optimal layout of the 6 terminal device [43]. . . . .  | 133 |
| 4.28 Snapshots of optimization procedure for the 6 device example in Figure (3.9). From the left, configurations at iteration 2, 5, and final result. . . . .   | 134 |

- 4.29 Snapshots of flow field at the optimization procedure for the 6 device example in Figure (3.9). From the left, configurations at iteration 2, 5, and final result. . . . . 134
- 4.30 Resulting optimal configurations for the 6 device example in Figure (3.9). . . . . 135
- 4.31 Performance comparison of the optimizer schemes. . . . . 135
- 4.32 Examples of checkboard patterns in the developed examples. . 136
- 4.34 Snapshots of optimization procedure and effects on the flow field and pollute distribution for the bend channel example in Figure (3.10) for  $\xi = 0$ . Left column, Iteration 2, right column, final iteration. . . . . 138
- 4.33 Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration for  $\xi = 0$  . . . . . 139
- 4.35 Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration for  $\xi = 1$ . . . . . 141
- 4.36 Snapshots of optimization procedure and effects on the flow field and pollute distribution for the bend channel example in Figure (3.10) for  $\xi = 1$ . Left column, Iteration 2, right column, final iteration. . . . . 142

- 4.37 Snapshots of optimization procedure and effects on the flow field and pollute distribution for the bend channel example in Figure (3.10) for  $\xi = 10$ . Left column, Iteration 2, right column, final iteration. . . . . 145
- 4.38 Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10) for  $\xi = 10$ . . . . . 146
- 4.39 Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration. . . . . 148
- 4.40 Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10). . . . . 149
- 4.42 Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10). . . . . 151
- 4.41 Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration. . . . . 152
- 4.43 Snapshots of optimization procedure for the bend channel example in Figure (3.10) for  $\xi = 0$  and  $\alpha = 1e6$ . Left column, Iteration 2, right column, final iteration. . . . . 154
- 6.1 Effect of the "do nothing" outflow condition shown by streamline plots for flow around an inclined ellipse at  $Re = 500$  after 100 time steps. From [47] . . . . . 164

|      |  |     |
|------|--|-----|
| 6.2  | Effect of the "do nothing" outflow condition shown by stream-line plots for flow through a bifurcating channel at $Re = 20$ .<br>From [46] . . . . . | 165 |
| 6.3  | Some inf-sup stable and unstable spaces. . . . .   | 168 |
| 6.4  | Mean Wall time Vs dof for the various preconditioners, and<br>PARDISO direct solver. . . . .   | 169 |
| 6.5  | Mean Wall time Vs dof for the various preconditioners, and<br>PARDISO direct solver. . . . .   | 170 |
| 6.6  | Result at thread varying for the SMP implementation of the<br>SIMPLE-based GMRES. . . . .  | 171 |
| 6.7  | Time comparison over 3 different degrees of freedom for dif-<br>ferent number of threads. . . . .  | 171 |
| 6.8  | Validation with COMSOL solver for the unsteady 2D example.   | 174 |
| 6.9  | Validation with COMSOL solver for the steady 3D example. .   | 175 |
| 6.10 | L2 error for pressure (left panel) and velocity (right panel). . .   | 177 |
| 6.11 | Pressure and Velocity results. . . . .   | 177 |
| 6.12 | PARAVIEW representation of approximate Womersley solu-<br>tion at $Wo = 12.5$ , time = 1.1s. . . . .   | 178 |
| 6.13 | L2 error for pressure (left panel) and velocity (right panel). . .   | 179 |

- 6.14 "Double Pipe" geometry in mm by INVENTOR [54] . . . . . 179
- 6.15 Relative difference between COMSOL and developed FEM solutions in  $\| \cdot \|_{L^2([0,T],L^2(\Omega))}$  norm at different Re and meshes. . . . . 180
- 6.16 Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 100. Considered instants of time t=0.2s, t=0.5s, t= 8.0s from left to right. . . . . 180
- 6.17 Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 1000. Considered instants of time t=0.5s, t=1.5s, t= 8.0s from left to right. . . . . 181
- 6.18 Cavity geometry. In orange inlet boundary (circle of radius 0.1m centered in (0, 0.5, 0.5)) and in blue outlet boundaries (circles of radius 0.05m, centered in (0.75, 0.5, 0)) and (0.75, 0.5, 1.0). 181
- 6.19 Relative difference between COMSOL and developed FEM solutions in  $\| \cdot \|_{L^2([0,T],L^2(\Omega))}$  norm at different Re and meshes. . . . . 182
- 6.20 Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 100 for t = 8s. Plane xy (left panel), yz (middle panel), and zx (right panel). . . . . 182
- 6.21 Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 1000 for t = 8s. Plane xy (left panel), yz (middle panel), and zx (right panel). . 183

# List of Tables

|     |  |     |
|-----|--|-----|
| 3.1 | Parameter settings in the topology optimization of the double pipe. . . . .                    | 102 |
| 3.2 | Parameter settings in the topology optimization of the unsteady double pipe. . . . .           | 104 |
| 3.3 | Parameter settings in the topology optimization of the 4 terminal device. . . . .              | 105 |
| 3.4 | Parameter settings in the topology optimization of the bend channel. . . . .                   | 106 |
| 3.5 | Parameter settings in the topology optimization of the three-dimensional bend channel. . . . . | 108 |
| 3.6 | Parameter settings in the topology optimization of the 6 terminal device. . . . .              | 109 |
| 3.7 | Parameter settings in the topology optimization of the bend channel. . . . .                   | 112 |



# Nomenclature

|              |   |
|--------------|---|
| $k_s$        | Sink rate of the transported specie. [1/s]  |
| $\gamma_i$   | Decision/design variables. [1].   |
| $\mu$        | Fluid dynamic viscosity. [N/m <sup>2</sup> · s].  |
| $\nu$        | Fluid cinematic viscosity. [J/kg · s].  |
| $\Omega$     | Governing equations domain.   |
| $\rho$       | Fluid density. [kg/m <sup>3</sup> ].  |
| $\mathbf{f}$ | External forcing term of the NS momentum equation (2.4.1) in $\Omega$ . [N].              |
| $\mathbf{n}$ | External normal to $\Omega$ . [1].  |
| $\mathbf{u}$ | Velocity vectorial function in $\Omega$ . Solution of Navier-Stokes system (2.4.1). [m/s] |
| $g$          | Inequality constraint of the TO problem   |
| $J$          | Optimization/objective functional.  |
| $p$          | Pressure scalar function. Solution of Navier-Stokes system (2.4.1)                        |
| $R_h$        | Physic governing equations. Constraint in TO problem                                      |
| $V$          | Volume of the optimization domain covered by fluid.                                       |
| $V_0$        | Volume of the optimization domain. [m <sup>3</sup> ]                                      |
| $V_r$        | Maximal fraction of the initial optimization volume, $V_0$ , covered by fluid. [1]        |

- c Concentration of transported specie. [1]
- D Diffusion coefficient of transported specie. [ $\text{m}^2/\text{s}$ ]

# Chapter 1

## Introduction

The science of optimisation aims of the best possible use of the available resources in a given physical process. Commonly speaking, any optimization strategy answers the following questions: what is the best way to employ resources? Are Nature processes designed to be optimal? The impacts of answering these questions are so important to society that the interest in this topic has been prosperous since the ancient age.

Indeed, starting from greeks analysis for geometrical problems (see for instance Dido's problem of finding the polygon of minimal perimeter and maximum area, analyzed by Zenodorus in 300 BC), many mathematicians and scientists worked on the optimisation field. As a result, some pivotal outcomes were already derived at the rising of the scientific revolution (see Kepler's work on the secretary problem and wine barrel or Galileo's analysis on the best shape for a hanging chain [1]).

However, the scientific community has had at its disposal the suitable tool to handle optimization problems only with the foundation of the Calculus of Variations in the late 17th century. Nowadays, an incredible amount of theoretical properties and procedures have been consolidated in literature, constituting the basis of the knowledge in many scientific fields [2]; further, several optimization algorithms have become unavoidable tools even for simple time scheduling problems [3]. Nevertheless, the progress in the optimization field

is far from over, and the research is still open for more advanced techniques, like the ones in Layout optimization [4].

## **1.1 The Topology Optimization, an overview**

The layout optimization, or rather the research for the optimal disposition of the available material (in the sense of maximizing or minimizing a chosen functional), has applied with growing interest in several production environments in the last decades [5]. Usually, the layout optimization can be categorized into three types, namely, the size optimization, the shape optimization, and the topology optimization. A plastic demonstration of the differences between these three methods is the optimization of the compliance of a cantilever, reported in Figure 1.1.

In the size optimization, the structure is parameterized using sizes and positions of geometrical features, and the corresponding parameters are the optimization variables. The shape optimization, instead, parameterizes the geometrical characteristics of structures using the spline interpolation, where the sample points are the optimization variables. After the parameterization, the optimization variables are iteratively evolved in the predefined feasible regions, using global optimization or gradient information-based optimization algorithm. The size optimization and shape optimization are subjected to the inflexibility on changing the topology of the initial guessed geometry, and thus are characterized by a strong dependence on the starting solution of the problem [6].

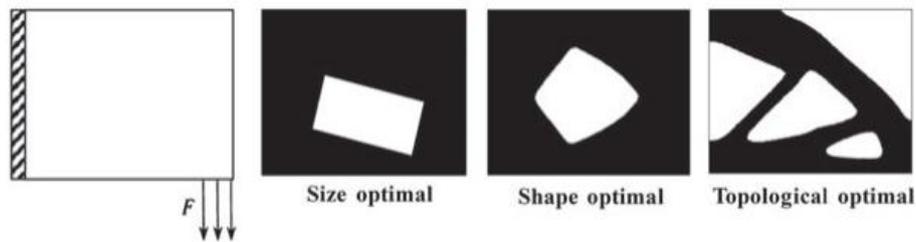


Figure 1.1: *Size optimization, shape optimization and topology optimization of a cantilever with minimizing compliance*

The topology optimization on the other side is less constrained by the initial configuration, and should therefore be a more powerful, flexible and robust tool. Usually the variables of optimization are the material distribution inside the domain. For this reason the topology optimization can fix not only the topology, but also the shape and size of the geometry simultaneously[6].

## 1.2 Motivation and project guidelines

Given the growing interest over the last decade in the Topology Optimization algorithms (TO) [7] [8], this thesis aims, first, to develop from scratch a TO application for fluid flows, second, to extend such TO application to the transport of species in flows.

More specifically, such solver should find a theoretical best material distribution for problems involving laminar flows for incompressible Newtonian fluids, whose physics is governed by the well known Navier-Stokes (NS) equations.

While publications for TO problems on fluid flows increase every year (see Figure 1.2), the state of the art of methods, algorithms and applicability does

not show the same trend. Most of the articles do not present clear improvements and almost no one reports results in terms of performance nor validity of the obtained optimal configurations [8]. Hence, the current research seems more focused on reproducing what is already present in the literature than to propose innovative applications.

For this reason, the proposed model will be validated by comparing not only the results with some well-established optimal configurations, but also the performance to converge at the solution.

The performance of a TO algorithm is essentially played by the choice of the optimizer algorithm and the solver for the Navier-Stokes system. Solving the Navier-Stokes equations is by far the heaviest requirement of the entire optimization process, and it must be performed at each update of the geometry. For this reason the use of a fast numerical technique is of fundamental importance, as well as the features implemented to accelerate the fluid dynamics solution.

Not an irrelevant role, anyway, is played by the choice of the optimizer algorithm. Such a scheme is what dictates the evolution of the geometry in terms of the current flow characteristic, and determines the required number of optimization iterations to achieve the convergence. Pointless to say that the less are the iterations, the less will be the calls to the Navier-Stokes equations solver, with strong benefits in terms of computational costs. On the other hand, fast optimizers may be unstable or have local minimum, thus preventing the achievement of the best material configuration. Some of the most used optimizers will be developed and then compared in chapter 3 and 4.

Another very important part of TO algorithm relies on the so called "adjoint

system". The optimization process consists in a loop of NS system solver, adjoint system solver, and geometry update. The solution of the "adjoint" system is a fundamental requirement to provide at the optimizers enough information on the flow, in order to update the geometry.

The adjoint system does not represent a problematic computational task, because, despite its similarities with the NS system, its formulation is free from non linearities. Accordingly, the solution of the adjoint system can be performed usually faster than that of the governing equations. However, the derivation of the adjoint system is not straightforward. The literature shows only few works where the adjoint system was explicitly derived. For general conditions and time dependent scenarios, but they lack, in any case, of clarity in the computations.

The formulation here proposed has the goal of being simple, but less formal to allow an easier comprehension of the method.

In short, each section of this work will analyze a part of the optimization algorithm and its derivation.

In chapter 2 we discuss the main features of the implemented TO and FEM algorithms, with some basic concepts about solving techniques.

Then, chapter 3 will explain the details of the main methods, which include the derivation of the adjoint system (3.1.1), a discussion of the most popular optimization schemes (3.2), and the derivation of a fast solving technique for the Navier-Stokes system (3.3.2).

Finally, the developed Topology Optimazion solver for fluid flows will be tested over well-validated results reported in the literature, with particular attention over the differences between the various optimization schemes.

The application of this model for problems including the transport of species is an incredibly promising research area for topology optimization algorithms

[8]. For this reason this thesis proposes a parallel derivation of the TO algorithm for a simplified version of the flow-driven scalar transport problem.

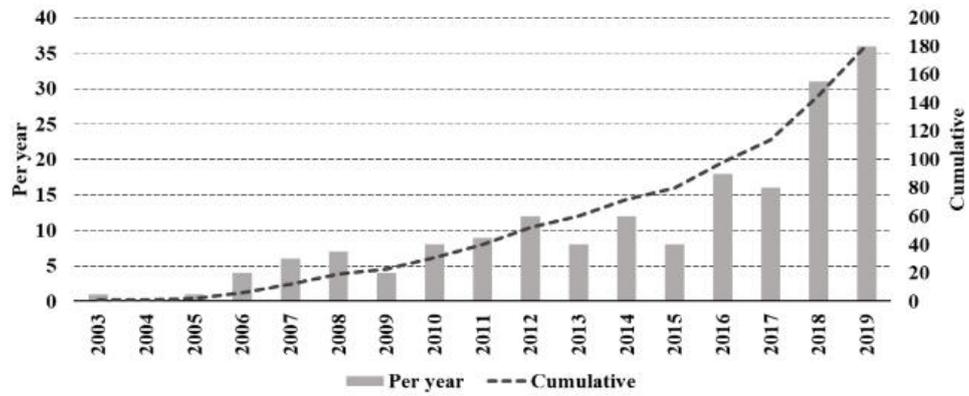


Figure 1.2: Number of papers on TO problems published per year and total accumulated publications over time. [8]

## 1.3 State of the Art

This brief review follows the excellent work of J. Alexandersen and C. Scousboe Andreassen [8], since no big advancement in state of the art has been proposed later.

It is noting that the articles published until 2020 for TO applied to fluid flows are 186. Most of them regards steady-state laminar flow models, with only 13 out of 186 papers treating unsteady problems (see Figure 1.3), probably due to the high requirements in both computational and storage costs. The developing of a fully 3D approach is another issue, mandatory for complex engineering applications, that was treated in only 58 papers (31%). Three dimensional problems and transient analysis are mostly solved with shared memory processing to achieve acceptable performances.

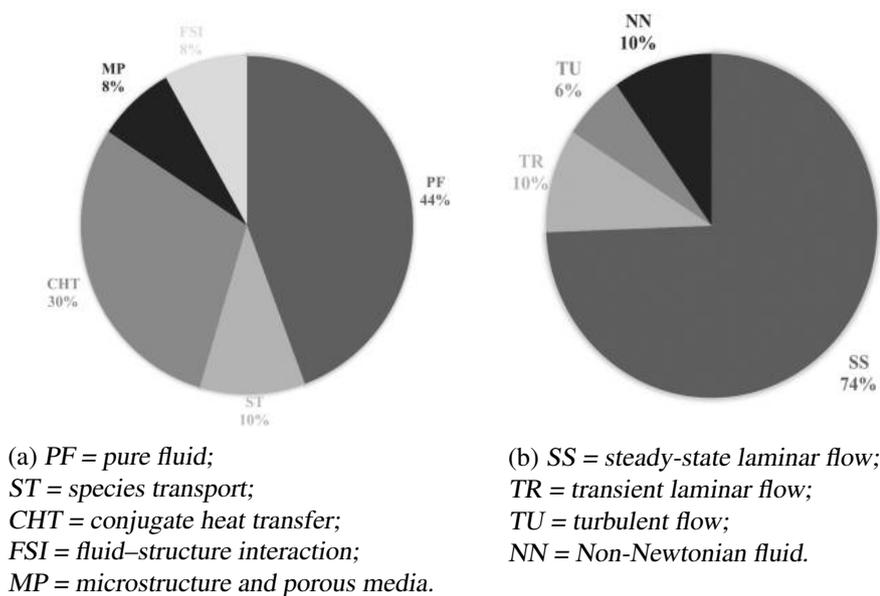


Figure 1.3: Distribution of papers in overall problem type (left). Distribution of papers for fluid model type (right). [8]

TO algorithms mainly differs in three different aspects, i.e., the design ap-

proach for the topology evolution, the non linear optimizer and the discretization techniques for the Navier-Stokes equations.

Among the design approaches, the density and the level set method are the most widely adopted.

The density method is a standard penalization method, in which the design parameters are locally associated to the material porosity of a fictitious material, that is used to differentiate the free channel portions to the solid regions. This scheme is simple, and only requires the additional computation of one term in the governing equations. Besides, it manages extremely well the changes of the topology, allowing both the creation and the filling of holes. However, problems may arise in the parameter calibrations, such as the choice of a proper interpolation rule for material properties or the penalization parameter, to the point that the algorithm can become unstable.

The level set method is more recent, and defines a specific function, called *level set function*, which determines whether a point is within the fluid or solid domain. In common applications, this level set function takes value zero on the fluid-solid interface, positive value on the solid domain and negative value in the fluid domain. The velocity inside the solid domain is identically set to zero, and domain updates are obtained by evolution of the zero-level-set of the level set function.

A well defined and sharp interface facilitates both the iterative scheme and coupling with different physics, producing smoother results for the manufacturability. This method has a more solid mathematical background and it currently is the most used approach. The level set method requires surface-capturing methods, as X-FEM, and can only apply changes in topology by evolving the shape of the existing holes of the domain, so it can remove holes,

but not create them. Accordingly, to allow the convergence in problems where the fluid domain is not given a priori, the initial configuration should always present a large number of cavities.

Discretization schemes instead varies between FEM (Finite Element Methods), FVM (Finite Volume Methods), LBM (Lattice Boltzmann Methods) and PM (Particle Methods). FEM approach covers the majority of the publications, but LBM and PM techniques have only been applied in recent years, and seem to provide better results in terms of solver performances and stability of the solution.

Finally, about the optimization schemes, the 98% of the papers analyzed used gradient-based optimisation approaches, of which Svamberg Methods of Moving Asymptotes is the most employed by far.

Topology Optimization scheme for fluid can be generalized by including a scalar convection-diffusion equation, to optimize the material distribution for temperature field, the concentration of a specie or the volume fractions of two miscible fluids. In such scenarios the flow solution is assumed to not depend on the scalar field. As shown in Figure (1.3), only 19 out of 186 papers, up to the 2020, have studied the effects of the species transport in a TO algorithm.

In this paper TO algorithm is developed with a density method approach, a FEM scheme for the solving of the Navier-Stokes (and eventually Transport) equations, relative adjoint equations and three different gradient-based algorithms for the optimization, namely the Optimality Criteria Method, the Generalized Optimality Criteria (OC and GOC, respectively [9]), and the Method

of Moving Asymptotes (MMA [10]).

The TO algorithm should work on both two and three dimensional problems, and for stationary and time-dependent scenarios.

# Chapter 2

## Numerical models: background

### 2.1 Topology Optimization: general description

The general formulation for a Topology Optimization problem is

$$\left\{ \begin{array}{l} \min_{\gamma} J(\mathbf{x}) \\ \text{subject to : } g_i(\mathbf{x}) \leq 0, \quad i = 1, \dots, m; \\ R_h(\mathbf{x}) = 0, \quad h = 1, \dots, l; \\ \mathbf{x} \in \mathcal{X}, \end{array} \right. \quad (2.1.1)$$

where  $\mathbf{x} = \{x_i, i = 1, \dots, n\}$  are called the decision (or design) variables, in the space  $\mathcal{X} = \{\mathbf{x} \in \mathbf{x}^n, x_i \in \{x_{min}, x_{max}\} \forall i = 1, \dots, n\}$ ,  $J$  the objective functional of the problem,  $g_i$  the constraints, and  $R_h$  the governing equations. Almost every topology optimization problem can be formulated as in (2.1.1), with usually  $x_{min} = 0$ , and  $x_{max} = 1$ , which represent the solid and void volume, respectively.

The theoretical formulation of the problem is then based on discrete design variables, and follows the heuristic goal of perfect discrimination between solid and void regions. However, since the solving techniques for continuous problems are extremely faster than those for discrete variables, in most ap-

plications such exact formulation is always relaxed to  $\mathcal{X} = \{\mathbf{x} \in \mathbf{x}^n, x_i \in [x_{min}, x_{max}] \forall i = 1, \dots, n\}$ . Of course, the final solution should anyway provide a  $\mathbf{x}$  with values as close as possible to  $x_{min} = 0$  or  $x_{max} = 1$ .

As said above, several schemes have been developed in last decades to deal with this discrete-continuous adaptation, like the level set approach, or evolutionary structural optimization (ESO). The simplest but still effective methodology still remains the penalization method, which is based on the idea to "penalize" the physics for every intermediate value of the  $x_i$ . Usually, it is possible by introducing a penalization parameter,  $q$ , and assuming, for example,  $(x_i)^q$ ; in such a way all values lower than  $x_{max} = 1$  produce effects more and more similar to the value  $x_{min} = 0$  as  $q$  is chosen high. Heuristically, for an optimization algorithm this would imply that, to change effectively the functional values, the decision variables have to be clustered as much as possible close to zero or one. Very good results are for example obtained with the SIMP (Solid Isotropic Material with Penalization) method of Bendsøe [11] for structural mechanics problems, where each local elemental stiffness matrix  $K_{el}$  was reformulated as  $K_{el} = x_{el}^q K^*$ , with  $K^*$  corresponding to the standard stiffness matrix for a completely solid element. It is to stress that, since too high values of the penalty parameter usually force the solution to convergence towards a local minimum,  $q$  usually ranges between 2 and 5.

The bottleneck of the model is the numerical solution of the PDEs system,  $R_h = 0$ , influencing the choice for the optimization schemes to the ones with faster convergence and less required calls to the PDEs solver. The three most common algorithms for general optimization algorithms are the gradient-free, the gradient-based, and the hessian-based algorithms.

The gradient-free techniques have almost zero storing requirements and fast

design updates, but are often subject to a local minimum and/or slow convergence, and are thus almost never applied for TO algorithms.

The most applied algorithms are the gradient based, because they balance the required iteration for convergence and storing requirements. Lastly, it is worth noting that the Hessian-based methods would actually lead to faster convergence, but they cannot be used in practical applications, due to the high computational cost and storing requirements. Currently, with high degrees of freedom, some of the best approaches apply gradient-based techniques, which are speed up by the partial knowledge of the hessian matrix, e.g., the like Sequential Linear/Non-linear Quadratic Programming (SLQP/SNLQP) or the Evolutionary Algorithms (see for example Yonekura and Kanno [12]).

A crucial step for TO is computing the functional derivatives as a function of the decision variables, also called sensitivities. This task can be faced with two different approaches, which depend on the characteristics of the problem: the direct or the adjoint approach. However, apart from TO in structural mechanics, almost every solver in literature applies the adjoint technique, by a discrete or continuous formulation [7].

The adjoint approach computes sensitivities of the objective function via a so called "adjoint state", which are used as input data to the gradient-based optimization algorithm, like the MMA. The power of such methodology relies on the fact that the whole sensitivities can be computed via two only solver calls that do not depend on the number on decision variables.

## 2.2 Topology Optimization for fluid flow

For the specific case of fluid flow problems, TO formulation (2.1.1) reduces to

$$\left\{ \begin{array}{ll} \min_{\gamma} J(\gamma) & \\ \text{subject to} & : g_i(\gamma) \leq 0, \quad i = 1, \dots, m \text{ Constraints,} \\ & : 0 \leq \gamma(\mathbf{x}) \leq 1, \quad \text{Design variable bounds,} \\ & : \text{NS equations,} \quad \text{Governing equations.} \end{array} \right. \quad (2.2.1)$$

In (2.2.1), the decision variables are now renamed as  $\gamma_i$ , and are associated to a node of the mesh. As a consequence, the linearized function results  $\gamma = \gamma(\mathbf{x}) = \sum_{j=1}^N \gamma_j \phi_j(\mathbf{x})$ , with  $\phi_j$  an appropriate shape function of the node  $j$ .

The chosen numerical approach to force the clustering of continuous variables for 0/solid to 1/fluid extremes is the so called density penalization method. The idea is to introduce an artificial friction force, proportional to the fluid velocity and weighted by the penalization parameter  $q$ , so that only a negligible fluid fraction can enter in the solid regions. The physical meaning is that the domain is a porous media, whose local porosity is the optimization parameter. Solid and fluid regions then correspond to very low and very high permeability of the medium, respectively.

The density method adopts a friction force,  $\mathbf{f}$ , to approximate the behaviour of the artificial porous media. Assuming laminar flow regime, i.e., low Reynolds

numbers ( $Re = UL/\nu$ , with  $U$  and  $L$  characteristic velocity and length of the problem), the friction force law follows the Stokes' drag, i.e.,  $\mathbf{f} = -\alpha\mathbf{u}$ , with  $\alpha$  appropriate dampen parameter. From a the physical point of view, it is reasonable for the drag force to increase with viscosity and decrease with higher permeability; consequently, the parameter  $\alpha$  has the form  $\alpha(\mathbf{x}) = \mu/k(\mathbf{x})$ , with  $k(\mathbf{x})$ , the local permeability of the medium. This approach is consistent for  $Re < 1$ , but practically works also for higher Reynolds numbers, with minor uncertainties.

The design variable,  $\gamma(\mathbf{x}) \in [0, 1]$ , controls the local permeability of the introduced medium.  $\gamma = 0$  corresponds to the solid material and  $\gamma = 1$  to fluid. Using a convex interpolation to relate  $\alpha(\mathbf{x})$  and  $\gamma(\mathbf{x})$ , the formulation reads

$$\alpha(\gamma) \equiv \alpha_{min} + (\alpha_{max} - \alpha_{min}) \frac{q(1 - \gamma)}{q + \gamma}, \quad (2.2.2)$$

where  $\alpha_{min}$  and  $\alpha_{max}$  are limit values for  $\alpha$ . It is worth noting that the wall is perfectly impermeable for  $\alpha_{max} \rightarrow +\infty$ . However, numerical approximations shows that too large values of  $\alpha$  may create instabilities in the solution. Therefore,  $\alpha_{max}$  needs to be calibrated for each problem. On the contrary, the best value for  $\alpha_{min}$  is 0.

The parameter  $q$  controls the effect of non discrete value of the design parameter, i.e., between 0 and 1, to the value of the damping parameter  $\alpha$ . When  $q$  is large,  $\alpha(\gamma) \simeq \alpha_{min}$  only for  $\gamma$  very close to 1, while all the others intermediate values produce an  $\alpha(\gamma)$  close to its maximum,  $\alpha_{max}$ . For this reason, when  $q$  is large, the topology optimization algorithm is forced to converge to a discrete solutions, or at least to have an almost exact  $\gamma = 1$  value in the fluid region.

It is important to highlight that increasing  $q$  the TO algorithm may be forced to quick discrete convergence, and is than more likely to stop on local minimum.

The role of permeability on the fluid dynamics is quantified by the Darcy number,  $Da$ , which describes the effect of the permeability of the medium versus its cross-sectional area,

$$Da = \frac{k}{L^2} = \frac{\mu}{\alpha_{max}L^2}, \quad (2.2.3)$$

where  $k = \mu/\alpha_{max}$  is the medium permeability and  $L$  the characteristic length of the section. It is evident that almost impermeable materials are associated to very low Darcy numbers. Fixing the threshold to  $Da \leq 10^{-5}$ , one should then always choose an  $\alpha_{max}$  big enough to guarantee impermeable material for  $\gamma = 0$ .

The implemented Optimization algorithm can then be summarized as:

1. To choose an initial value for  $\gamma$ ,
2. To solve Governing equations for  $\mathbf{u}$  and  $p$  with a FEM algorithm (2.5),
3. To compute derivative of objective function and constraints as a function of  $\gamma$ :
  - (a) To solve the adjoint of the Navier-Stokes problem, to eliminate partial derivative of implicit functions from the formulation
  - (b) To compute the gradient of the objective function
4. To use MMA/GOC/OC updating rule to find the value of  $\gamma$  that minimizes  $J$  by means of the past iteration history and gradient informations,

5. To check the convergence rule. If not satisfied, to go back to step (2), otherwise to end the process.

The computational time of the process is mainly given by Point 2, because, for each iteration, the algorithm needs to solve a system of non linear partial differential equations.

### 2.2.1 Objective functionals

Standard optimization problems are usually formulated as minimization problems unless described differently. Therefore a reduction of the objective functional leads to an improved design. Stationary objective functionals are usually classified as boundary or volume based, and their common representation for TO in fluid flows is

$$J(\mathbf{u}[\gamma], p[\gamma]; \gamma) = \beta_1 \int_{\Omega} A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma) d\mathbf{x} + \beta_2 \int_{\partial\Omega} B(\mathbf{u}, p; \gamma) d\sigma, \quad (2.2.4)$$

where  $\beta_1$  and  $\beta_2$  are space independent parameters that can be used to weight the importance of the two terms of the functional (volume and boundary based). Time dependent objective functionals simply take the form

$$J(\mathbf{u}[\gamma], p[\gamma]; \gamma) = \int_0^T \int_{\Omega} \beta_1 A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma) d\mathbf{x} + \int_0^T \int_{\partial\Omega} \beta_2 B(\mathbf{u}, p; \gamma) d\sigma, \quad (2.2.5)$$

with  $T$  the period of time of the considered transient.

Hereinafter we consider for simplicity only the stationary case, keeping in mind that the transient scheme can be obtained just by an additional integration in time.

Objective functionals need to be written as  $J(u, p; \gamma)$ . This general functional

may obviously include the characteristics of the flow (like in the case of minimal energy dissipation), but it still always depend uniquely on the decision variable  $\gamma$ . By Equation (2.2.1), given  $\gamma$ , the material distribution in  $\Omega$  is set, and the governing equations can be immediately solved for  $\mathbf{u}$  and  $p$  without any further assumption.

Reasonably, there are infinite possible formulations for functionals with the structure of (2.2.4), but we will only analyze here the effect of the most popular choice, namely, the minimization of the power dissipation. Other possible and popular objective functionals are described in [13].

### Power dissipation

In this thesis we minimize the power dissipation inside the domain, whose functional reads

$$J = \int_{\Omega} \left[ \frac{1}{2} \mu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^T\|^2 + \alpha \|\mathbf{u}\|^2 \right] d\mathbf{x}, \quad (2.2.6)$$

where  $A = \frac{1}{2} \mu \|\nabla \mathbf{u} + \nabla^T \mathbf{u}\|^2 + \alpha \|\mathbf{u}\|^2$  (see Equation 2.2.8). This formulation is valid in absence of heat forces and adiabatic BCs, and it is derived by scalar multiplying the velocity field with the momentum equation. Using the integration by parts,  $J$  is the sum of the power dissipation due to the viscosity (proportional to the stress tensor), and the power of the external force field. The latter term is  $\mathbf{f} \cdot \mathbf{u}$ , with  $\mathbf{f} = \alpha \mathbf{u}$ , the artificial force field. Applying the divergence theorem to the functional definition [[13], one can easily check, that the above formulation is equivalent to

$$J = \int_{\partial\Omega} [\sigma \cdot \mathbf{u} - \frac{1}{2} \rho \|\mathbf{u}\|^2 \mathbf{u}] \cdot \mathbf{n} d\sigma, \quad (2.2.7)$$

where  $\sigma = \mu \frac{\nabla \mathbf{u} + \nabla^T \mathbf{u}}{2} - p \mathbf{I}$  is the Cauchy stress tensor.  $\sigma \cdot \mathbf{n}$  is thus the external force acting on the boundary, and  $\sigma \cdot \mathbf{u} \cdot \mathbf{n}$  is the related work, which one wants as close as possible to the the work due to the dynamic pressure,  $\frac{1}{2} \rho \|\mathbf{u}\|^2 \cdot \mathbf{u} \cdot \mathbf{n}$ .

In addition, in the case of solid wall BCs, i.e.,  $\mathbf{u} = 0$  along some edges of  $\partial\Omega$ , above formulation reduces to

$$J = - \int_{\partial\Omega} [p + \frac{1}{2} \rho \|\mathbf{u}\|^2] \mathbf{u} \cdot \mathbf{n} d\sigma. \quad (2.2.8)$$

Borrvall and Petersson [14] showed that for Stokes flows with only Dirichlet boundary conditions on  $\partial\Omega$ , the problem of minimizing the total power dissipation inside the domain subject to a volume constraint on the material distribution is mathematically well-posed. Moreover, it was proven that in the case of an  $\alpha$ , which is linear function of  $(\gamma)$ , the optimal material distribution takes only discrete values even in the continuous formulation.

The used interpolation rule is given by Equation (2.2.2), and is in fact almost linear when  $q \gg 1$ .

Hence, for large values of  $q$  the interpolation is almost linear and almost discrete interfaces should be expected, whereas, for small  $q$ , interfaces should be smeared out in the optimized solution. Many hydraulic optimization problems could be formulated according to the dissipation power method. For example, if (2.2.8) holds, when prescribed the flow rate, minimizing the total power dissipation is equivalent to minimizing the pressure drop across the system. If instead a prescribed pressure drop is given, then minimizing the total power dissipation is analogous to maximizing the flow rate [15].

### 2.2.2 Constraints

TO formulation (2.2.1) is adaptable to any constraint  $g(\gamma, \mathbf{u}[\gamma], p[\gamma]) < 0$ . However, since all gradient-based optimization schemes require both the functional and the constraints sensitivities, the treatment of the implicit dependencies  $\mathbf{u}[\gamma], p[\gamma]$  is non trivial, and may require the computation of an additional adjoint problem.

In general, constraints arise from specific needs to reduce the set of admissible solutions, given the characteristics of the analyzed problem.

The only implemented constraint in this work, is the volume constraint, namely

$$\int_{\Omega} \gamma(\mathbf{x}) d\mathbf{x} - \beta|\Omega| \leq 0. \quad (2.2.9)$$

Equation 2.2.9 forces the material distribution to occupy at least  $(1 - \beta)$  of the total domain, and modifies the search for the best configuration among the ones with a maximum flow channel set to  $\beta|\Omega|$ . The volume constraint is widely used in literature, because it prevents the scheme to converge toward trivial solutions [16].

## 2.3 The Topology optimization for Transport of Species

The formulation for Topology Optimization in the case of species transport is just an evolution of 3.5.1, namely

$$\left\{ \begin{array}{ll} \min_{\gamma} J(\gamma) & \\ \text{subject to} & : g_i(\gamma) \leq 0, \quad i = 1, \dots, m \text{ Constraints,} \\ & : 0 \leq \gamma(\mathbf{x}) \leq 1, \quad \text{Design variable bounds,} \\ & : \text{NS equations,} \quad \text{Governing equations,} \\ & : \text{ADR equations,} \quad \text{Governing equations,} \end{array} \right. \quad (2.3.1)$$

with "ADR" standing for "Advection-Diffusion-Reaction". In this case, the procedure to find the optimal material distribution is almost the same of standard fluid dynamics problems. By the density method, the optimal layout is given by the optimization of the material porosity of the domain through a dampen parameter  $\alpha(\gamma)$  defined as in (2.2).

Such formulation, however, allows the functional to be also concentration-dependent,  $J(\gamma, \mathbf{u}[\gamma], p[\gamma], c[\gamma])$ .

For this reason, the computation of the functional sensitivities will now require the solution of an additional adjoint system for the Advection-Diffusion-Reaction problem.

The implemented Optimization algorithm is summarized in the following:

1. To choose an initial value for  $\gamma$ ,
2. To solve Navier-Stokes equations for  $\mathbf{u}$  and  $p$  with a FEM algorithm 2.5,
3. To solve Advection-Diffusion-Reaction equation for  $c$  with a FEM algorithm, using the approximated velocity field  $\mathbf{u}$
4. To compute derivative of objective function and constraints as a function of  $\gamma$ :

- (a) To solve the adjoint of the Advection-Diffusion-Reaction eliminate the partial derivative of  $c[\gamma]$  from the formulation
  - (b) To solve the adjoint of the Navier-Stokes problem, to eliminate partial derivative of implicit functions  $\mathbf{u}[\gamma]$  and  $p[\gamma]$  from the formulation
  - (c) To compute the gradient of the objective function
5. To use MMA/GOC/OC updating rule to find the value of  $\gamma$  that minimizes  $J$  by means of the past iteration history and gradient informations,
  6. To check the convergence rule. If not satisfied, to go back to step (2), otherwise to end the process.

The computational time of the process is again mainly given by the solver for the Navier-Stokes system.

### 2.3.1 Objective functionals and constraints

TO applications for scalar transport problems driven by fluid flows are still new in literature, and the used constraint and objective functionals are all problem-dependent.

#### Pollute dissipation

If the object is to minimize the concentration level of a contaminant at the outlet boundary, a simple formulation is just

$$J = \int_{\Gamma_{out}} c d\sigma. \quad (2.3.2)$$

### Target concentration

If a target value for the final concentration is known a-priori, the inverse-design problem can be formulated with

$$J = \frac{1}{\beta} \ln \int_{\Gamma_{out}} e^{\beta(c-c_{target})^2} d\sigma, \quad (2.3.3)$$

proposed by Makhija and Maute [17], where a larger value for  $\beta$  should increase the accuracy of approximating the maximum value, but may result in numerical issues if chosen too large.

### Measure of mixing

If the goal is to find the best form of a mixer, in the case of fluid flows with different specie concentration, the functional results in [18]

$$J = \frac{\int_{\Gamma_{out}} (c-c_{target})^2 d\sigma}{\int_{(c_0-c_{target})^2} d\sigma}, \quad (2.3.4)$$

where  $\Gamma_{in}$  and  $\Gamma_{out}$  are the inlet and outlet of the mixer, respectively,  $c_0$  is the reference concentration (usually equal to the one specified at the inlet), and  $c_{target}$  is the target concentration, usually chosen as the average concentration.

For the scope of this thesis, the idea is to analyze the pollute dispersion in a closed region subject to a dissipation rate, considering the effects of the geometries to the energy dissipation of the fluid. For this reason the considered functional shows a multi-objective formulation, to minimize both the concentration at the outlet boundary, and the power dissipation inside the domain.

The resulting form is

$$J = \beta_1 \int_{\Omega} \left[ \frac{1}{2} \mu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^T\|^2 + \alpha \|\mathbf{u}\|^2 \right] d\mathbf{x} + \beta_2 \int_{\Gamma_{out}} c d\sigma, \quad (2.3.5)$$

where  $\beta_1$  and  $\beta_2$  are weights used to properly calibrate the functional dependencies.

Also in this case the generalization to unsteady flows just require an additional integration in time.

The constraints for this type of TO problems are usually very similar to the one adopted for the fluid flows scenarios. The volume constraint is then again used for stabilization purposes even for the coupling with species transport.

## 2.4 FEM formulation

### 2.4.1 Navier-Stokes

Recall the complete system for the Navier-Stokes problem

$$\left\{ \begin{array}{ll} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla) \mathbf{u} + \frac{1}{\rho} \nabla p - \nu \Delta \mathbf{u} = \mathbf{f}^* & \text{in } \Omega \times [0, T] \\ \nabla \cdot \mathbf{u} = 0 & \text{in } \Omega \times [0, T] \\ \mathbf{u}(\mathbf{x}) = \mathbf{g}_D(\mathbf{x}), & \text{on } \Gamma_D \times [0, T] \\ \left[ \frac{p}{\rho} \mathbf{I} - \nu \nabla \mathbf{u} \right] \cdot \mathbf{n} = \mathbf{g}_N(\sigma), & \text{on } \Gamma_N \times [0, T] \\ \mathbf{u} = \mathbf{u}_0 & \text{in } \Omega \times \{0\}, \end{array} \right. \quad (2.4.1)$$

with  $\mathbf{x} \in \Omega$  the point inside the domain,  $\mathbf{u}(\mathbf{x})$  the velocity field,  $p$  the pressure field,  $\rho$  the fluid density,  $\nu$  the fluid cinematic viscosity, and  $\mathbf{f}^*$  the body force on the fluid. For TO application with density method, the forcing term is  $\mathbf{f}^* = -\alpha \mathbf{u} + f$ , with  $f$  true body forces of the system. The first two equations in 2.4.1 describes the balance of momentum, and the balance of mass

(continuity equation), respectively, while, in order, the last three equations are respectively the Dirichlet and the Neumann boundary conditions, prescribing either the velocity at the boundary, or the external forces,  $\mathbf{n} \cdot \boldsymbol{\sigma}$ , and the velocity initial condition.

Once defined the mappings  $u(t) : \mathcal{I} \rightarrow \mathcal{V}$  and  $p(t) : \mathcal{I} \rightarrow \mathcal{Q}$ , the solution of the above system takes the form  $(\mathbf{u}, p) \in L^2(\mathcal{I}, \mathcal{V}) \times L^2(\mathcal{I}, \mathcal{Q})$ , where (see chap. 5 [19]):

$$\begin{aligned}\mathcal{V} &= [H_{0,\Gamma_D}^1(\Omega)]^d \\ \mathcal{Q} &= L^2(\Omega),\end{aligned}\tag{2.4.2}$$

with  $d$  the space dimension of the problem,  $L^2$  Lebesgue space of squared integrable functions,  $H^1$  standard Hilbert space,  $\Gamma_D$  Dirichlet subset of the boundary and

$$H_{0,\Gamma_D}^1(\Omega) = \{\mathbf{v} \in H^1(\Omega) \mid \mathbf{v} = 0 \text{ in } \Gamma_D \subset \partial\Omega\}\tag{2.4.3}$$

It is well known that resulting weak formulation for test functions  $\mathbf{v} \in \mathcal{V}$ , and  $q \in L^2(\Omega)$  is

$$\begin{aligned}\int_{\Omega} (\partial_t \mathbf{u}) \cdot \mathbf{v} d\mathbf{x} + \int_{\Omega} [\mathbf{u} \cdot \nabla \mathbf{u}] \cdot \mathbf{v} d\mathbf{x} + \int_{\Omega} \nu \nabla \mathbf{u} : \nabla \mathbf{v} d\mathbf{x} \\ - \frac{1}{\rho} \int_{\Omega} (\nabla \cdot \mathbf{v}) p d\mathbf{x} + \int_{\partial\Omega} \left( \frac{1}{\rho} p \mathbf{I} - \nu \Delta u \right) \cdot \mathbf{n} = \int_{\Omega} \mathbf{f}^* \cdot \mathbf{v} d\mathbf{x}, \\ \int_{\Omega} q \nabla \cdot \mathbf{u} = 0,\end{aligned}\tag{2.4.4}$$

which in compact notation is equivalent to

$$\begin{aligned}
 (\partial_t \mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + m_\alpha(\mathbf{u}, \mathbf{v}) &= F(\mathbf{v}) - \frac{1}{\rho} (\mathbf{g}, \mathbf{v})_{\Gamma_N}, \\
 b(\mathbf{u}, q) &= 0,
 \end{aligned} \tag{2.4.5}$$

for all test functions,  $\mathbf{v} \in \mathcal{V}$  and  $q \in \mathcal{Q}$ , with

$$\begin{aligned}
 (\mathbf{v}, \mathbf{w}) &= \int_{\Omega} \mathbf{v} \cdot \mathbf{w} d\mathbf{x} & n(\mathbf{v}, \mathbf{w}, \mathbf{z}) &= \int_{\Omega} (\mathbf{v} \cdot \nabla \mathbf{w}) \cdot \mathbf{z} d\mathbf{x} \\
 a(\mathbf{v}, \mathbf{w}) &= \nu \int_{\Omega} \nabla \mathbf{v} : \nabla \mathbf{w} d\mathbf{x} & m_\alpha(\mathbf{v}, \mathbf{w}) &= \frac{1}{\rho} \int_{\Omega} \alpha(\mathbf{r}) \mathbf{v} \cdot \mathbf{w} d\mathbf{x} \\
 b(\mathbf{v}, q) &= -\frac{1}{\rho} \int_{\Omega} q \nabla \cdot \mathbf{v} d\mathbf{x} & F(\mathbf{v}) &= \int_{\Omega} \mathbf{f} \cdot \mathbf{v} d\mathbf{x} \\
 (g, v)_{\Gamma_N} &= \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{v} d\sigma
 \end{aligned}$$

Galerkin discretized formulation, using backward Euler time-stepping in matrix form, reads

$$\begin{bmatrix} \left( \frac{M}{\Delta t} + N(\mathbf{u}_h^{n+1}) + M_\alpha + H \right), & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_h^{n+1} \\ p_h^{n+1} \end{bmatrix} = \begin{bmatrix} \frac{1}{\Delta t} M \mathbf{u}^n + \mathbf{f} - \mathbf{f}_N \\ 0 \end{bmatrix}, \tag{2.4.6}$$

where

$$\begin{aligned}
M &= \{m_{ij}\} & m_{ij} &= (\mathbf{w}_i, \mathbf{w}_j) = \int_{\Omega} \mathbf{w}_i \cdot \mathbf{w}_j d\mathbf{x} & i, j &= 1, \dots, N; \\
N(\mathbf{u}_h) &= \{n_{ij}(\mathbf{u}_h)\} & n_{ij}(\mathbf{u}_h) &= n(\mathbf{u}_h, \mathbf{w}_i, \mathbf{w}_j) = \int_{\Omega} [(\mathbf{u}_h \cdot \nabla) \mathbf{w}_i] \cdot \mathbf{w}_j d\mathbf{x} & i, j &= 1, \dots, N; \\
M_{\alpha} &= \{m_{\alpha ij}\} & m_{\alpha ij} &= (\alpha \mathbf{w}_i, \mathbf{w}_j) = \int_{\Omega} \alpha(\mathbf{x}) \mathbf{w}_i \cdot \mathbf{w}_j d\mathbf{x} & i, j &= 1, \dots, N; \\
H &= \{h_{ij}\} & h_{ij} &= a(\mathbf{w}_i, \mathbf{w}_j) = \nu \int_{\Omega} \nabla \mathbf{w}_i : \nabla \mathbf{w}_j d\mathbf{x} & i, j &= 1, \dots, N; \\
B &= \{b_{ki}\} & b_{ki} &= b(\mathbf{w}_i, J_k) = -\frac{1}{\rho} \int_{\Omega} \phi_k \nabla \cdot \mathbf{w}_i d\mathbf{x} & k &= 1, \dots, M; i = 1, \dots, N; \\
\mathbf{f} &= \{f_i\} & f_i &= \int_{\Omega} \mathbf{f} \cdot \mathbf{w}_i d\mathbf{x} & i &= 1, \dots, N, \\
\mathbf{f}_N &= \{f_{Ni}\} & f_{Ni} &= \frac{1}{\rho} \int_{\Gamma_N} \mathbf{g} \cdot \mathbf{w}_i d\sigma & i &= 1, \dots, N,
\end{aligned}$$

with  $\mathbf{w} \in \mathcal{V}_h(\mathcal{T}_h^V) = \{v \in \mathcal{C}^0(\Omega) : v|_T \in [\mathcal{P}_1(T)]^d, \forall T \in \mathcal{T}_h^V\} = \text{Span}(w_1, \dots, w_{dN_V}) \subset [H_{\Gamma_D}^1(\Omega)]^d$ , and  $\phi \in \mathcal{Q}_h(\mathcal{T}_h^P) = \{q \in \mathcal{C}^0(\Omega) : q|_T \in \mathcal{P}_1(T), \forall T \in \mathcal{T}_h^P\} = \text{Span}(\phi_1, \dots, \phi_{N_P}) \subset H_{\Gamma_D}^1(\Omega)$ , the test functions in the velocity and pressure spaces respectively. Dirichlet boundary conditions,  $\mathbf{u} = g_D$  on  $\Gamma_D$  are then be imposed via lifting operators (see 6.1).

The inf-sup stability condition [20] is guaranteed by the choice of  $\mathcal{P}_1 - iso \mathcal{P}_2/\mathcal{P}_1$  elements [19], through an enrichment of the velocity space as in Figure 2.1, where  $\mathcal{T}_h^V$  and  $\mathcal{T}_h^P$  are triangulation for velocity and pressure spaces respectively (see appendix 6.2 for details).

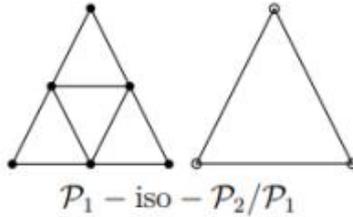


Figure 2.1:  $\mathcal{P}_1 - iso \mathcal{P}_2/\mathcal{P}_1$  stable elements. Pressure  $\mathcal{T}_h^P$  (left) and velocity  $\mathcal{T}_h^V$  (right) discretization.

The convection term  $n(\mathbf{u}_h^{n+1}, \mathbf{u}_h^{n+1}, \mathbf{v})$  is linearized by a Picard approach, and a Streamline-Upwind-Petrov-Galerkin scheme stabilizes oscillations due to convection-diffusion treatment (see [21]).

Three different types of boundary conditions have been implemented, namely, the Dirichlet, the Neumann and the Symmetry conditions (see appendix 6.1).

## 2.4.2 Advection-Diffusion-Reaction

The unsteady advection-diffusion-reaction problem subject to an incompressible velocity field can be formulated as

$$\begin{cases} \partial_t c + \mathbf{u} \cdot \nabla c - \nabla \cdot (D \nabla c) = R & \text{in } \Omega \times [0, T] \\ c(\mathbf{x}) = g_D(\mathbf{x}), & \text{on } \Gamma_D \times [0, T] \\ D \nabla c \cdot \mathbf{n} = g_N(\sigma), & \text{on } \Gamma_N \times [0, T] \\ c = c_0 & \text{in } \Omega \times \{0\}, \end{cases} \quad (2.4.7)$$

with  $c$  concentration of the transported specie,  $D$  diffusion coefficient,  $\mathbf{u}$  the fluid velocity and  $R$  source/sink term. Let in the following consider  $R = -k_s c + r$ , to split concentration-dependent terms from independent ones, assuming a default sink term  $-k_s c$ , with  $k_s$  sink rate.

The weak formulation for the problem, assuming test functions and solution  $c \in \mathcal{V} = H_{\Gamma_D}^1(\Omega) = \{v \in H^1(\Omega), v = 0 \in \Gamma_D\}$ , is

$$\int_{\Omega} [\partial_t c - \text{div}(D \nabla u) + \mathbf{u} \cdot \nabla c + kc] v d\Omega = \int_{\Omega} r v d\mathbf{x}, \quad (2.4.8)$$

which, using Green's Lemma, reduces to

$$(\partial_t c, v) + a(c, v) + b(c, v) + m_k(c, v) - (g_N, v)_{\Gamma_N} = F(v) \quad (2.4.9)$$

with

$$(v, w) = \int_{\Omega} v w d\mathbf{x}; \quad a(c, v) = \int_{\Omega} D \nabla c \cdot \nabla v d\mathbf{x} \quad (2.4.10)$$

$$b(v, w) = \int_{\Omega} \mathbf{u} \cdot \nabla v w d\mathbf{x} \quad m_k(v, w) = \int_{\Omega} k c w d\mathbf{x} \quad (2.4.11)$$

$$(v, w)_{\partial\Gamma_N} = \int_{\Gamma_N} v w d\sigma, \quad F(v) = \int_{\Omega} r d\mathbf{x} \quad (2.4.12)$$

where  $\mathbf{n}$  is the external normal to the domain. Dirichlet boundary conditions can then be imposed via lifting operators (see 6.1). Consequently, the Galerkin discretized formulation, approximating the test space  $\mathcal{V}$  with  $\mathcal{V}_h$ , reads

$$\mathcal{V}_h = \{v \in C^0(\bar{\Omega}) : v|_T \in \mathcal{P}^1(T) \forall T \in \mathcal{T}_h\}, \quad (2.4.13)$$

$\mathcal{T}_h$  is the given triangulation, gives

$$\left[ \frac{1}{\Delta t} M + H + B + M_k \right] c_h = f + f_N, \quad (2.4.14)$$

where

$$\begin{aligned}
M &= \{m_{ij}\} & m_{ij} &= (\phi_i, \phi_j) = \int_{\Omega} \phi_i \cdot \phi_j d\mathbf{x} & i, j &= 1, \dots, N; \\
H &= \{h_{ij}\} & h_{ij} &= a(\phi_i, \phi_j) = \int_{\Omega} D \nabla \phi_i : \nabla \phi_j d\mathbf{x} & i, j &= 1, \dots, N; \\
B &= \{b_{ij}\} & b_{ki} &= b(\phi_i, \phi_j) = \int_{\Omega} (\mathbf{u} \cdot \nabla \phi_j) \phi_i d\mathbf{x} & i, j &= 1, \dots, N; \\
M_k &= \{m_{ki,j}\} & m_{ki} &= \int_{\Omega} k \phi_i, \phi_j d\mathbf{x} & i, j &= 1, \dots, N, \\
f &= \{f_i\} & f_i &= \int_{\Gamma_N} r \phi_i d\sigma & i &= 1, \dots, N, \\
f_N &= \{f_{Ni}\} & f_{Ni} &= \int_{\Gamma_N} g_N \phi_i d\sigma & i &= 1, \dots, N,
\end{aligned}$$

$\phi_i \in \mathcal{V}_h, i = 1, \dots, N, N$  number of nodes of the discretization.

### Stabilization term: Streamline Upwind diffusion

Several authors in the literature [21] [19] discussed the importance of stabilization techniques for convection-dominated problems, to preserve coercivity of the system bilinear form and guarantee the uniqueness of the solution. For this reason a simple Streamline Upwind diffusion (SUD) approach has been considered, in the form

$$S = \{s_{ij}\}, \quad s_{ij} = \sum_{T \in \mathcal{T}_h} \int_T \tau \frac{h_{el}}{|\mathbf{u}_{el}| D_{el}} (\mathbf{u} \cdot \nabla \phi_j) (\mathbf{u} \cdot \nabla \phi_i) d\mathbf{x}, \quad (2.4.15)$$

with  $h_{el}$ ,  $|\mathbf{u}_{el}|$ , and  $D_{el}$ , respectively the reference size, velocity and diffusion coefficient for the local element  $T$ . The stabilization term,  $S$ , in Equation 2.4.15, adds "artificial diffusion" in the direction of the streamlines, scaled by a calibration parameter  $\tau$ .

By introducing this additive stabilizing term,  $S$ , the final system becomes

$$\left[ \frac{1}{\Delta t} M + H + B + M_k + S \right] c_h = f + f_N. \quad (2.4.16)$$

Note that the stabilization terms depends on the mesh Péclet number [19], locally defined element by element as

$$\mathcal{P}e_h = \frac{|\mathbf{u}_{el}| h_{el}}{D_{el}},$$

thst gives an approximate measure of the local ratio between convection and diffusion terms. As the mesh is refined, i.e.,  $h \rightarrow 0$ , the local Péclet numbers vanishes, and the stabilization term is not necessary. However, in practical application still misses a theory that determines an exact value of  $\mathcal{P}e_h$  that guarantees convergence. For this reason the parameter  $\tau$  has calibrated independently for each of the considered problems.

## 2.5 Solver

### 2.5.1 Navier-Stokes

As explained in the introduction, to solve the NS system of equation (2.4.1) is by far the heaviest process of the complete TO algorithm.

Numerical solvers for linear systems can essentially be divided in two big classical groups, namely, the direct and the iterative methods [22]. The direct methods provide the exact solution to the system through a succession of simple operations, whose number depends on the size of the system matrix. Most direct methods rely on the acceleration of the Gaussian elimination procedure and, whether on one hand they can be extremely fast for two-dimensional problems, where the dof number is relatively small, on the other hand, they are not suitable for 3D scenarios, when the problem size is large, and requires high CPO time and storage capacity.

The path to solve three-dimensional problems is then through iterative methods, which build a sequence of approximations of the final solution to eventually reach the exact one, if the scheme is convergent. Usually these methods use a stopping criterion associated to the residual norm of the current solution, which blocks the algorithm when a "sufficiently good" approximation is found.

Let us consider the system of equation  $K\mathbf{x} = \mathbf{b}$ , with  $K \in \mathbf{R}^n \times \mathbf{R}^n$ ,  $\mathbf{x}, \mathbf{b} \in \mathbf{R}^n$ ,  $n \in \mathbf{N}$  number of unknowns. Calling  $\mathbf{x}_k$  the approximate solution at step  $k$  and  $\mathbf{h}$  the real solution, the error and residual vector can be defined as

$$\boldsymbol{\epsilon}_k = \mathbf{h} - \mathbf{x}_k$$

$$\mathbf{r}_k = K\mathbf{h} - K\mathbf{x}_k.$$

The matrix associated to the NS system (2.4.1), can be synthetically formulated as

$$K = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix}, \quad (2.5.1)$$

with  $A = (M/\Delta t + N(\mathbf{u}_h) + M_\alpha + H)$ , and shows a strong non symmetry as the convective is large.

There are several possible iterative algorithms used to solve non-symmetric systems, mainly based on Krylov subspaces, like GMRES (Generalized Minimal Residual), or QMR (Quasi Minimal Residual) [23].

However, from a practical viewpoint, the choice of the iterative method is not so relevant for efficient computations, but the choice of the preconditioner.

For this reason a simple GMRES scheme has been implemented, and in the following sections are described some possible formulations of good preconditioners.

### GMRES scheme

The Generalized Minimal Residual method is a famous iterative method for linear system, firstly introduced by Saad and Shultz in 1986 [24]. It is based on finding the approximate solution  $\mathbf{x}_m$  through a subspace  $\mathcal{K}_m \subset \mathbf{R}^n$  of size  $m < n$  by the minimization of the current residual norm  $\|\mathbf{r}_m\|_2$ .

The space  $\mathcal{K}_m$  is defined as

$$\mathcal{K}_m(K, \mathbf{r}_0) = \text{span}\{\mathbf{r}_0, K\mathbf{r}_0, K^2\mathbf{r}_0, \dots, K^{m-1}\mathbf{r}_0\}, \quad (2.5.2)$$

where  $\mathbf{r}_0$  is the residual related to the initial guess  $\mathbf{x}_0$ .  $\mathcal{K}_m$  is called *Krylov subspace* of size  $m$  generated by  $\mathbf{K}$  and  $\mathbf{r}_0$ .

Approximate solutions are then found through the search of the best correction  $\mathbf{z} \in \mathcal{K}_m$  of the initial guess, namely

$$\mathbf{x}_m = \mathbf{x}_0 + \mathbf{z}, \quad \mathbf{z} \in \mathcal{K}_m(K, \mathbf{x}_0), \quad (2.5.3)$$

where  $\mathbf{z}$  is given by  $\operatorname{argmin}_z \|\mathbf{b} - K(\mathbf{x}_0 + \mathbf{z})\|_2$ .

Therefore, at each iteration, the algorithm has to enlarge the Krylov subspace by computing a new linearly independent (l.i.) vector, like  $K^m \mathbf{r}_0$ . Defining with,  $\mathbf{v}_1, \dots, \mathbf{v}_m$  the  $m$  vectors in the current Krylov subspace, and with  $V_m$  the  $n \times m$  matrix that has them as columns, the new l.i. vector is computed as  $\mathbf{w} = AV_m$ , and is then orthogonalized against all the other vectors in  $V_m$ . Starting from the initial residual,  $\mathbf{r}_0$ , the algorithm enlarges the current Krylov subspace by creating a vector,  $\mathbf{w} = K^m \mathbf{r}_0$ , with  $m$  current iteration, l.i. from the other vector in the base, adding it to the current base after the orthogonalization with respect to the other vectors,  $\mathbf{v}_1, \dots, \mathbf{v}_m$ , in the base. The orthogonalization process is a necessary process in finite arithmetic, because the vectors  $\mathbf{r}_0, K\mathbf{r}_0, \dots, K^{m-1}\mathbf{r}_0$  are "little linearly dependent" [22], and it is usually performed by Householder process or modified Graham-Schmidt scheme, chosen in this work.

Residual at step  $m$  is then  $\mathbf{r}_m = \mathbf{b} - K(\mathbf{x}_0 + V_m y) = \mathbf{r}_0 - KV_m y$ , with  $y \in \mathbf{R}^m$  the vector containing the coefficients for the optimal linear combination of  $\mathbf{v}_1, \dots, \mathbf{v}_m$ .

It can be proved [22], that, naming  $\bar{H}_m \in \mathbf{R}^{m+1} \times \mathbf{R}^m$  the upper Hessemberg matrix defined by  $h_{jk} = \mathbf{v}_j^T K \mathbf{v}_k$ ,

$$KV_m = V_{m+1} \bar{H}_m. \quad (2.5.4)$$

With this result, the residual at iteration  $m$  reads

$$\mathbf{r}_m = \mathbf{r}_0 - KV_m z = V_{m+1}(\beta \mathbf{e}_1 - \bar{H}_m y),$$

with  $\beta = \|\mathbf{r}_0\|_2$  and  $\mathbf{e}_1 = [1, 0, \dots, 0]^T \in \mathbf{R}^{m+1}$ .

Once the new vector of the base,  $\mathbf{v}_{m+1}$ , is defined, the scheme aims at the best combination of the vectors in the base,  $\mathbf{z}$ , which minimizes the 2-norm of the residual, i.e.,

$$\operatorname{argmin}_y \|\mathbf{r}_m\|_2 = \|\beta \mathbf{e}_1 - \bar{H}_m y\|_2,$$

being  $V_{m+1}^T V_{m+1} = \mathcal{I}$  by construction. Finally,  $z$  is usually obtained by a QR factorization of  $H_m$ .

The overall GMRES scheme runs as follows [25]:

### GMRES

1. Choose  $x_0$  and a restart parameter  $m$
2. **Arnoldi process:**
  - For**  $j = 1, \dots, m$ 
    - (a) Compute  $\mathbf{w} = K\mathbf{v}_j$
    - (b) For  $i = 1, \dots, j$ 

$$\begin{cases} h_{i,j} = \langle \mathbf{w}, \mathbf{v}_i \rangle \\ \mathbf{w} = \mathbf{w} - h_{i,j}\mathbf{v}_i \end{cases}$$
    - (c) Compute  $h_{j+1,j} = \|\mathbf{w}\|_2$  and  $\mathbf{v}_{j+1} = \mathbf{w}/h_{j+1,j}$ .
3. Define  $\mathcal{V}_m = [\mathbf{v}_1, \dots, \mathbf{v}_m]$
4. Form the approximate solution:
  - (a) Compute  $y = \operatorname{argmin}_y \|\beta \mathbf{e}_1 - \bar{H}_m y\|_2$
  - (b) Compute  $x_m = x_0 + \mathcal{V}_m z$

5. If residual less then tolerance STOP, else restart and go to 2.

The restart parameter,  $m$ , is usually set by the available storing limits.

### Preconditioning

The GMRES algorithm has in principle a maximum possible number of iterations required to achieve the exact solution, due to its finite termination property [22]. The only possible breakdown of the algorithm occurs when the the new computed vector  $K\mathbf{v}_m$ , used to enlarge the Krylov subspace, is linearly dependent from the other vectors in the base. In this case the Krylov subspace has reached its maximum dimension ("luckily" smaller than  $N$ ) and we can compute immediately the exact solution.

Using the Chebyshev polynomials it can be proved that at the  $k_{th}$  step, in case of a diagonalizable matrix system  $K$ , the residual norm is bounded by

$$\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_0\|_2} \leq k(U) \min_{\mathcal{P}_k} \|p_k(\Lambda)\|_\infty,$$

with  $U$  the eigenvector matrix of  $K$  and  $\mathcal{P}_k$  the space of polynomials of degree  $k$  and  $p_k \in \mathcal{P}_k$ . Better bounds arises in case of a symmetric matrix or when the symmetric part of  $K$  is positive definite [26].

A sufficient condition for a good preconditioner, is that the preconditioned matrix  $T = KM^{-1}$  has a low degree minimum polynomial, that is a small number of different eigenvalues [27]. The maximum dimension dimension of the Krylov subspace is indeed bounded by the number of different eigenvalues of  $K$  [28], and computational experience indicates that a good clustering of the eigenvalues of the system matrix far from the origin of the complex plane usually leads to a fast convergence.

In this sense a matrix  $M$ , whose inverse approximates the one of  $K$ , would produce a matrix with eigenvalues closer to the unity, and should likely provide the convergence in few iterations.

When not specified all the approximated matrix have been used as right preconditioners.

1. *left preconditioning*

$$M^{-1}Kx = M^{-1}b$$

2. *right preconditioning*

$$KM^{-1}x^* = b, \quad x = M^{-1}x^*.$$

### Jacobi type preconditioner

The Jacobi preconditioner is the simplest, but in general widely used preconditioner. According to this scheme,  $M$  is taken as the diagonal matrix with the same entries of the main diagonal of  $K$ . This method clearly leads to a huge simplification of  $K^{-1}$ , but is of great ease implementation and could actually work fine with diagonally dominant matrices.

Unfortunately, in the Navier-Stokes problem (2.4.1), the pattern of the system matrix does not allow this specific condition, but shows a particular block matrix formulation, with all zeros in the last  $m \times m$  entries (m number of pressure nodes) leading to a zero division.

A trivial idea to adapt the Jacobi preconditioner for the NS system, is to substitute all the zeros entries in the diagonal with ones.

The Jacobi preconditioning requires almost the same cost of the preconditioner-free GMRES scheme, but some better approaches can be derived, based on the particular block matrix form of the system.

### Schur based preconditioner

The Schur preconditioner exploit the particular block pattern of the NS matrix.

The Schur decomposition of a block matrix, is

$$K = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix}. \quad (2.5.5)$$

A variety of articles, like [29] use the so called "triangular preconditioner" for the right or left preconditioners. The idea is to approximate the product of the two left (or right) matrices in (2.5.5), and use it as left (or right) preconditioner.

Let us choose the right preconditioners,

$$M = \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} = \begin{bmatrix} A & B^T \\ 0 & S \end{bmatrix},$$

whose inverse is

$$M^{-1} = \begin{bmatrix} A^{-1} & -B^T \\ 0 & S^{-1} \end{bmatrix},$$

where  $S = -BA^{-1}B^T$  is the Schur complement.

Hence, in this way, the multiplication of the system matrix with the preconditioner,

$$KM^{-1} = \begin{bmatrix} I & 0 \\ B & I \end{bmatrix},$$

produce a triangular matrix, with all ones in the diagonal and thus all eigenvalues clustered to one.

In practice, the exact construction of  $A^{-1}$  or  $S^{-1}$  is impossible due to storage and CPU requirements; consequently, approximated matrices need to be derived in some smart but simple ways. Now all these matrices have non zeros

on the main diagonal, then it is natural to approximate them just with their diagonal, similarly to the Jacobi approach.

The inverses of  $A$  and the Schur complement  $S$  are thus  $M_1^{-1} = \text{diag}(A)^{-1}$  and  $M_2^{-1} = \text{diag}(-BM_1^{-1}B^T)^{-1}$ , respectively.

Including the left Schur matrix as left preconditioner, the original linear system is reduced to

$$P_L^{-1}KP_R^{-1}y = b, \quad x = P_L^{-1}y,$$

with

$$P_L^{-1} = \begin{bmatrix} I & 0 \\ -BM_1^{-1} & I \end{bmatrix}$$

$$P_R^{-1} = \begin{bmatrix} M_1^{-1} & -B^T \\ 0 & M_2^{-1} \end{bmatrix}.$$

This method has proven to be extremely faster than the standard GMRES or even the Jacobi preconditioned one.

## 2.5.2 Advection-Diffusion-Reaction

The advection-diffusion-reaction problem is solved with the prescribed velocity field obtained by solving the NS problem (2.4.1). For this reason, the two systems are called "one-way" coupled, meaning that the two systems do not need to be solved simultaneously.

The convective term in ADR problem leads to the following non symmetric matrix:

$$K = \begin{bmatrix} \frac{1}{\Delta t}M + H + B + M_k + S \\ \end{bmatrix},$$

which is iteratively solved by a GMRES scheme.

In this case the matrix does not show any particular block formulation to exploit for fast preconditioning. Since the number of unknowns is extremely lower than for the NS system, the solving process is faster. Accordingly, a simple Jacobi preconditioning is sufficient to speed up the solution, and the computational time is negligibly small compared with the time required to solve the fluid dynamics problem.

Both the ADR and NS solvers have been validated in appendix C (6.4).

# Chapter 3

## Materials and Methods

### 3.1 Adjoint derivation

#### 3.1.1 The Adjoint derivation for the Navier-Stokes equations

Let us consider at first the case for a steady solution of Navier-Stokes system (2.4.1) and a stationary functional (2.2.4). The adjoint system is a fundamental tool to manage the implicit dependence of the state variables  $u$  and  $p$  from the design variable  $\gamma$  when computing the functional sensitivities.

Recall now some known theoretic results for derivatives of operators.

The Fréchet derivative of a function at a point  $u \in U$ ,  $f : U \rightarrow \mathcal{W}$ , with  $U \subset \mathcal{V}$ , open set, and  $\mathcal{V}, \mathcal{W}$  Banach spaces, is defined as the bounded linear operator  $D : \mathcal{V} \rightarrow \mathcal{W}$  that satisfies

$$\lim_{\|v\|_{\mathcal{V}} \rightarrow 0} \frac{\|f(u+v) - f(u) - Df(v)\|_{\mathcal{W}}}{\|v\|_{\mathcal{V}}} = 0,$$

whenever it exists. Fréchet differentiability is thus just an extension of the total derivative to linear normed spaces, in the sense that if  $f$  is defined on an open set in  $\mathbf{R}^n$ , with  $\|\cdot\|$  being the Euclidean norm, then  $Df(x, h)$  is exactly

equivalent to the total differential of  $f$ . In the Fréchet derivatives, most of the results of the total derivative, hold, such as the like linearity or the chain rule for derivation, i.e.,

$$D(g \circ f)(u) = Dg(f(u)) \circ Df(u).$$

The Functionals in (2.2.4) are actually  $J : U \times Q \times W \rightarrow \mathbf{x}$ , with  $U \subset [H^1_\Gamma(\Omega)]^d$ ,  $Q \subset L^2(\Omega)$  and  $W \subset L^2(\Omega)$ ,  $u : W \rightarrow U$ ,  $p : W \rightarrow Q$ , hence  $U$ ,  $Q$  and  $W$  are Banach spaces. It can be easily proved that all the functionals and operator used in the present model are Fréchet differentiable in each point of the domain  $\Omega$ . In this case, the Fréchet derivatives coincides with Gateaux derivatives, which are much simpler to obtain.

With the same conditions stated for the Fréchet derivative ( $\mathcal{V}$ ,  $\mathcal{W}$  Banach spaces and  $U \subset \mathcal{V}$ , open set), the Gateaux derivative at a point  $u \in U$  is defined as the bounded and linear operator  $d_u f : U \rightarrow W$ , such that

$$\lim_{t \rightarrow 0} \frac{f(u + tv) - f(u)}{t} = d_u f(v), \quad \forall v \in U,$$

whenever it exists. Functions  $d_u f(v)$  are called Gateux derivatives of  $f$  at  $u$  in direction  $v$ .

Hereinafter, both the Fréchet and Gateaux derivatives will be identified with the standard derivative symbols. In addition, each of the operator will be considered as Fréchet differentiable in  $\Omega$ . More details can be found in [30].

With such concepts, we can write the explicit dependence of the gradient of  $J$ , by chain rule, i.e.,

$$\frac{d}{d\gamma}[J(\mathbf{u}[\gamma], p[\gamma], \gamma)] = \frac{\partial J}{\partial \gamma} + \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{d\gamma} + \frac{\partial J}{\partial p} \frac{dp}{d\gamma}. \quad (3.1.1)$$

Since  $\mathbf{u}[\gamma]$  and  $p[\gamma]$  are implicit, direct evaluation the derivative  $\frac{d}{d\gamma}$  and  $\frac{\partial p}{\partial \gamma}$  is not a simple task. The adjoint method allows us to overcome this issue. In this frame, a complete formal derivation is provided for optimization of constrained PDEs by Ulbrich [31]. Here, we prefer to present a still rigorous but more intuitive derivation.

The ground idea is that the dependence of  $\mathbf{u}$  and  $p$  on  $\gamma$  may not be of trivial, but both variables are uniquely determined once the material distribution (i.e.  $\gamma$ ) has been set via the Navier-Stokes system. In other words, this means that for each possible  $\gamma$ , each of the Navier-Stokes equations must be verified for the computed value of  $\mathbf{u}$  and  $p$ , and then so must be their weak formulation.

The steady NS weak formulation residual for a general test function  $\mathbf{v} \in [H^1(\Omega)]^d$  (not  $[H_{\Gamma_D}^1]^d$ ) and  $q \in L^2(\Omega)$  is

$$\begin{aligned} R_u(\mathbf{u}[\gamma], p[\gamma], \gamma, \mathbf{v}) &= n(\mathbf{u}, \mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + m_\alpha(\mathbf{u}, \mathbf{v}) + k(\mathbf{u}, p, \mathbf{v}) \\ &\quad - F(\mathbf{v}) \\ R_p(\mathbf{u}[\gamma], q) &= b(\mathbf{u}, q). \end{aligned} \quad (3.1.2)$$

with  $R_u$  and  $R_p$  the residual operators associated to the momentum and continuity equation of NS system, respectively, and

$$k(\mathbf{u}, p, \mathbf{v}) == \int_{\partial\Omega} \left( \frac{1}{\rho} p \mathbf{I} - \nu \nabla \mathbf{u} \right) \mathbf{n} \cdot \mathbf{v} d\sigma. \quad (3.1.3)$$

Since  $\mathbf{u}$  and  $p$  are the (steady) theoretical solutions of the strong formula-

tion (2.4.1),  $R_u$  and  $R_p$  must be exactly zero for every possible choice of  $\mathbf{v} \in [H^1(\Omega)]^d$  and  $q \in L^2(\Omega)$ .

Most of the authors in the literature, (see, e.g., [32]), define now a Lagrangian function, by summing to  $J$  the constraints  $R_u = 0$  and  $R_p = 0$  weighted by appropriate functions  $u_a$  and  $p_a$  (later called adjoint variables). The Karush Kuhn Tacker optimality principles are then used to find the sensitivities of the original functional  $J$ . This approach is extremely solid, but it is not so intuitive for readers having a not so strong mathematical background; therefore a much simpler idea is explained in the following.

Since the residuals (3.1.2) have to be zero for all possible functions  $\gamma \in H^1(\Omega)$ , also

$$\begin{aligned} \frac{dR_u}{d\gamma} &= \frac{\partial R_u}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\gamma} + \frac{\partial R_u}{\partial p} \frac{dp}{d\gamma} + \frac{\partial R_u}{\partial \gamma} = 0 \quad \forall v \in [H^1(\Omega)]^d \\ \frac{dR_p}{d\gamma} &= \frac{\partial R_p}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\gamma} = 0 \quad \forall q \in L^2(\Omega). \end{aligned} \quad (3.1.4)$$

We can thus add those zero quantities to the formulation for functional sensitivity, and collect terms with the same implicit derivative

$$\begin{aligned} \frac{d}{d\gamma} J &= \left[ \frac{\partial J}{\partial \gamma} + \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{\partial \mathbf{u}}{\partial \gamma} + \frac{\partial J}{\partial p} \cdot \frac{\partial p}{\partial \gamma} \right] + \left[ \frac{\partial R_u}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \gamma} + \frac{\partial R_u}{\partial p} \frac{\partial p}{\partial \gamma} + \frac{\partial R_u}{\partial \gamma} \right] + \left[ \frac{\partial R_p}{\partial \mathbf{u}} \frac{\partial \mathbf{u}}{\partial \gamma} \right] \\ &= \frac{\partial J}{\partial \gamma} + \frac{\partial R_u}{\partial \gamma} + \left( \frac{\partial J}{\partial \mathbf{u}} + \frac{\partial R_u}{\partial \mathbf{u}} + \frac{\partial R_p}{\partial \mathbf{u}} \right) \frac{\partial \mathbf{u}}{\partial \gamma} + \left( \frac{\partial J}{\partial p} + \frac{\partial R_u}{\partial p} \right) \frac{\partial p}{\partial \gamma}. \end{aligned} \quad (3.1.5)$$

Since derivatives in (3.1.4) are zero for all test functions  $\mathbf{v} \in H^1(\Omega)$  and  $q \in L^2(\Omega)$ , we search for a pair  $(\mathbf{v}, q)$  (if exist) that send to zero the round

parenthesis multiplied by implicit derivatives  $\partial \mathbf{u} / \partial \gamma$  and  $\partial p / \partial \gamma$ , and hence avoid their computation.

The obtained system is

$$\begin{cases} D_u := \frac{\partial R_u}{\partial \mathbf{u}}(\cdot, \mathbf{v}) + \frac{\partial R_p}{\partial \mathbf{u}}(\cdot, q) + \frac{\partial J}{\partial \mathbf{u}} = 0 \\ D_p := \frac{\partial R_u}{\partial p}(\cdot, q) + \frac{\partial J}{\partial p} = 0, \end{cases} \quad (3.1.6)$$

where "  $\cdot$  " indicates the dependence for variables fixed by the Navier-Stokes equations ( $\mathbf{u}$  and  $p$  are the solution of system (2.4.1), so that  $R_u$ ,  $R_p$  and their derivatives in  $\gamma$  are zero). This system prescribes the conditions of  $D_u = 0$  and  $D_p = 0$ , with 0 the null operator.

If a pair  $(\mathbf{v}, q) = (\mathbf{u}_a, p_a)$  satisfies System (3.1.6), the computation for sensitivity in (3.1.5) reduces to

$$\frac{d}{d\gamma} J = \frac{\partial J}{\partial \gamma}(\mathbf{u}, p, \gamma) + \frac{\partial R_u}{\partial \gamma}(\mathbf{u}, p, \mathbf{u}_a), \quad (3.1.7)$$

ans does not require any of the derivation for the implicit functions.

Given the assumption of Fréchet differentiability these operators can be easily computed via Gateaux derivative. Being  $\mathbf{w}$  and  $r$  the derivative directions, the above system should be valid for every search direction  $\mathbf{w} \in \mathcal{V}$  and  $r \in \mathcal{Q}$  (spaces of  $\mathbf{u}$  and  $p$ , derivation functions).

The Equation (3.1.6) is exactly the base of the adjoint system for the Navier-Stokes equations.

Let now reformulate system (3.1.6) to impose conditions on each directional derivative,

$$\begin{cases} \frac{\partial R_u}{\partial \mathbf{u}}(\cdot, \mathbf{u}_a, \mathbf{w}) + \frac{\partial R_p}{\partial \mathbf{u}}(\cdot, p_a, \mathbf{w}) + \frac{\partial J}{\partial \mathbf{u}}(\mathbf{w}), & \forall \mathbf{w} \in \mathcal{V} \\ \frac{\partial R_u}{\partial p}(\cdot, \mathbf{u}_a, r) = -\frac{\partial J}{\partial p}(r), & \forall r \in \mathcal{Q}. \end{cases} \quad (3.1.8)$$

It can be proved that this is exactly the weak formulation of the problem given by (3.1.6). It should be noted that, although this derivation seems of simpler understanding compared to the one proposed in the literature, it does not allow a full mathematical comprehension of what an "adjoint system" really represent, for which a classical derivation is suggested.

Let us compute now the derivatives of terms in  $R_u$  and  $R_p$ ,

$$\begin{aligned} \frac{\partial a(\mathbf{u}, \mathbf{u}_a)}{\partial \mathbf{u}}(\mathbf{w}) &= \lim_{t \rightarrow 0} \frac{\nu \int_{\Omega} [\nabla(\mathbf{u} + t\mathbf{w})\nabla\mathbf{u}_a - \nabla\mathbf{u}\nabla\mathbf{u}_a] d\mathbf{x}}{t} \\ &= \nu \int_{\Omega} \nabla\mathbf{w}\nabla\mathbf{u}_a d\mathbf{x} = a(\mathbf{w}, \mathbf{u}_a) = a(\mathbf{u}_a, \mathbf{w}) \end{aligned} \quad (3.1.9)$$

$$\begin{aligned} \frac{\partial m_{\alpha}(\mathbf{u}, \mathbf{u}_a)}{\partial \mathbf{u}}(\mathbf{w}) &= \frac{1}{\rho} \lim_{t \rightarrow 0} \frac{\int_{\Omega} \alpha(\mathbf{x}) [(\mathbf{u} + t\mathbf{w})\mathbf{u}_a - \mathbf{u}\mathbf{u}_a] d\mathbf{x}}{t} \\ &= \frac{1}{\rho} \int_{\Omega} \mathbf{w}\mathbf{u}_a d\mathbf{x} = m_{\alpha}(\mathbf{w}, \mathbf{u}_a) = m_{\alpha}(\mathbf{u}_a, \mathbf{w}) \end{aligned} \quad (3.1.10)$$

$$\begin{aligned} \frac{\partial b(\mathbf{u}, p_a)}{\partial \mathbf{u}}(\mathbf{w}) &= -\frac{1}{\rho} \lim_{t \rightarrow 0} \frac{\int_{\Omega} p_a [\nabla \cdot (\mathbf{u} + t\mathbf{w}) - \nabla \cdot \mathbf{u}] d\mathbf{x}}{t} \\ &= -\frac{1}{\rho} \int_{\Omega} p_a \nabla \cdot \mathbf{w} d\mathbf{x} = b(\mathbf{w}, p_a) \end{aligned} \quad (3.1.11)$$

$$\begin{aligned}
\frac{\partial n(\mathbf{u}, \mathbf{u}, \mathbf{u}_a)}{\partial \mathbf{u}}(\mathbf{w}) &= \lim_{t \rightarrow 0} \frac{\int_{\Omega} [(\mathbf{u} + t\mathbf{w}) \cdot \nabla(\mathbf{u} + t\mathbf{w}) - (\mathbf{u} \cdot \nabla \mathbf{u})] \cdot \mathbf{u}_a d\mathbf{x}}{t} \\
&= \int_{\Omega} [\mathbf{w} \cdot \nabla \mathbf{u} + \mathbf{u} \cdot \nabla \mathbf{w}] \cdot \mathbf{u}_a d\mathbf{x} \\
&= \int_{\Omega} (\mathbf{w} \cdot \nabla \mathbf{u}) \cdot \mathbf{u}_a d\mathbf{x} + \int_{\Omega} (\mathbf{u} \cdot \nabla \mathbf{w}) \cdot \mathbf{u}_a d\mathbf{x} \\
&= n(\mathbf{w}, \mathbf{u}, \mathbf{u}_a) + n(\mathbf{u}, \mathbf{w}, \mathbf{u}_a)
\end{aligned} \tag{3.1.12}$$

$$\begin{aligned}
\frac{\partial b(\mathbf{u}_a, p)}{\partial p}(r) &= -\frac{1}{\rho} \lim_{t \rightarrow 0} \frac{\int_{\Omega} [(p + tr)\nabla \cdot \mathbf{u}_a - p\nabla \cdot \mathbf{u}_a] d\mathbf{x}}{t} \\
&= -\frac{1}{\rho} \int_{\Omega} r \nabla \cdot \mathbf{u}_a d\mathbf{x} = b(\mathbf{u}_a, r)
\end{aligned} \tag{3.1.13}$$

Recalling that  $\mathbf{u}$  and  $p$  are the steady solution of (2.4.1), and given that boundary conditions do not change in  $\gamma$ ,  $k$  is actually equivalent to (exclusively for Dirichlet or Neumann boundaries)

$$\begin{aligned}
k(\mathbf{u}, p, \mathbf{v}) &= \int_{\partial\Omega} \left( \frac{1}{\rho} p \mathbf{I} - \nu \nabla \mathbf{u} \right) \mathbf{n} \cdot \mathbf{v} d\sigma \\
&= \int_{\Gamma_N} g_N \mathbf{n} \cdot \mathbf{v} d\sigma + \int_{\Gamma_D} \left( \frac{1}{\rho} p \mathbf{I} - \nu \nabla \mathbf{u}_D \right) \mathbf{n} \cdot \mathbf{v} d\sigma = k(\cdot, p, \mathbf{v}),
\end{aligned} \tag{3.1.14}$$

resulting independent on  $\mathbf{u}[\gamma]$ .

The derivatives of Equation (3.1.14) are then

$$\frac{\partial k(\mathbf{u}, p, \mathbf{u}_a)}{\partial \mathbf{u}}(\mathbf{w}) = 0, \tag{3.1.15}$$

and

$$\begin{aligned} \frac{\partial k(\mathbf{u}, p, \mathbf{u}_a)}{\partial p}(r) &= \frac{1}{\rho} \lim_{t \rightarrow 0} \frac{\int_{\Gamma_D} (p + tr - p) \mathbf{n} \cdot \mathbf{u}_a d\mathbf{x}}{t} \\ &= \frac{1}{\rho} \int_{\Gamma_D} r \mathbf{n} \cdot \mathbf{u}_a d\mathbf{x} = \frac{1}{\rho} \int_{\Gamma_D} \mathbf{u}_a \cdot \mathbf{n} r d\mathbf{x}. \end{aligned} \quad (3.1.16)$$

Substituting the terms in (3.1.9 – 3.1.16) into (3.1.6) the adjoint is

$$\begin{cases} a(\mathbf{u}_a, \mathbf{w}) + n(\mathbf{w}, \mathbf{u}, \mathbf{u}_a) + n(\mathbf{u}, \mathbf{w}, \mathbf{u}_a) + b(\mathbf{w}, p_a) + m_\alpha(\mathbf{u}_a, \mathbf{w}) \\ \quad = -\frac{\partial J}{\partial \mathbf{u}}(\mathbf{w}), & \forall \mathbf{w} \in \mathcal{V} \\ b(\mathbf{u}_a, q) + \frac{\partial k}{\partial p}(r) = -\frac{\partial J}{\partial p}(r), & \forall r \in \mathcal{Q}. \end{cases} \quad (3.1.17)$$

It is evident how the structure in (3.1.17) reminds the structure of the weak formulation for the governing equations. Almost each term, except the convection operator  $n(\mathbf{u}, \mathbf{u}, \mathbf{v})$  has an analogous in the adjoint weak formulation, where  $\mathbf{u}_a$  and  $p_a$  are the new unknowns, and  $\mathbf{w}$ ,  $r$  are the new test functions. The only difference in the adjoint system relies on the non linear convection operator  $n(\mathbf{u}, \mathbf{u}, \mathbf{v})$ , that here is converted in two linear operators  $n(\mathbf{w}, \mathbf{u}, \mathbf{u}_a) + n(\mathbf{u}, \mathbf{w}, \mathbf{u}_a)$ . This difference is however of extreme relevance for the numerical analysis, since the lack of non linear terms allows a direct solving of the adjoint problem. For this reason, the computational effort to solve this system is considerably small compared to the numerical costs of the governing equations.

Finally, the forcing term, compressibility, and boundary conditions are governed by the functional derivatives in  $\mathbf{u}$  and  $p$ . Let us consider a stationary

functional as (2.2.4),

$$J = \beta_1 \int_{\Omega} A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma) d\mathbf{x} + \beta_2 \int_{\partial\Omega} B(\mathbf{u}, p; \gamma) d\sigma,$$

and compute derivatives in  $\mathbf{u}$  and  $p$ .

Since  $\mathbf{u}$  and  $p$  are solutions of the steady NS system, and the boundary conditions are fixed for all values of  $\gamma$ , it is possible to enforce them as done for  $k(\mathbf{u}, p, \mathbf{v})$  term above, obtaining

$$\begin{aligned} J &= \beta_1 \int_{\Omega} A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma) d\mathbf{x} + \beta_2 \int_{\Gamma_N} B(\mathbf{u}, p; \gamma) d\sigma + \beta_2 \int_{\Gamma_D} B(\mathbf{u}, p; \gamma) d\sigma \\ &= \beta_1 \int_{\Omega} A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma) d\mathbf{x} + \beta_2 \int_{\Gamma_N} B(\mathbf{u}, \nabla \mathbf{u}; \gamma) d\sigma + \beta_2 \int_{\Gamma_D} B(\mathbf{u}_D, p; \gamma) d\sigma. \end{aligned} \quad (3.1.18)$$

In (3.1.18), by the Neumann conditions,  $p$  can be written as a function of  $g_N$  and  $\nabla \mathbf{u}$  on  $\Gamma_N$ . Derivative in  $\mathbf{u}$  is then

$$\begin{aligned} \frac{\partial J}{\partial \mathbf{u}}(\cdot, \mathbf{w}) &= \lim_{t \rightarrow 0} \frac{1}{t} \beta_1 \int_{\Omega} [A(\mathbf{u} + t\mathbf{w}, \nabla(\mathbf{u} + t\mathbf{w}), p; \gamma) - A(\mathbf{u}, \nabla \mathbf{u}, p; \gamma)] d\mathbf{x} \\ &\quad + \lim_{t \rightarrow 0} \frac{1}{t} \beta_2 \int_{\Gamma_N} [B(\mathbf{u} + t\mathbf{w}, \nabla(\mathbf{u} + t\mathbf{w}); \gamma) - B(\mathbf{u}, \nabla \mathbf{u}; \gamma)] d\sigma. \end{aligned}$$

Further, adding and subtracting  $A(\mathbf{u}, \nabla(\mathbf{u} + t\mathbf{w}), p; \gamma)$  in the first limit, and  $B(\mathbf{u}, \nabla(\mathbf{u} + t\mathbf{w}))$  in the second, the derivative reads

$$\frac{\partial J}{\partial \mathbf{u}}(\cdot, \mathbf{w}) = \beta_1 \int_{\Omega} \left[ \frac{\partial A}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial A}{\partial \nabla \mathbf{u}}(\mathbf{w}) \right] d\mathbf{x} + \beta_2 \int_{\Gamma_N} \left[ \frac{\partial B}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial B}{\partial \nabla \mathbf{u}}(\mathbf{w}) \right] d\sigma. \quad (3.1.19)$$

Similarly, the derivative of  $J$  in  $p$ , reads

$$\begin{aligned} \frac{\partial J}{\partial p}(\cdot, r) &= \lim_{t \rightarrow 0} \frac{1}{t} \beta_1 \int_{\Omega} [A(\mathbf{u}, \nabla \mathbf{u}, p + tr; \gamma) - A(\mathbf{u}, \nabla \mathbf{u}, p, \gamma)] d\mathbf{x} \\ &\quad + \lim_{t \rightarrow 0} \frac{1}{t} \beta_2 \int_{\Gamma_D} [B(\cdot, p + tr, \gamma) - B(\mathbf{u}, p, \gamma)] d\sigma \\ &= \beta_1 \int_{\Omega} \frac{\partial A}{\partial p}(r) d\mathbf{x} + \beta_2 \int_{\Gamma_D} \frac{\partial B}{\partial p}(r) d\sigma \end{aligned} \quad (3.1.20)$$

Substituting (3.1.19) and (3.1.20) in (3.1.17) the adjoint system weak formulation results in

$$\left\{ \begin{array}{l} a(\mathbf{u}_a, \mathbf{w}) + n(\mathbf{w}, \mathbf{u}, \mathbf{u}_a) + n(\mathbf{u}, \mathbf{w}, \mathbf{u}_a) + b(\mathbf{w}, p_a) + m_\alpha(\mathbf{u}_a, \mathbf{w}) \\ \quad = -\beta_1 \int_{\Omega} \left[ \frac{\partial A}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial A}{\partial \nabla \mathbf{u}}(\mathbf{w}) \right] d\mathbf{x} \\ \quad \quad - \beta_2 \int_{\Gamma_N} \left[ \frac{\partial B}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial B}{\partial \nabla \mathbf{u}}(\mathbf{w}) \right] d\sigma, \\ b(\mathbf{u}_a, q) + \frac{1}{\rho} \int_{\Gamma_D} \mathbf{u}_a \cdot \mathbf{n} r d\mathbf{x} = -\beta_1 \int_{\Omega} \frac{\partial A}{\partial p}(r) d\mathbf{x} - \beta_2 \int_{\Gamma_D} \frac{\partial B}{\partial p}(r) d\sigma, \end{array} \right. \quad (3.1.21)$$

for all  $\mathbf{w} \in \mathcal{V}$  and  $r \in \mathcal{Q}$ . The boundary conditions for the adjoint system are defined from the boundary terms of (3.1.21). The Neumann BCs naturally arise from the weak formulation by the integral on the Neumann boundary  $\Gamma_N$ , i.e., in this case they are already applied, being  $\beta_2 \int_{\Gamma_N} \left[ \frac{\partial B}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial B}{\partial \nabla \mathbf{u}}(\mathbf{w}) \right] d\sigma$  in (3.1.21).

The Dirichlet boundary conditions are instead obtained by comparing the two integrals on the Dirichlet boundary,  $\Gamma_D$  in (3.1.21), obtaining

$$\int_{\Gamma_D} \mathbf{u}_a \cdot \mathbf{n} r d\sigma = -\rho \beta_2 \int_{\Gamma_D} \frac{\partial B}{\partial p}(r) d\sigma. \quad (3.1.22)$$

For example if  $B = p$ ,

$$\int_{\Gamma_D} \mathbf{u}_a \cdot \mathbf{n} r d\sigma = -\rho \beta_2 \int_{\Gamma_D} 1 r d\sigma \implies \mathbf{u}_a \cdot \mathbf{n} = -\rho \beta_2.$$

Since  $\mathbf{w}$  is contained in the space  $\mathcal{V}$ , hence vanishes at Dirichlet boundaries, it is mandatory to have essential equation for the value of  $\mathbf{u}_a$  on  $\Gamma_D$ . Given that no other condition is originated from (3.1.21) for the tangential components of  $\mathbf{u}_a$ , the simplest choice adopted in literature is to set (3.1.22) as

$$\int_{\Gamma_D} \mathbf{u}_a r d\sigma = -\rho\beta_2 \int_{\Gamma_D} \frac{\partial B}{\partial p}(r) \mathbf{n} d\sigma. \quad (3.1.23)$$

In above example ( $B = p$ ), this implies  $\mathbf{u}_D = -\rho\beta_2 \mathbf{n}$ .

From system (3.1.21), with some elementary calculation, it is possible to derive the operators  $D_u$  and  $D_p$  of (3.1.6), and from (3.1.6) obtain the strong formulation of the adjoint system. The result is the following [16]:

$$\left\{ \begin{array}{l} -\nu\Delta\mathbf{u}_a - (\mathbf{u} \cdot \nabla)\mathbf{u}_a + (\nabla\mathbf{u})\mathbf{u}_a + \frac{1}{\rho}\nabla p_a \\ \qquad \qquad \qquad = -\beta_1 \left( \frac{\partial A'}{\partial \mathbf{u}} - \nabla \cdot \frac{\partial A'}{\partial \nabla \mathbf{u}} \right) - \frac{1}{\rho}\alpha\mathbf{u}_a, \text{ in } \Omega \\ -\frac{1}{\rho}\nabla \cdot \mathbf{u}_a = -\beta_1 \frac{\partial A}{\partial p}, \text{ in } \Omega \\ \mathbf{u}_a = -\rho \frac{\partial B'}{\partial p} \mathbf{n}, \text{ on } \Gamma_D \\ \left[ -\frac{1}{\rho}p_a \mathbf{I} + \nu\nabla\mathbf{u}_a \right] \mathbf{n} = -(\mathbf{u} \cdot \mathbf{n})\mathbf{u}_a - \beta_1 \frac{\partial A}{\partial \nabla \mathbf{u}} \mathbf{n} - \beta_2 \frac{\partial B}{\partial \mathbf{u}}, \text{ on } \Gamma_N, \end{array} \right. \quad (3.1.24)$$

where for simplicity of notation we have assumed  $A$  and  $B$  such that  $\frac{\partial A}{\partial \mathbf{u}}(\mathbf{w}) = \frac{\partial A'}{\partial \mathbf{u}} \mathbf{w}$ ,  $\frac{\partial A}{\partial \nabla \mathbf{u}}(\mathbf{w}) = \frac{\partial A'}{\partial \nabla \mathbf{u}} \mathbf{w}$  and  $\frac{\partial B}{\partial p}(r) = \frac{\partial B'}{\partial p} r$ .

The system can be thus solved with a FEM algorithm, similarly to the Navier-Stokes system (2.4.1).

Once solved the system (3.1.21), the solution  $(\mathbf{u}_a, p_a)$  is known, and from (3.1.7) the sensitivities can be found as

$$\frac{d}{d\gamma} J(\mathbf{u}, p, \gamma) = \frac{\partial J}{\partial \gamma}(\mathbf{u}, p, \gamma) + \frac{\partial R_u}{\partial \gamma}(\mathbf{u}, p, \gamma, \mathbf{u}_a),$$

where, by the Gateaux derivatives in a generic direction  $h \in L^2(\Omega)$ ,

$$\begin{aligned} \frac{\partial J}{\partial \gamma}(\cdot, h) &= \beta_1 \int_{\Omega} \frac{\partial A}{\partial \gamma}(h) d\mathbf{x} + \beta_2 \int_{\Gamma_N} \frac{\partial B}{\partial \gamma}(h) d\sigma \\ &= \beta_1 \int_{\Omega} \frac{\partial A'}{\partial \gamma} h d\mathbf{x} + \beta_2 \int_{\Gamma_N} \frac{\partial B'}{\partial \gamma} h d\sigma. \end{aligned} \quad (3.1.25)$$

$$\begin{aligned} \frac{\partial R_u}{\partial \gamma}(\cdot, h) &= \frac{\partial m_\alpha}{\partial \gamma} = \frac{1}{\rho} \int_{\Omega} \frac{\partial \alpha}{\partial \gamma}(h) \mathbf{u} \cdot \mathbf{u}_a d\mathbf{x} \\ &= \frac{1}{\rho} \int_{\Omega} \frac{\partial \alpha'}{\partial \gamma} \mathbf{u} \cdot \mathbf{u}_a h d\mathbf{x}, \end{aligned} \quad (3.1.26)$$

Summing the two the result, the derivative of  $J$  in  $\gamma$  is simply

$$\begin{aligned} \frac{d}{d\gamma} J(\mathbf{u}, p, \gamma)(h) &= \int_{\Omega} \left[ \beta_1 \frac{\partial A}{\partial \gamma} + \frac{\partial \alpha'}{\partial \gamma} \mathbf{u} \cdot \mathbf{u}_a \right] h d\mathbf{x} + \int_{\partial\Omega} \beta_2 \frac{\partial B'}{\partial \gamma} h d\sigma \\ &= \langle \beta_1 \frac{\partial A}{\partial \gamma} + \frac{\partial \alpha'}{\partial \gamma} \mathbf{u} \cdot \mathbf{u}_a, h \rangle_{L^2(\Omega)} + \langle \beta_2 \frac{\partial B'}{\partial \gamma}, h \rangle_{L^2(\partial\Omega)}, \end{aligned} \quad (3.1.27)$$

with  $\langle \cdot, \cdot \rangle_{L^2(\Omega)}$   $L^2$  scalar product in  $\Omega$  and  $\langle \cdot, \cdot \rangle_{L^2(\partial\Omega)}$   $L^2$  scalar product in  $\partial\Omega$ . This finally implies

$$\frac{d}{d\gamma} J(\mathbf{u}, p, \gamma) = \langle \beta_1 \frac{\partial A}{\partial \gamma} + \frac{\partial \alpha'}{\partial \gamma} \mathbf{u} \cdot \mathbf{u}_a, \cdot \rangle_{L^2(\Omega)} + \langle \beta_2 \frac{\partial B'}{\partial \gamma}, \cdot \rangle_{L^2(\partial\Omega)} \quad (3.1.28)$$

### The Derivation for the time dependent scenario

In the time dependent scenario, the objective functional  $J$  is given by (2.2.5).

Let  $R_u$  and  $R_p$  be the total residual of the Navier-Stokes equations in time

$[0, T]$ , i.e.,

$$\begin{aligned}
R_u(\mathbf{u}[\gamma], p[\gamma], \gamma, \mathbf{v}) &= \int_0^T [(\partial_t \mathbf{u}, \mathbf{v}) + n(\mathbf{u}, \mathbf{u}, \mathbf{v}) + a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) + m_\alpha(\mathbf{u}, \mathbf{v}) \\
&\quad + k(\mathbf{u}, p, \mathbf{v}) - F(\mathbf{v})] dt + \int_\Omega [\mathbf{u} - \mathbf{u}_0] \Big|_{t=0} \mathbf{v} d\mathbf{x} \\
R_p(\mathbf{u}[\gamma], q) &= \int_0^T b(\mathbf{u}, q) dt,
\end{aligned} \tag{3.1.29}$$

where the initial condition is  $\mathbf{u} = \mathbf{u}_0$  at  $t = 0$ .

Steps up to (3.1.8) are valid also in this case, even if now the target is a map  $\mathbf{u}_a(t) : \mathcal{I} \rightarrow [H^1(\Omega)]^d$  and  $p_a(t) : \mathcal{I} \rightarrow \mathcal{Q}$ , i.e., solutions  $\mathbf{u}_a \in L^2(\mathcal{I}, [H^1(\Omega)]^d)$  and  $p_a \in L^2(\mathcal{I}, \mathcal{Q})$ . Derivatives (3.1.9) and (3.1.20) are thus consistent, apart from an additional integration in time in  $[0, T]$ .

The latter yields a new time derivative term appearing in  $R_u$ ,  $(\partial_t \mathbf{u}, \mathbf{v})$  and the new initial condition term  $q = \int_\Omega [u - u_0] d\mathbf{x} \Big|_{t=0}$ , resulting

$$\begin{aligned}
\frac{\partial q}{\partial \mathbf{u}}(\mathbf{w}) &= \lim_{t \rightarrow 0} \frac{\int_0^T \int_\Omega [(\mathbf{u} + t\mathbf{w})\mathbf{u}_a - \mathbf{u}\mathbf{u}_a] \Big|_{t=0} d\mathbf{x}}{t} \\
&= \int_0^T \int_\Omega [\mathbf{w}\mathbf{u}_a] \Big|_{t=0} d\mathbf{x} \\
\frac{\partial(\partial_t \mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}(\mathbf{w}) &= \lim_{t \rightarrow 0} \frac{\int_0^T \int_\Omega [\partial_t(\mathbf{u} + t\mathbf{w})\mathbf{u}_a - (\partial_t \mathbf{u})\mathbf{u}_a] d\mathbf{x}}{t} \\
&= \int_0^T \int_\Omega (\partial_t \mathbf{w})\mathbf{u}_a d\mathbf{x}
\end{aligned}$$

On the assumption required by the Fubini-Tonelli theorem [33], it is possible to switch the two integrations in (3.1.1) and apply integration by parts, i.e.,

$$\begin{aligned}
\frac{\partial(\partial_t \mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}(\mathbf{w}) &= \int_\Omega \int_0^T \partial_t(\mathbf{w}\mathbf{u}_a) d\mathbf{x} - \int_\Omega \int_0^T \mathbf{w} \partial_t(\mathbf{u}_a) d\mathbf{x} \\
&= \int_\Omega [\partial_t(\mathbf{w}\mathbf{u}_a)]_0^T d\mathbf{x} - \int_0^T \int_\Omega \mathbf{w} \partial_t(\mathbf{u}_a) d\mathbf{x}.
\end{aligned}$$

Hence, the sum of the two terms yields

$$\frac{\partial q}{\partial \mathbf{u}}(\mathbf{w}) + \frac{\partial(\partial_t \mathbf{u}, \mathbf{v})}{\partial \mathbf{u}}(\mathbf{w}) = \int_{\Omega} [\mathbf{w} \mathbf{u}_a] \Big|_{t=T} d\mathbf{x} - (\partial \mathbf{u}_a, w). \quad (3.1.30)$$

This result for  $t = T$  has exactly the same form of  $\int_{\Omega} [\mathbf{w}(\mathbf{u}_a - \mathbf{u}_{a,T})] \Big|_{t=T} d\mathbf{x}$ , with  $\mathbf{u}_{a,T} = 0$ , and can be easily linked to a strong formulation, where we impose the final condition instead than the initial one. As for the steady case, it is possible to deduce from an hypothetical strong formulation of the problem.

The result is [16]:

$$\left\{ \begin{array}{l} -\frac{\partial \mathbf{u}_a}{\partial t} - \nu \Delta \mathbf{u}_a - (\mathbf{u} \cdot \nabla) \mathbf{u}_a + (\nabla \mathbf{u}) \cdot \mathbf{u}_a + \frac{1}{\rho} \nabla p_a \\ \qquad \qquad \qquad = -\beta_1 \left( \frac{\partial A'}{\partial \mathbf{u}} - \nabla \cdot \frac{\partial A'}{\partial \nabla \mathbf{u}} \right) + \frac{1}{\rho} \alpha \mathbf{u}_a, \text{ in } [0, T] \times \Omega \\ -\frac{1}{\rho} \nabla \cdot \mathbf{u}_a = -\beta_1 \frac{\partial A'}{\partial p}, \text{ in } [0, T] \times \Omega \\ \mathbf{u}_a(T, \mathbf{x}) = 0, \text{ in } \Omega \\ \mathbf{u}_a = -\beta_2 \rho \frac{\partial B'}{\partial p} \mathbf{n}, \text{ on } \Gamma_D \\ \left[ -\frac{1}{\rho} p_a \mathbf{I} + \nu \nabla \mathbf{u}_a \right] \mathbf{n} = -(\mathbf{u} \cdot \mathbf{n}) \mathbf{u}_a - \beta_1 \frac{\partial A'}{\partial \nabla \mathbf{u}} \mathbf{n} - \beta_2 \frac{\partial B'}{\partial \mathbf{u}}, \text{ on } \Gamma_N. \end{array} \right. \quad (3.1.31)$$

Finally, it can be easily proved that for the unsteady scenario

$$\frac{d}{d\gamma} J = \int_0^T \left[ \left\langle \beta_1 \frac{\partial A}{\partial \gamma} + \frac{\partial \alpha'}{\partial \gamma} \mathbf{u} \cdot \mathbf{u}_a, \cdot \right\rangle_{L^2(\Omega)} + \left\langle \beta_2 \frac{\partial B'}{\partial \gamma}, \cdot \right\rangle_{L^2(\partial\Omega)} \right] dt \quad (3.1.32)$$

The adjoint system is modeled with a FEM approach, analogous to the one for Navier-Stokes equations. Real sensitivities are then the derivatives with respect to the local parameters  $\gamma_i$ , which are obtained by the chain rule from (3.1.28) or (3.1.32). In both cases,  $\gamma = \sum_i \gamma_i \phi_i$ , with  $i$  nodes of the mesh and  $\phi_i$  the relative shape functions.

Considering the power dissipation scenario,

$$J = \int_{\Omega} \left[ \frac{1}{2} \mu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^T\|^2 + \alpha \|\mathbf{u}\|^2 \right] d\mathbf{x}, \quad (3.1.33)$$

it can be easily proved that

$$\begin{aligned} \frac{\partial A}{\partial \mathbf{u}} &= 2\alpha \mathbf{u}, \\ \frac{\partial A}{\partial \nabla \mathbf{u}} &= 2 [\nabla u + (\nabla \mathbf{u})^T], \\ \frac{\partial A}{\partial p} &= 0, \\ \frac{\partial B}{\partial \mathbf{u}} &= 0, \\ \frac{\partial B}{\partial p} &= 0. \end{aligned}$$

Hence, the final implemented adjoint system takes the form

$$\left\{ \begin{array}{l} -\frac{\partial \mathbf{u}_a}{\partial t} - \nu \Delta \mathbf{u}_a - (\mathbf{u} \cdot \nabla) \mathbf{u}_a + (\nabla \mathbf{u}) \cdot \mathbf{u}_a + \frac{1}{\rho} \nabla p_a \\ \quad = -\beta_1 \left( 2\alpha \mathbf{u} - 2\nabla \cdot [\nabla u + (\nabla \mathbf{u})^T] \right) + \frac{1}{\rho} \alpha \mathbf{u}_a, \text{ in } [0, T] \times \Omega \\ -\frac{1}{\rho} \nabla \cdot \mathbf{u}_a = 0, \text{ in } [0, T] \times \Omega \\ \mathbf{u}_a(T, \mathbf{x}) = 0, \text{ in } \Omega \\ \mathbf{u}_a = 0, \text{ on } \Gamma_D \\ \left[ -\frac{1}{\rho} p_a \mathbf{I} + \nu \nabla \mathbf{u}_a \right] \mathbf{n} = -(\mathbf{u} \cdot \mathbf{n}) \mathbf{u}_a - 2\beta_1 [\nabla u + (\nabla \mathbf{u})^T] \mathbf{n}, \text{ on } \Gamma_N. \end{array} \right. \quad (3.1.34)$$

### 3.1.2 The Adjoint system for the Advection-Diffusion-Reaction module

Consider for the derivation a steady functional with separable dependence on the physics-governing parameters,  $\mathbf{u}$ ,  $p$ ,  $c$ , in the form

$$J(\mathbf{u}, p, c; \gamma) = \beta_1 J_1(\mathbf{u}, p, \gamma) + \beta_2 J_2(c), \quad (3.1.35)$$

as the functional for the pollute dissipation coupled with power dissipation (2.3.5). Extension to more complex functionals can be easily deduced from the following steps.

By chain rule, the objective functional sensitivity in  $\gamma$  is

$$\frac{d}{d\gamma}[J(\mathbf{u}, p, c; \gamma)] = \frac{\partial J}{\partial \gamma} + \frac{\partial J}{\partial \mathbf{u}} \cdot \frac{d\mathbf{u}}{d\gamma} + \frac{\partial J}{\partial p} \frac{dp}{d\gamma} + \frac{\partial J}{\partial c} \frac{dc}{d\gamma}. \quad (3.1.36)$$

Again, the implicit dependence of  $u$ ,  $p$ , and  $c$  for  $\gamma$  will be handled using appropriate adjoint problems.

Consider the NS weak residual operators,  $R_u$  and  $R_p$  as in (3.1.2) and the residual for the weak ADR problem for a test function  $v \in H_{\Gamma_D}^1(\Omega)$  (from 2.4.10)

$$R_c(c[\gamma], \mathbf{u}[\gamma]v) = a(c, v) + b(c, \mathbf{u}, v) + m_k(c, v) - (g_N, v)_{\Gamma_N} - F(v). \quad (3.1.37)$$

Let us assume the specie diffusion coefficient,  $D$ , to be independent from  $\gamma$ , i.e., to be the perfectly equal between solid and fluid regions. This assumption is clearly a huge simplification of the optimization task, but allows an extremely simpler solving process. In order to reduce the approximation's given errors, the obtained results in chapter 4 are truly obtained with a very

low diffusion coefficient, hence minimizing the diffusion both for the solid and fluid regions, for an advection-dominated flow.

As for the simple fluid-flow case, all these residuals have to be zero for all the possible formulations of the design function  $\gamma$ , so  $R_u$  and  $R_p$  must satisfy (3.1.4), and

$$\frac{dR_c}{d\gamma} = \frac{\partial R_c}{\partial \mathbf{u}} \frac{d\mathbf{u}}{d\gamma} + \frac{\partial R_c}{\partial c} \frac{dc}{d\gamma} = 0, \quad \forall v \in H^1(\Omega). \quad (3.1.38)$$

Summing to (3.1.36) the zero quantities, the sensitivity results in

$$\begin{aligned} \frac{d}{d\gamma} J &= \frac{\partial J}{\partial \gamma} + \frac{\partial R_u}{\partial \gamma} + \left( \frac{\partial J}{\partial \mathbf{u}} + \frac{\partial R_u}{\partial \mathbf{u}} + \frac{\partial R_p}{\partial \mathbf{u}} + \frac{\partial R_c}{\partial \mathbf{u}} \right) \frac{d\mathbf{u}}{d\gamma} + \left( \frac{\partial J}{\partial p} + \frac{\partial R_u}{\partial p} \right) \frac{dp}{d\gamma} \\ &\quad + \left( \frac{\partial J}{\partial c} + \frac{\partial R_c}{\partial c} \right) \frac{dc}{d\gamma}. \end{aligned} \quad (3.1.39)$$

Consider now the last round parenthesis. Since the derivative (3.1.38) is zero for all  $v \in H_{\Gamma_D}^1(\Omega)$ , the idea is thus again to search for the test function  $v$  which send such term to zero, leading to the equation

$$D_c := \frac{\partial R_c}{\partial c}(\cdot, v) + \frac{\partial J}{\partial c} = 0, \quad (3.1.40)$$

where ” $\cdot$ ” hides the implicit dependence for variables fixed by NS and ADR equations. Define  $c_a$  the solution of such equation. Computing directional derivatives as described in the previous section,  $\frac{\partial a(c, c_a)}{\partial c}(w) = a(c_a, w)$ ,

$\partial m_k(c, c_a)/\partial c(w) = m_k(c_a, w)$ , and

$$\begin{aligned}
\frac{\partial b(c, c_a)}{\partial c}(w) &= \lim_{t \rightarrow 0} \frac{\int_{\Omega} \mathbf{u} \cdot \nabla(c + tw - c)c_a d\mathbf{x}}{t} = \int_{\Omega} (\mathbf{u} \cdot \nabla w)c_a d\mathbf{x} \\
&= \int_{\Omega} \operatorname{div}(\mathbf{u}w c_a) d\mathbf{x} - \int_{\Omega} \operatorname{div}(\mathbf{u}c_a)w d\mathbf{x} \\
&= \int_{\partial\Omega} (c_a \mathbf{u} \cdot \mathbf{n})w d\sigma - \int_{\Omega} (\mathbf{u} \cdot c_a)w d\mathbf{x} = \int_{\Omega} (\mathbf{u} \cdot c_a)w d\mathbf{x} \\
&= -b(c_a, w),
\end{aligned} \tag{3.1.41}$$

by divergence theorem and fluid incompressibility ( $\nabla \cdot \mathbf{u} = 0$ ). Equation (3.1.40) has to be valid for all search derivatives  $\mathbf{w}$ , so the advection-diffusion-reaction adjoint weak formulation is

$$a(c_a, w) - b(c_a, \mathbf{u}, w) + m_k(c_a, w) = -\frac{\partial J}{\partial c}, \quad \forall \mathbf{w} \in H_{\Gamma_D}^1(\Omega), \tag{3.1.42}$$

If  $J_2(c) = \int_{\Gamma_{out}} c d\sigma$ ,

$$\frac{\partial J(c)}{\partial c}(w) = \lim_{t \rightarrow 0} \frac{\int_{\Gamma_{out}} (c + tw - c) d\sigma}{t} = \int_{\Gamma_{out}} w d\sigma, \tag{3.1.43}$$

the weak formulation (3.1.42) can be easily associated to a continuous scalar transport problem in the form

$$\begin{cases} -\mathbf{u} \cdot \nabla c_a - \nabla \cdot (D \nabla c_a) = 0 & \text{in } \Omega \\ c_a(\mathbf{x}) = 0, & \text{on } \Gamma_D \\ D \nabla c \cdot \mathbf{n} = -\beta_2, & \text{on } \Gamma_N. \end{cases} \tag{3.1.44}$$

Once the adjoint concentration,  $c_a$ , is determined, an adjoint NS problem has to be developed to kill the other round parenthesis in (3.1.39). Let us first



## 3.2 The Optimization schemes

Here are summarized the main features of each of the chosen optimization schemes.

### 3.2.1 The Optimality Criteria (OC)

The Optimality Criteria is a very simple scheme widely used for TO in structural mechanics. It is based on the Karush-Kuhn-Tacker optimality conditions, and has the limitation of admitting just one equality constraint in the formulation. By (2.2.1), we have

$$\left\{ \begin{array}{l} \min_{\gamma} J(\gamma) \\ \text{subject to} \quad : g(\gamma) = 0 \quad \text{Equality constraint,} \\ \quad \quad \quad : R(\gamma) = 0 \quad \text{Governing equations,} \\ \quad \quad \quad : \gamma_{min} \leq \gamma \leq \gamma_{max}, \quad \text{Design variable bounds.} \end{array} \right. \quad (3.2.1)$$

It is now possible convert this constrained optimization problem into an unconstrained one by defining the Lagrange function :

$$L(\gamma, \lambda) = J(\gamma) + \lambda(g(\gamma)). \quad (3.2.2)$$

The Karush-Kuhn-Tucker first-order optimality conditions state that

$$\left\{ \begin{array}{l} \frac{\partial L}{\partial \gamma} = \frac{\partial J}{\partial \gamma} + \lambda \frac{\partial g(\gamma)}{\partial \gamma} = 0 \\ \frac{\partial L}{\partial \lambda} = g(\gamma) = 0. \end{array} \right. \quad (3.2.3)$$

Since the Lagrange multiplier  $\lambda$  and the design variable  $\gamma_i$  are coupled, the two equations must be solved simultaneously, but, being nonlinear equations, the solution is computationally expensive and difficult due to numerical instabilities. The common solving procedure is composed of two-level loops. In the inner loop, given  $\lambda$ , the design variable  $\gamma_i$  is updated to satisfy the first condition of (3.2.3). On the other hand, the outer loop is used to update the value of  $\lambda$  in order to satisfy the constraint.

Once defined the element scale factor

$$D_i = \frac{|\frac{\partial J}{\partial \gamma_i}|}{\lambda \frac{\partial V}{\partial \gamma_i}}, \quad (3.2.4)$$

the inner-loop update rule for the  $\gamma_i$  is

$$\gamma_i^{new} = \gamma_i^{old} \sqrt{D_i}, \quad \gamma_i^{min} \leq \gamma_i^{new} \leq \gamma_i^{max}. \quad (3.2.5)$$

One can easily verify that with this choice the first equation in (3.2.3) is verified. In Equation (3.2.3), note how the optimality conditions are satisfied when the objective sensitivity has same absolute value of the constraint sensitivity times  $\lambda$ . Thus, in an optimal configuration the ratio of the two quantities ( $D_i$ ) should be one. Therefore, the design does not change when  $D_i = 1$  and the optimality condition is already satisfied. When  $D_i < 1$ , increasing the material density  $\gamma_i$  (that is decreasing its porosity) has a greater effect on the constraint bond than in decreasing the compliance, so its value decreases. On the opposite, when  $D_i > 1$ , increasing  $\gamma_i$  becomes convenient since it produces an higher improvement in the functional than in the constraint. To prevent the scheme from instabilities due to hasty shifts, we impose a maximum value for the change of each variable.

Consequently, in the outer loop the Lagrange multiplier is updated to satisfy the constraint  $g(\gamma) = 0$  using a simple bisection method. Starting from some

given lower and upper bound for  $\lambda$ ,  $\lambda_{min}$  and  $\lambda_{max}$ ], the range  $[\lambda_{min}, \lambda_{max}]$  is halved at each iteration of the outer loop, and the incumbent  $\lambda$  is chosen as the middle value of the range. If the procedure is to evaluate the volume constraint,  $V(\gamma) - V_r V_0$ , when the result is positive, the upper half of the range is kept and the lower is discarded (greater values of  $\lambda$  results in smaller  $D_i$  and probably smaller densities, hence smaller volume). The opposite is done when  $V(\gamma) - V_r V_0$  is negative, and this routine is repeated until the range is smaller than a given tolerance.

As said above, this method is very fast and simple to implement but it can not be applied to more complex scenarios, like the multi-constrained problems. Such concern will be overcome in the generalized version of the OC method, based on a different approach for the  $\gamma - \lambda$  coupled problem.

### 3.2.2 The Generalized Optimality Criteria (GOC)

The generalized Optimality Criteria Method (GOCM) for topology optimization [9] extends the capability of the OCM to multiple inequality constraints, although it may lead to a loss of the efficiency. The general optimization problem solvable with this technique can be defined as

$$\left\{ \begin{array}{l} \min_{\gamma} J(\gamma) \\ \text{subject to} : g_i(\gamma) \leq 0, \quad i = 1, \dots, NC, \\ \quad \quad \quad : R(\gamma) = 0, \quad \text{Governing equations,} \\ \quad \quad \quad : \gamma_{min} \leq \gamma \leq \gamma_{max}, \quad \text{Design variable bounds,} \end{array} \right. \quad (3.2.6)$$

with  $NC$  number of constraints. This formulation can clearly represent also greater-than-or-equal-to inequalities constraint and multiple functionals by weighted sums.

The associated Lagrangian unconstrained optimization problem has now the form

$$\text{minimize } L(\gamma, \lambda, s) = J(\gamma) + \sum_{i=1}^{NC} \lambda_i (g_u(\gamma) + s_i^2), \quad (3.2.7)$$

with  $\lambda_i$  the Lagrange multiplier associated to constraint  $g_i$ . The variables  $s_i$ , called "slack variables", are used to convert the inequalities constraint into equalities, and are not zero (less than zero) only when the constrain is satisfied ( $g_i < 0$ ). The necessary conditions for optimality are

$$\begin{aligned} \frac{\partial J}{\partial \gamma} + \sum_{i=1}^{NC} \lambda_i \frac{\partial g_i}{\partial \gamma} &= 0, \\ g_i(\gamma) + s_i^2 &= 0, \quad i = 1, \dots, NC \\ \lambda_i s_i &= 0, \quad i = 1, \dots, NC. \end{aligned} \quad (3.2.8)$$

The third set of equations in (3.2.8) represents the "complementary slackness conditions", which state that only active constraints need to be considered in the necessary conditions ( if  $s_i \neq 0$  the constraint is satisfied, and  $\lambda_i = 0$ , so that it does not influence the optimization).

Conversely, the second and third sets of equations in (3.2.8) are satisfied by identifying active constraints. For this reason the GOC goal is only to solve the first part of (3.2.8). As in (3.2.3), the Lagrange multipliers  $\lambda_i$  and the decision variables  $\gamma_i$  are coupled, but, in this case, the double-loop approach of OC method can not be exploited, since it would require an additional loop for each new constraint.

The Generalized Optimality Criteria overcomes this problem by relaxing the

controls over the Lagrange multipliers, stating that they do not have to satisfy Equations (3.2.8) in each iteration. It then introduces an update routine to assure gradually the constraints as the optimization algorithm converges.

More details on the method and on the possible updating routines for the Lagrange multipliers and the design parameters are given in [9].

In this project we use

$$\lambda_i^{k+1} = \lambda_i^k [1 + p_0(g_i^k + \Delta g_i^k)] \quad (3.2.9)$$

as update rule for the Lagrange multipliers, with  $k$  the iteration number of the optimization problem,  $\Delta g_i$  the variation on the value of  $i_{th}$  constraint from the previous iteration, and  $p_0$  the chosen update parameter (usually  $p_0 = 0.5$  or 1). For the decision variables we still to use a scale factor  $D_{iel}$  for the updating algorithm, now redefined as

$$D_{iel} = -\frac{\left\langle \frac{\partial J}{\partial \gamma_{iel}} \right\rangle_- + \sum_{i=1}^{NC} \lambda_i \left\langle \frac{\partial g_i}{\partial \gamma_{iel}} \right\rangle_-}{\left\langle \frac{\partial J}{\partial \gamma_{iel}} \right\rangle_+ + \sum_{i=1}^{NC} \lambda_i \left\langle \frac{\partial g_i}{\partial \gamma_{iel}} \right\rangle_+} \quad (3.2.10)$$

where  $\langle x \rangle_- = \min(0, a)$  and  $\langle x \rangle_+ = \max(0, a)$ . When  $D_{iel} = 1$ , this formula also satisfies the stationary condition of the Lagrange function.

From Lagrange multiplier theory, it is well known that  $\lambda_i = 0$  only if the relative constraint,  $g_i$ , is not active, while  $\lambda_i > 0$  when  $g_i$  is active. In poor words, it is unnecessary to consider Lagrange multipliers for inactive constraints. From a coding point of view, it is not so easy to turn off/on different constraints as the optimization proceeds, and thus all constraints and Lagrange multipliers are retained. When a constraint is inactive, the corresponding Lagrange multiplier will converge to its lower bound, which reduces its effect on the optimality criteria.

### 3.2.3 The Method of Moving Asymptotes (MMA)

The method of moving asymptotes, MMA, was developed by Krister Svamberg ([10]-[34]), and it is an extremely efficient algorithm for constrained non linear optimization. The MMA is based on a special type of convex approximation. Here, we do not show all the computations needed to completely define the method. The code used in this thesis has been adapted from the MATLAB version[35].

Let us consider an optimization problem (in a standard form) :

P: minimize

$$f_0(\mathbf{x}) \quad (\mathbf{x} \in \mathbf{R}^N),$$

subject to

$$\begin{aligned} f_i(\mathbf{x}) &\leq \hat{f}_i, \quad \text{for } i = 1, \dots, M \\ \underline{x}_j &\leq x_j \leq \bar{x}_j, \quad j = 1, \dots, N, \end{aligned}$$

where  $\mathbf{x} = (x_1, \dots, x_N)^T$  is the vector of design variable ( $x_j = \gamma_j$ ),  $f_0(\mathbf{x})$  is the objective function ( $J$  in our case), and  $f_i(\mathbf{x}) \leq \hat{f}_i$  are the constraints, with the additional lower and upper bounds for each variables  $\underline{x}_j$  and  $\bar{x}_j$ , respectively. In short, the MMA scheme follow this iterative scheme:

1. Choose a starting point  $\mathbf{x}^0$ , and let the iteration index  $k = 0$ .
2. Given an iteration point  $\mathbf{x}^{(k)}$ , compute  $f_i^{(x^{(k)})}$  and the gradients  $\nabla f_i(\mathbf{x}^{(k)})$  for  $i = 1, \dots, M$ .
3. Generate a subproblem  $P^{(k)}$  by replacing in  $P$  the (usually implicit)

functions  $f_i$  by approximating explicit functions  $f_i^{(k)}$ , based on the calculations from Step 2.

4. Solve  $P^{(k)}$  and let the optimal solution of this subproblem be the next iteration point  $x^{(k+1)}$ . Let  $k = k + 1$  and go to Step 2.

The algorithm stops when the chosen convergence criterion is fulfilled. Substantially, each  $f_i^{(k)}$  is obtained by a linearization of  $f_i$  in variables of the type  $1/(x_j - L_j)$  or  $(U_j - x_j)$  dependent on the signs of the derivatives of  $f_i$  at  $\mathbf{x}^{(k)}$ .  $L_j$  and  $U_j$  are usually called "moving asymptotes" and are free to change between iterations.

It is clear that to define the MMA we need

- to define the functions  $f_i^{(k)}$
- to indicate the solving method of each subproblem  $p^{(k)}$

Here, we face only the former point, leaving only a brief suggestion to the solving method.

On assuming to have chosen  $L_j^{(k)}$  and  $U_j^{(k)}$  for the current iteration  $k$ , such that  $L_j^{(k)} < x_j^{(k)} < U_j^{(k)}$ , for each  $i = 0, \dots, M$ , we can define  $f_i^{(k)}$  as

$$f_i^{(k)}(\mathbf{x}) = r_i^{(k)} + \sum_{j=1}^N \left( \frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right), \quad (3.2.11)$$

where

$$p_{ij}^{(k)} = \begin{cases} (U_j^{(k)} - x_j^{(k)})^2 \partial f_i / \partial x_j, & \text{if } \partial f_i / \partial x_j > 0 \\ 0, & \text{if } \partial f_i / \partial x_j \leq 0 \end{cases} \quad (3.2.12)$$

$$q_{ij}^{(k)} = \begin{cases} 0, & \text{if } \partial f_i / \partial x_j \geq 0 \\ -(x_j^{(k)} - L_j^{(k)})^2 \partial f_i / \partial x_j, & \text{if } \partial f_i / \partial x_j < 0 \end{cases} \quad (3.2.13)$$

$$r_i^{(k)} = f_i(\mathbf{x}^{(k)}) - \sum_{j=1}^N \left( \frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j^{(k)}} + \frac{q_{ij}^{(k)}}{x_j^{(k)} - L_j^{(k)}} \right) \quad (3.2.14)$$

with the derivatives  $\partial f_i / \partial x_j$  evaluated at  $\mathbf{x} = \mathbf{x}^{(k)}$ . According to this set of equations,  $f_i^{(k)}$  is a first order approximation of  $f_i$  at  $\mathbf{x}^{(k)}$ , i.e.,

$$f_i^{(k)}(\mathbf{x}^{(k)}) = f_i(\mathbf{x}^{(k)})$$

and

$$\partial f_i^{(k)} / \partial x_j = \partial f_i / \partial x_j \text{ at } \mathbf{x} = \mathbf{x}^{(k)}, \quad \text{for } i = 0, \dots, M, \quad j = 1, \dots, N.$$

It can be easily checked that since  $p_{ij}^{(k)}, q_{ij}^{(k)} \geq 0$ ,  $f_i^{(k)}$  is a convex function, with the following second derivatives at  $\mathbf{x} = \mathbf{x}^{(k)}$

$$\frac{\partial^2 f_i^{(k)}}{\partial x_j^2} = \begin{cases} \frac{2\partial f_i / \partial x_j}{U_j^{(k)} - x_j^{(k)}}, & \text{if } \partial f_i / \partial x_j > 0 \\ -\frac{2\partial f_i / \partial x_j}{x_j^{(k)} - L_j^{(k)}}, & \text{if } \partial f_i / \partial x_j < 0. \end{cases} \quad (3.2.15)$$

According to Equation (3.2.15), the closer  $L_j^{(k)}$  and  $U_j^{(k)}$  are chosen to  $x_j^{(k)}$ , the larger are the second derivatives. This means that  $f_i$  turns out to be more "curved", and, since the approximation is valid only close to the points  $\mathbf{x}^{(k)}$ , the overall MMA result is strictly conservative. On the opposite, when  $L_j^{(k)}$  and  $U_j^{(k)}$  are chosen far from  $\mathbf{x}^{(k)}$ , the approximating functions become more

linear,  $f_i^{(k)}(\mathbf{x}) = f_i(\mathbf{x}^{(k)}) + \sum_j (\partial f_i / \partial x_j)(x_j - x_j^{(k)})$ , in the limit case of  $L_j^{(k)} \rightarrow -\infty$  and  $U_j^{(k)} \rightarrow +\infty$ , which are the approximating function used by the well-known "sequence of linear programs" method.

On assuming  $\underline{x}_j$  and  $\bar{x}_j$  physically reasonable, as simple choice for  $L_j^{(k)}$  and  $U_j^{(k)}$  is

$$L_j^{(k)} = x_j - s_0(\bar{x}_j - \underline{x}_j), \quad \text{and} \quad U_j^{(k)} = \bar{x}_j + s_0(\bar{x}_j - \underline{x}_j), \quad (3.2.16)$$

with  $s_0 \in [0, 1]$ . To enhance the model efficiency and prevent setting  $L_j$  and  $U_j$  at each iteration, the adoptions of some rules to lend the asymptotes is preferred by far.

A potential solution is based on the heuristic approach and states,

- to move the asymptotes closer to the current iteration point when the process tends to oscillate and needs to be stabilized.
- to move the asymptotes away from the current iteration point if the process is monotone and slow, and needs to be relaxed.

According to these rules, the  $P^{(k)}$  subproblem results in,

$P^{(k)}$ : minimize

$$\sum_{j=1}^N \left( \frac{p_{0j}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{0j}^{(k)}}{x_j - L_j^{(k)}} \right) + r_0^{(k)}$$

subject to

$$\sum_{j=1}^N \left( \frac{p_{ij}^{(k)}}{U_j^{(k)} - x_j} + \frac{q_{ij}^{(k)}}{x_j - L_j^{(k)}} \right) + r_i^{(k)} \leq \hat{f}_i, \quad \text{for } i = 1, \dots, M$$

$$\underline{x}_j < x_j < \bar{x}_j, \quad \text{for } j = 1, \dots, N.$$

From this formulation the solution is usually obtained by a primal-dual algo-

rithm, by solving the Karush Kuhn Tacker conditions by a Newton or Fletcher  
Reeves gradient method [10].

### 3.3 SIMPLE preconditioning and shared-memory multiprocessing

The SIMPLE (Semi-Implicit Pressure Linked Equation) has been introduced by Patankar as an iterative method to solve the finite volume discretized incompressible Navier-Stokes equations, using a staggered grid arrangement of the unknowns [36]. The convergence of the method depends on the relaxation parameter between velocity and pressure, but is usually quite slow. Nevertheless, the SIMPLE is widely used for CFD simulation and in many commercial packages like FLUENT.

The SIMPLE algorithm has been adopted just as preconditioner for the chosen GMRES algorithm, according to the procedure shown by Vuik in 2010 [37].

#### 3.3.1 The SIMPLE scheme

Let first analyze the SIMPLE approximation scheme for the solution of the system:

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} r_u \\ r_p \end{bmatrix}.$$

The first step is to rewrite the system with the Schur decomposition, i.e.,

$$\begin{bmatrix} A & 0 \\ B & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^T \\ 0 & I \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} r_u \\ r_p \end{bmatrix}.$$

By changing  $u$  and  $p$  with  $u^* = u + A^{-1}B^T p$  and  $p^* = p$ , respectively, the system reduces to

$$\begin{bmatrix} A & 0 \\ B & S \end{bmatrix} \begin{bmatrix} u^* \\ p^* \end{bmatrix} = \begin{bmatrix} r_u \\ r_p \end{bmatrix}.$$

It is possible to separate the new variables, solve the system for  $u^*$  and  $p^*$  separately, and finally express the solution in terms of  $u$  and  $p$ .

In short, the basic SIMPLE algorithm is

- Solve  $Au^* = r_u$
- Solve  $Sp^* = r_p - Bu^*$
- Update  $u = u^* - A^{-1}B^T p$
- Update  $p = p^*$

Of course this computations would lead to the exact solution only if the matrix inverses were assembled completely, which is not a possibility (see section 2.5). This formulation is usually developed in an iterative scheme by the use of approximate inverse of  $A^{-1}$  [36].

We then take advantage from this easy formulation and apply it as preconditioner for each step of the GMRES algorithm, applying the Jacobi preconditioners to the small systems  $Au^*$  and  $Sp^*$ .

### 3.3.2 The flexible inner preconditioned GMRES

A standard GMRES iteration requires the computation of the product  $w = Kv$ , with  $K$  matrix of the system. When a right preconditioner is available this

becomes  $KP_R^{-1}v^*$ , with  $v^* = P_R v$ , and the scheme works by the construction of a basis of the Krylov space of  $KP_R^{-1}$ .

If the exact  $K^{-1}$  were available as preconditioner, it would lead to

$$w = KK^{-1}v^*, \quad K^{-1}v^* = v.$$

Of course the system  $K^{-1}v^* = v$  has the same complexity of the original problem, and despite the use of  $K^{-1}$  allows us to solve the iterative cycle in just one step, we do not expect any gain in the model performance. From what stated in Section (2.5.1), using the exact  $K^{-1}$  as preconditioner would transform the system matrix into the identity, which has only one eigenvalue, so the system is solved immediately by the scheme. Identify now, for simplicity of notations,  $v$ , with  $v^*$ , and,  $z$ , with  $v$ , so that

$$w = Kz, \quad K^{-1}v = z.$$

The real solution would solve immediately the problem, but we may imagine an approximate solution by solving the system with another preconditioner  $P_R$  instead of  $K$ , i.e.,

$$P_R \tilde{z} = v,$$

where  $\tilde{z}$  is the approximate solution of  $z$ . In this sense approximating the solution of the system  $Kz = v$  is comparable to the use of a different preconditioner at each iteration of the GMRES scheme. For this reason, the reference scheme is called flexible GMRES (FGMRES) algorithm [25].

Defining with  $M_j$  the preconditioner used at the  $j_{th}$  iteration, the scheme for GMRES with variable preconditioning is the following,

**GMRES with variable preconditioning**

1. Choose  $x_0$  and a restart parameter  $m$
2. **Arnoldi process:**
  - For**  $j = 1, \dots, m$ 
    - (a) Compute  $z_j$  from  $M_j z_j = v_j$
    - (b) Compute  $w = K z_j$
    - (c) For  $i = 1, \dots, j$   $\begin{cases} h_{i,j} = \langle w, v_i \rangle \\ w = w - h_{i,j} v_i \end{cases}$
    - (d) Compute  $h_{j+1,j} = \|w\|_2$  and  $v_{j+1} = w/h_{j+1,j}$ .
3. Define  $\mathcal{Z}_m = [z_1, \dots, z_m]$
4. Form the approximate solution:
  - (a) Compute  $y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$ ,  $e_1 = [1, 0, \dots, 0]^T$
  - (b) Compute  $x_m = x_0 + \mathcal{Z}_m y_m$
5. If residual less than tolerance STOP, else restart and go to 2.

### The SIMPLE for variable preconditioning

The SIMPLE preconditioning scheme does not work with different preconditioners  $M_j$  as in the FGMRES scheme, but it has anyway to save all the the partial results  $z_j$ , since systems  $K z_j = v_j$  are only solved approximately. Approximating such solutions with one iteration of the SIMPLE method, the implemented scheme finally results in

1. Choose  $x_0$  and a restart parameter  $m$
2. **Arnoldi process:**
  - For**  $j = 1, \dots, m$

- (a) SIMPLE: solve approximately system  $Kz_j = v_j$  for  $z_j$
- i. solve  $Au^* = r_u$
  - ii. Solve  $Sp^* = r_p - Bu^*$
  - iii. Update  $u = u^* - A^{-1}B^T p$
  - iv. Update  $p = p^*$
  - v.  $z_j = \begin{bmatrix} u \\ p \end{bmatrix}$
- (b) Compute  $w = Az_j$
- (c) For  $i = 1, \dots, j$   $\begin{cases} h_{i,j} = \langle w, v_i \rangle \\ w^- = h_{i,j}v_i \end{cases}$
- (d) Compute  $h_{j+1,j} = \|w\|_2$  and  $v_{j+1} = w/h_{j+1,j}$ .

3. Define  $\mathcal{Z}_m = [z_1, \dots, z_m]$

4. Form the approximate solution:

- (a) Compute  $y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_m y\|_2$ ,  $e_1 = [1, 0, \dots, 0]^T$
- (b) Compute  $x_m = x_0 + \mathcal{Z}_m y_m$

5. If residual less than tolerance STOP, else restart and go to 2.

The scheme comparison against the other proposed preconditioners and PAR-DISO commercial software [38] are shown in (6.3).

### Some theoretical properties

Let now prove that  $y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_{m+1,m} y\|_2$ , is the best possible set of coefficient for the linear combination of the  $\mathcal{Z}_m$  vectors to minimize the norm of the residual

$$r_m = b - K(x_0 + \mathcal{Z}_m y).$$

The main difference with the basic GMRES algorithm is that the Krylov subspaces enlarged with a vector obtained as

$$w_{k+1} = K z_k,$$

with  $z_k$  owing to a possible variable preconditioning as in (3.3.2).

The new basis vector thus reads

$$\hat{v}_{k+1} = K z_k - \sum_{j=1}^k h_{j,k} v_j, \quad v_{k+1} = \frac{\hat{v}_{k+1}}{\|\hat{v}_{k+1}\|_2},$$

with

$$h_{j,k} = \langle w_{k+1}, v_j \rangle = \langle K z_k, v_j \rangle = z_k^T K^T v_j = v_j^T K z_k.$$

Multiplying both sides by  $v_{k+1}^T$ , using  $\hat{v}_{k+1} = \|\hat{v}_{k+1}\|_2 v_{k+1}$ , and the orthonormality of all the vectors  $v_j$  by construction, we can prove that

$$\|\hat{v}_{k+1}\|_2 = v_{k+1}^T K z_k = h_{k+1,k},$$

that easily leads to

$$h_{k+1,k} v_{k+1} = K z_k - \sum_{j=1}^k h_{j,k} v_j \implies K z_k = \sum_{j=1}^{k+1} h_{j,k} v_j.$$

In matrix form the last result reads

$$K Z_m = V_{m+1} \bar{H}_{m+1,m}, \quad (3.3.1)$$

with  $\bar{H}_m (m+1) \times m$  the upper Hessenberg matrix collecting all the scalars  $h_{j,k}$ .

The residual of the linear system at the step  $m$  is equal to

$$r_m = b - K(x_0 + \mathcal{Z}_m y) = r_0 - A\mathcal{Z}_m y$$

and can thus be rewritten as

$$r_m = r_0 - V_{m+1} \bar{H}_{m+1,m} y,$$

which is exactly the formulation of the standard GMRES, and leads directly to

$$y_m = \operatorname{argmin}_y \|\beta e_1 - \bar{H}_{m+1,m} y\|_2$$

as the best set of coefficient to minimize the residual at the current iteration also for the GMRES scheme with variable preconditioning.

### Lucky and hard breakdown

It has not yet been discussed the role of the new vector  $z_k$ , when the newly created vector  $w_{k+1} = Kz_k$  is linearly dependent from other vectors of the base and thus the Krylov subspace dimension cannot increase. Does the "lucky breakdown" property still hold for the flexible GMRES? Is the condition  $h_{k+1,k} = 0$  still sufficient for an exact convergence?

Unfortunately the answer is no, in general.

In the standard GMRES if the new  $w_{k+1}$  is linearly dependent from the vector in the base, the orthogonalization scheme leads to  $h_{k+1,k} = 0$ , which means that the last row of the Hessemberg matrix  $H_{k+1,k}$  is all made by zero entries. Now, noticing that in such situation [39]

$$V_{k+1} H_{k+1,k} = V_k H_{k,k},$$

the  $k_{th}$  residual gets the form

$$r_m = r_0 - V_k H_{k,k} y, \quad (3.3.2)$$

which can be imposed equal to zero, leading to a  $m \times m$  linear system that can be solved exactly ("lucky" termination).

The problem with variable preconditioning is precisely in this final step. While GMRES guarantees that the final  $H_{k,k}$  is full rank, with the new approach, the condition  $h_{k+1,k} = 0$ , may lead to either a "lucky breakdown" case with a full rank  $H$ , or to a "hard breakdown" case, with  $H$  singular.

$H$  is singular when two or more columns are linearly dependent, which roughly means that the last vector  $w_{k+1}$  is linearly dependent from the previous ones  $w_j$ , with  $j \leq k$ , used to build the space. Since  $w_{k+1} = Kz_k$ , this can only happen if  $z_k$  is linearly dependent from the previously used  $z_j$ [40]. This is impossible in standard GMRES with  $z_k = v_k$ , because this would mean that a vector of the basis is linearly dependent from the others.

Summarizing, when FGMRES incurs in  $h_{k+1,k} = 0$ , the algorithm stops for "lucky breakdown" if  $H$  is of full rank, and fails for "hard breakdown", if  $z_k$  is linearly dependent from the previous  $z_j$ .

The SIMPLE preconditioning would thus present the same problematic, but, with  $z_k$  coming from the approximate solution of  $Kz_k = v_k$ , it is extremely unlikely for the  $z_j$  to be linearly dependent, especially for high degrees of freedom.

For the implementation, when the current  $z_k$  yields  $h_{k+1,k} = 0$  with non zero residual, the GMRES scheme is restarted with the last approximate solution obtained with the previous  $z_j$ .

### 3.3.3 The Shared memory multiprocessing

In order to decrease the required computational time for the solving process, the GMRES-SIMPLE scheme has been adapted to a SMP (Shared memory multiprocessing) routine. The implementation just aims at accelerating the time-consuming products between matrix and vector, and matrix by matrix, as well as each simple operation of easy parallelization. Used functions follows the openMP directives [41]. Some results on efficiency and speed up are shown in appendix B (6.3).

## 3.4 The Topology Optimization: the code structure

In this section we present a series of examples that are analyzed according to the TO approach. The procedure to run the simulations from the original domain of analysis is reported in the following subsection.

### 3.4.1 The Topology Optimization algorithm for fluid flow

First, the mesh discretizations have been obtained by COMSOL Multiphysics [42]. COMSOL output mesh file (.mphant or .mphbin formats) contains all the relevant information about domain and boundaries, which are read and re-elaborated to create and add an inner finer grid, which is required to build the chosen FEM stable elements (see section 2.4.1).

Once the geometry has been imported and elaborated by the preprocessing executable, each problem requires the compilation of three input files, namely the geometry, optimization and print file.

The geometry file specifies the problem name, according to the file in which the geometry has been saved, the solver for the NS problem, and all the physical properties of the fluid. The fluid density, viscosity, external forcing fields, and boundary conditions are all prescribed in this file.

The optimization file, instead, specifies the value for all the optimization parameters,  $\alpha_{max}$ ,  $q$ , the maximum number of optimization iterations, the admissible volume fraction,  $V_r$ . If the optimization domain is smaller than the total domain, this file includes also the coordinates of the bounds of the subdomain to optimize.

Finally, the print file contains the printing parameters, like the maximum number of the output files. A user may also choose to print just the material distribution rather than the fluid and pressure solution at each printing iteration.

Figure (3.1) shows the flowchart of the complete TO implemented scheme, while Figure (3.2) reports the NS solving routine.

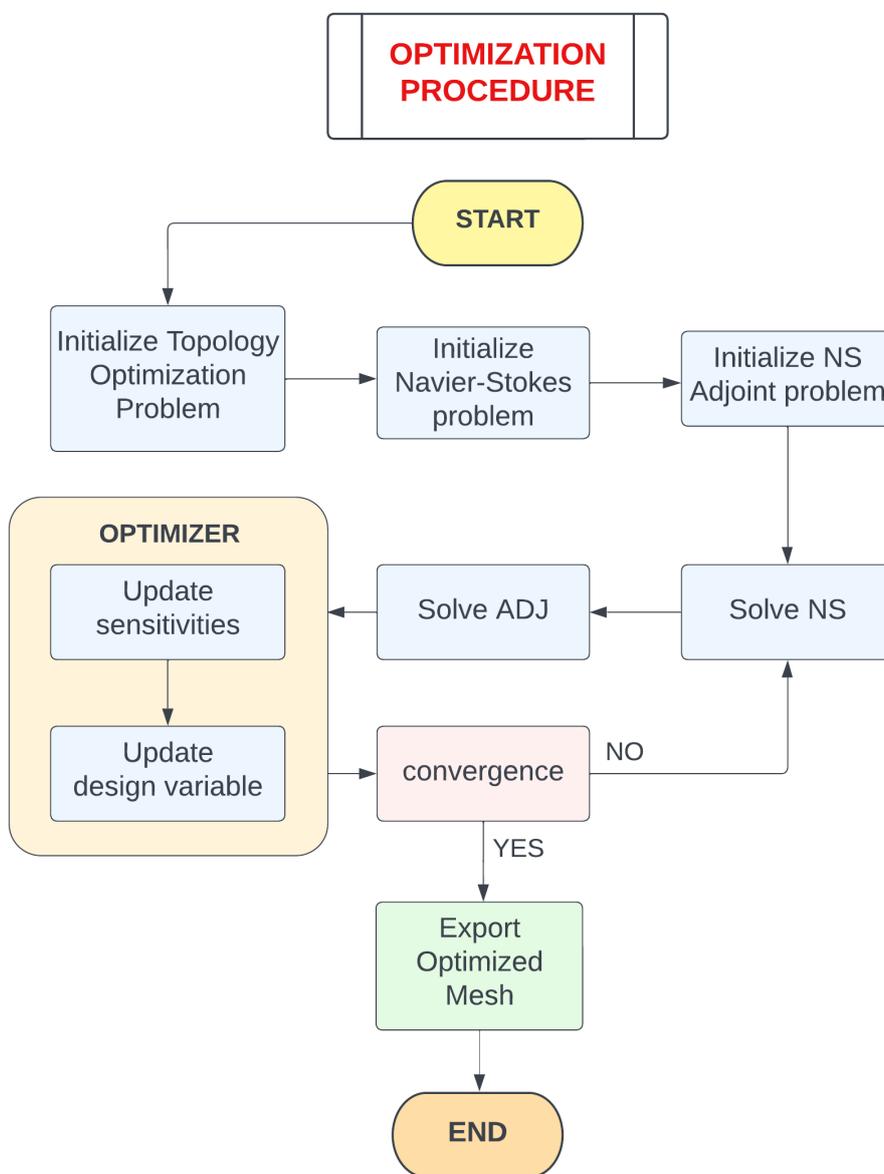


Figure 3.1: TO algorithm flowchart

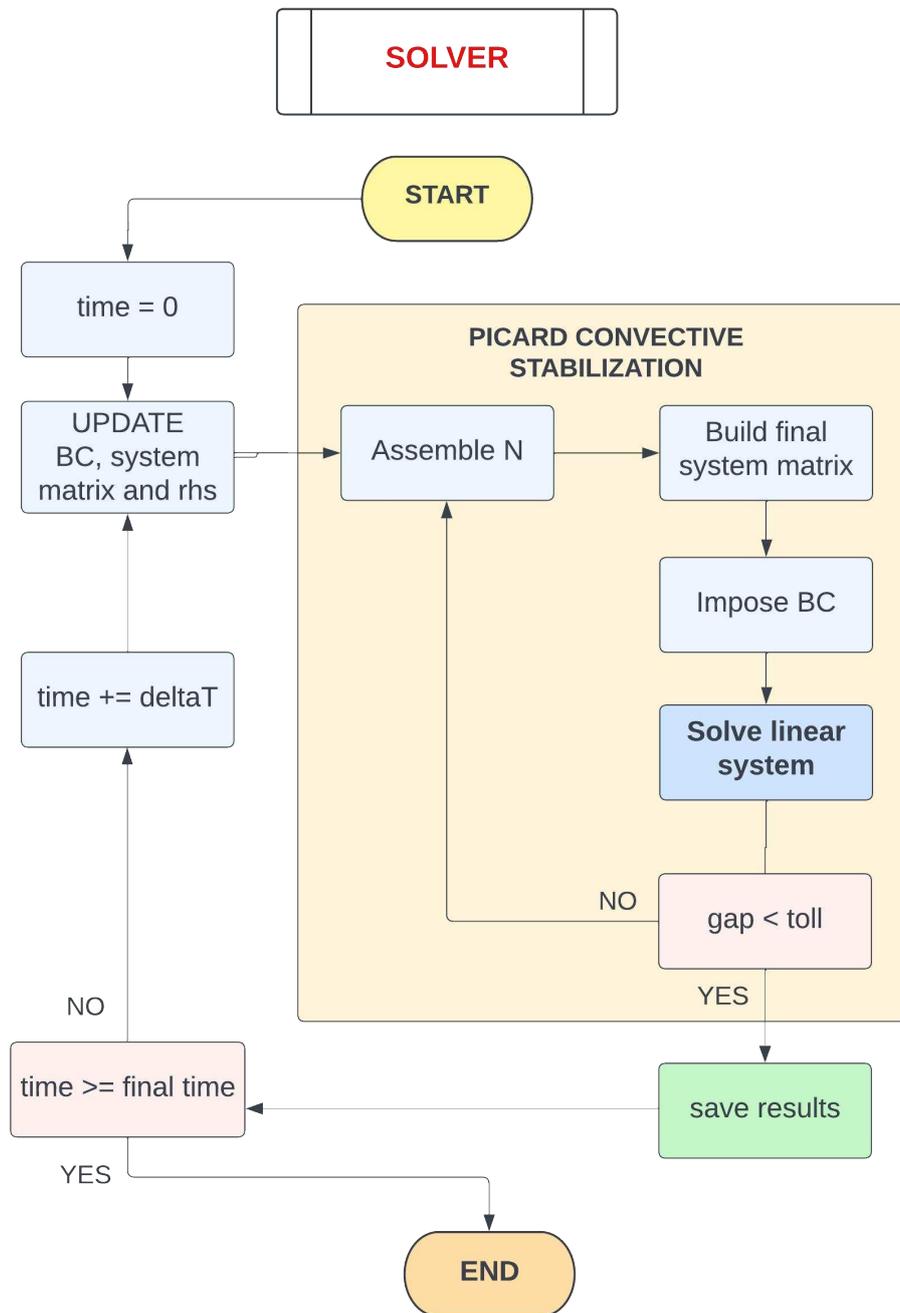


Figure 3.2: Navier-Stokes solver scheme

### 3.4.2 The TO for species transport

The scheme for the TO algorithm for species transport just requires the solving of two additional systems in the formulation, the ADR problem and its adjoint.

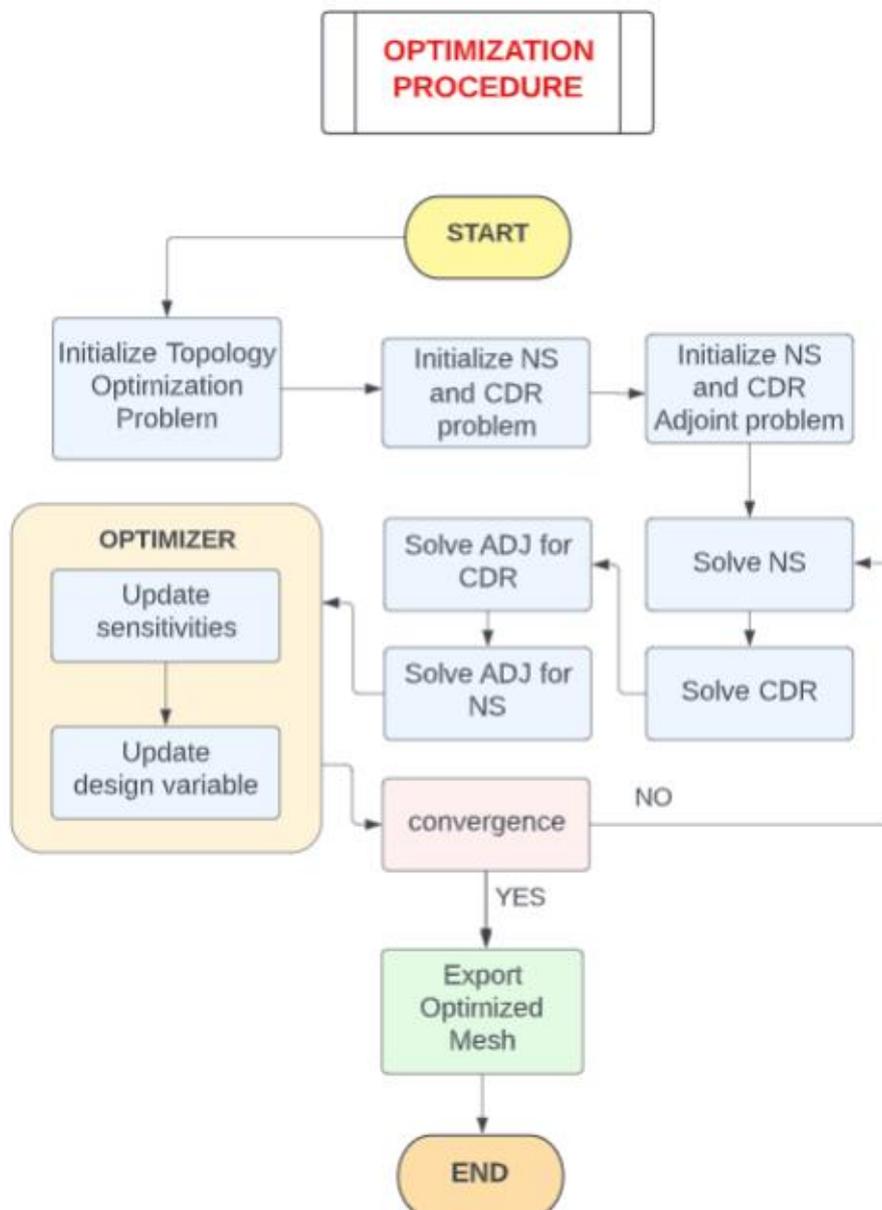


Figure 3.3: TO algorithm flowchart

## 3.5 Validation procedure and testing for the TO algorithm

### 3.5.1 TO for fluid flow

All the consider analysis for fluid flow problems have been developed over the TO formulation with minimization of the dissipation power, over the additional volume constraint, in absence of external force fields. Namely,

$$\left\{ \begin{array}{ll} \min_{\gamma} J(\gamma) = \int_{\Omega} \left[ \frac{1}{2} \mu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^T\|^2 + \alpha \|\mathbf{u}\|^2 \right] d\mathbf{x} \\ \text{subject to} & : \int_{\Omega} \gamma(\mathbf{x}) d\mathbf{x} - \beta |\Omega| \leq 0, \quad \text{Volume constraint,} \\ & : 0 \leq \gamma(\mathbf{x}) \leq 1, \quad \text{Design variable bounds,} \\ & : \text{NS equations (2.4.1),} \quad \text{Governing equations.} \end{array} \right. \quad (3.5.1)$$

The following examples are only validated against numerical results obtained in literature, since for most of them still suffer the lack of an experimental validation.

### 2D validation

#### Double Pipe problem

To validate the developed Topology Optimization algorithm for steady fluid flows problems, a first study has been conducted for the standard "double pipe" problem [32]. The double pipe geometry is showed in Figure (3.4). The computational domain consists of two straight channels, of dimension  $1/6 \times 0.5\text{m}$ , entering into a rectangular box, of height 1m and varying length,

$L$ . Finally two identical channels,  $1/6\text{m} \times 0.5\text{m}$ , exiting from the box. The optimization domain is the central box, and the parameter  $L$  is used to investigate the differences between optimal layouts at the varying of the box length.

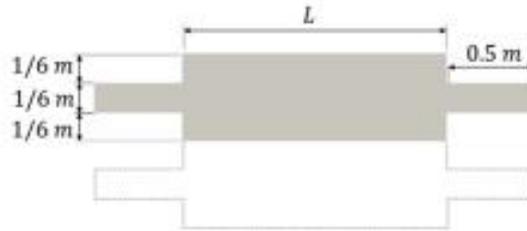


Figure 3.4: The "Double pipe" geometry, by [32]. Grey area represents the analyzed domain due to symmetry.

At the two inlet channels (left) and outlet (right) channels, we prescribe parabolic velocity, two outlet channels (at the right), and zero-pressure condition, respectively, whereas, wall bounds (no-slip boundaries) on the remaining contours. By symmetry, the problem can be cut at  $y = 0.5\text{m}$  (grey area), thereby the hydrodynamics is solved only on the upper half of the domain.

The fluid properties and topology parameters are listed in Table (3.1),

| $\mathbf{u}_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|--------------------------|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 1                        | 1                           | 1                         | 1e4                  | 1   | 0.34  | 50    |

Table 3.1: Parameter settings in the topology optimization of the double pipe.

The numerical experiment is studied for two scenarios,  $L = 1.0\text{m}$ , and  $L = 1.5\text{m}$ , with a triangular discretization of more than  $10^5$  elements.

### Unsteady double pipe problem

We use the "double pipe" also to validate the developed Topology Optimiza-

tion algorithm for unsteady problems. In this case, we simplify the domain as shown in Figure (3.5). The computational domain consists only in a box of dimensions  $1 \times 1\text{m}$ , which coincides with the optimization domain.

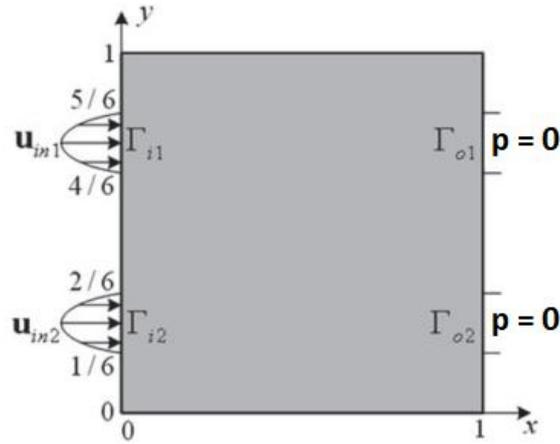


Figure 3.5: Unstable double pipe geometry, by [16].

In this unsteady problem we prescribe at the two inlet boundaries (at the left), the following oscillating parabolic velocity,

$$\mathbf{u}_{in,1} = -144(y - 4/6)(5/6 - y) \cos(t)\mathbf{n}, t \in [0, 2\pi]$$

$$\mathbf{u}_{in,2} = -144(y - 1/6)(2/6 - y) \sin(t)\mathbf{n}, t \in [0, 2\pi],$$

where  $\mathbf{u}_{in,1}$  and  $\mathbf{u}_{in,2}$  are the velocity at the upper and lower inlets, respectively. We also prescribe zero-pressure outlet conditions (at the right), and wall bounds, i.e. no-slip conditions, on the rest of the external geometry. A zero velocity field is assumed as initial condition.

The fluid properties and topology parameters adopted in the analysis are summarized in Table (3.2).

| $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 1                           | 1                         | 1e4                  | 1   | 0.34  | 50    |

Table 3.2: Parameter settings in the topology optimization of the unsteady double pipe.

The computational mesh consists in more than  $2 \cdot 10^5$  triangular elements, and the simulation analyze a period of 6.28 s with a time-step of 0.3925 s (equal to 1/16 of the period).

#### The 4 terminal device problem

One of the main properties of a Topology Optimization algorithm should be the capability to change the domain topology, for example generating holes inside the optimization domain. This effect is tested with the steady "4 terminal device problem" [16].

The computational domain consists of a  $1\text{m} \times 1\text{m}$  box, with two inlet boundaries at the lateral edges and two outlets on the upper/lower side. Each inlet/outlet is positioned at the center of each edge, with size of  $1/3$  m. The optimization domain coincides with the total domain (see Figure 3.6).

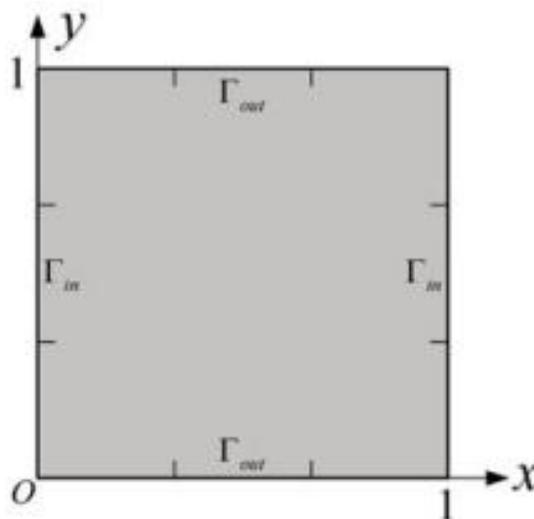


Figure 3.6: "4 terminal device" geometry, by [16].

The 4 terminal device presents two inlet boundaries  $\Gamma_{in}$  (left and right walls), where we prescribe parabolic velocity profile, two zero-pressure outlets  $\Gamma_{out}$  (top and bottom walls), and wall bounds on the rest of the external geometry.

The fluid properties and topology parameters are showed in Table (3.3)

| $\mathbf{u}_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|--------------------------|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 1                        | 1                           | 1                         | 1e4                  | 1   | 0.4   | 50    |

Table 3.3: Parameter settings in the topology optimization of the 4 terminal device.

The problem is studied by a triangular with of more than  $2.0 \cdot 10^5$  elements.

### The unsteady Bend channel problem

To observe the effect of the Reynolds number,  $Re = UL/\nu$ , with  $U$  the characteristic velocity and  $L$  the characteristic length of the problem, we analyze the unsteady "Bend channel" problem [16].

The computational domain consists of a  $5 \times 5$ m box, coinciding with the optimization domain, with an inlet (left) and an outlet channel (bottom), having size of  $1.0 \times 1.5$ m (see figure 3.7).

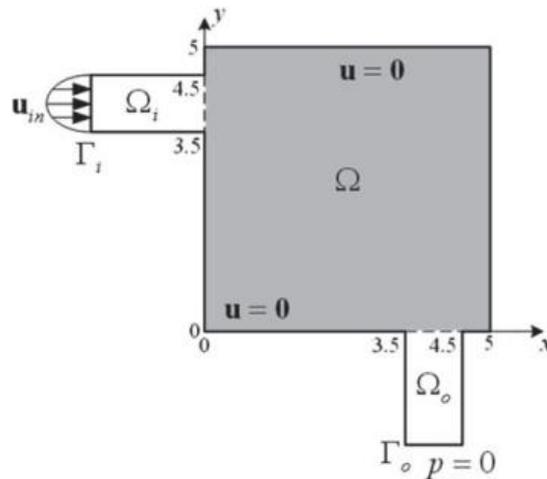


Figure 3.7: "Bend channel" geometry, by [16].

The bend channel problem is implemented with one inlet boundary condition, prescribing a time-dependent parabolic velocity profile, and one zero-pressure condition at the outlet. We finally prescribe wall bounds on the rest of the external geometry. A zero velocity field is assumed for the initial condition. The velocity profile is assumed to grow linearly in time as

$$\mathbf{u}_{\Gamma_i} = \mathbf{u}_{max}(y - 3.5)(4.5 - y)t, \quad (3.5.2)$$

with  $t$  varying from 0 to 1 second, which is the final computational time.

To test the effects of a different  $Re$ , two simulations have been developed changing the maximum inflow velocity  $\mathbf{u}_{max}$  from 1 to 100 m/s, assuming a unitary density and kinematic viscosity.

Fluid properties for the two simulations and topology parameters are summarized in Table (3.4),

| $\mathbf{u}_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|--------------------------|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 1                        | 1                           | 1                         | 1e4                  | 1   | 0.25  | 50    |
| 300                      | 1                           | 1                         | 1e4                  | 1   | 0.25  | 50    |

Table 3.4: Parameter settings in the topology optimization of the bend channel.

The problem is studied with a triangular mesh of more than  $2.0 \cdot 10^4$  elements, and a time step of 1/16 seconds.

### The 3D validation

#### The Bend channel

To validate the implemented Topology Optimization algorithm for three-dimensional

steady flow problems, at different Reynolds numbers, we consider a 3D adaptation of the "bend channel" example [43]. The computational domain consists of a box of  $1 \times 0.4 \times 1$  m, with a circular inlet at the left face and a circular outlet at the bottom face (see Figure 3.8). The inlet and outlet diameter is constant and equal to 0.2m.

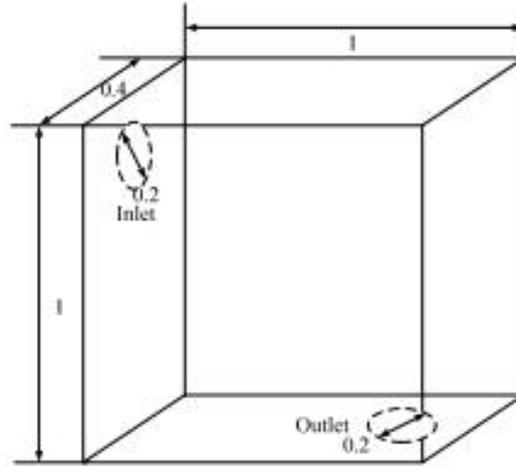


Figure 3.8: Three-dimensional "bend channel" geometry, by [43].

The 3D bend channel problem is implemented prescribing a parabolic velocity profile at the inlet, one zero-pressure at the outlet, and wall bounds on the rest of the external geometry.

To test the effects of  $Re$ , two simulations have been developed, changing the maximum inflow velocity from  $\mathbf{u}_{max} = 1$  m/s, to  $\mathbf{u}_{max} = 50$  m/s. The density and kinematic viscosity are kept fixed to the unitary value. By symmetry, the problem can be cut by the sagittal. Thereby the hydrodynamics can be solved only on the one half of the domain.

Fluid properties for the two simulations and topology parameters are showed in Table (3.5),

The problem is studied with a tetrahedral mesh with more than  $3.0 \cdot 10^6$  ele-

| $u_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|-----------------|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 1               | 1                           | 1                         | 1e4                  | 1   | 0.1   | 30    |
| 50              | 1                           | 1                         | 1e4                  | 1   | 0.1   | 30    |

Table 3.5: Parameter settings in the topology optimization of the three-dimensional bend channel.

ments.

### The 6 terminal device

Lastly, we consider the generalization of the 4 terminal device in three-dimensions, namely the "6 terminal device" problem Figure(3.9) [43]. The computational domain consists of a  $5 \times 5 \times 5$  m box, with a small circular channel (radius 0.5m height 1m) connected to the middle of each face.

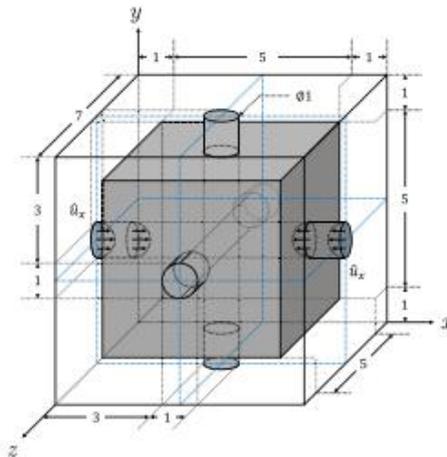


Figure 3.9: "6 terminal device" geometry, by [43].

The 6 terminal device presents two inlet boundaries (channel along the  $x$ -direction), where we prescribe parabolic velocity profile, four zero-pressure outlets (all the other channels), and wall bounds on the rest of the external geometry. By symmetry, the problem can be cut on the planes  $y = 3.5$ m and  $z = 3.5$ m, so that the hydrodynamics is solved only on 1/4 of the domain.

The fluid properties and topology parameters are showed in Table (3.6),

| $\mathbf{u}_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|--------------------------|-----------------------------|---------------------------|----------------------|-----|-------|-------|
| 200                      | 1                           | 1                         | 1e5                  | 1   | 0.1   | 30    |

Table 3.6: Parameter settings in the topology optimization of the 6 terminal device.

The problem is studied on a tetrahedral mesh of more than  $5 \times 10^5$  elements.

### The choice of the starting configuration

Each problem has been studied with the developed optimization schemes described in section 3.2(Optimality Criteria, Generalized Optimality Criteria and Method of Moving Asymptotes). Every optimization algorithm has its own dependence on the initial configuration, that may change the optimization process and the final result. A popular idea is to use initially a uniform material distribution exactly satisfying the volume constraint,  $\gamma = V_r$ . This initial configuration has proven to accelerate the scheme convergence velocity, but often leads to oscillations and checkboard effects on the solutions when a proper density filter is not applied [44]. The chosen interpolation rule for the damping parameter  $\alpha$ , Equation (2.2.2), makes easier for the optimization process to add material than to remove it. For this reason, a good initial configuration is simply a void optimization domain. In this way, the algorithms are able to build the final domain only by augmenting the material density in proper regions, and it is not necessary to remove material where the fluid is flowing.

This choice for the initial condition is not consistent with the imposed volume constraint, but all the implemented optimization schemes are able to evolve the domain even starting from unfeasible layouts. Accordingly, if not speci-

fied otherwise, all the results are obtained considering a completely fluid initial layout.

It is important to remark that none of the implemented optimization schemes is guaranteed to reach the true optimal configuration. The topology optimization is less constrained than the size and the shape optimization on the initial material distribution, but can still stop on local minimum. For this reason in practical applications the optimization scheme should be tested over several initial configurations and values of the optimization parameters,  $\alpha_{max}$ , and  $q$ , or different interpolation rules for  $\alpha(\gamma)$ . Moreover, when an intuitive solution is known, it may be important to use an initial material distribution which force the algorithm to evolve in that direction, and check whether the obtained result provides a better compliance than the ones with different initial configurations.

### 3.5.2 TO for species transport

Given the lack of well-known solutions for TO applications in flow-driven scalar transport problems, a single, but in depth analyzed scenario has been tested, to eventually create a well-documented benchmark for future applications. The validation will thus only be semi-qualitative, with a physical analysis on the common sense of the final configurations. The study is based on the weighted pollute-energy dissipation functional over the additional volume constraint. The optimization problem takes the final form,

$$\left\{ \begin{array}{ll} \min_{\gamma} J(\gamma) & = \beta_1 \int_{\Omega} \left[ \frac{1}{2} \mu \|\nabla \mathbf{u} + (\nabla \mathbf{u})^T\|^2 + \alpha \|\mathbf{u}\|^2 \right] d\mathbf{x} + \beta_2 \int_{\Gamma_{out}} c d\sigma \\ \text{subject to} & : \int_{\Omega} \gamma(\mathbf{x}) d\mathbf{x} - \beta |\Omega| \leq 0, \quad \text{Volume constraint,} \\ & : 0 \leq \gamma(\mathbf{x}) \leq 1, \quad \text{Design variable bounds,} \\ & : \text{NS equations (2.4.1),} \quad \text{Governing equations,} \\ & : \text{ADR equations (2.4.7),} \quad \text{Governing equations.} \end{array} \right. \quad (3.5.3)$$

The ratio  $\xi = \beta_2/\beta_1$  fix the schemes optimization priority. When  $\xi \rightarrow 0$ , the optimization should be focused on the minimization of the energy dissipation, while for  $\xi \rightarrow \infty$  the objective should only keep count of the pollute dissipation. Let us assume in the following zero specie generation inside the domain,  $f = 0$ , Equation (2.4.2).

#### The Bend channel problem

The case study derives from the bend channel adopted in the only CFD analysis. The computational domain consists of a rectangle of dimensions  $1 \times 4\text{m}$ , with an inlet bound of size  $0.25\text{m}$  on the left wall, and an outlet edge of the

same size on the right wall (see Figure 3.10).

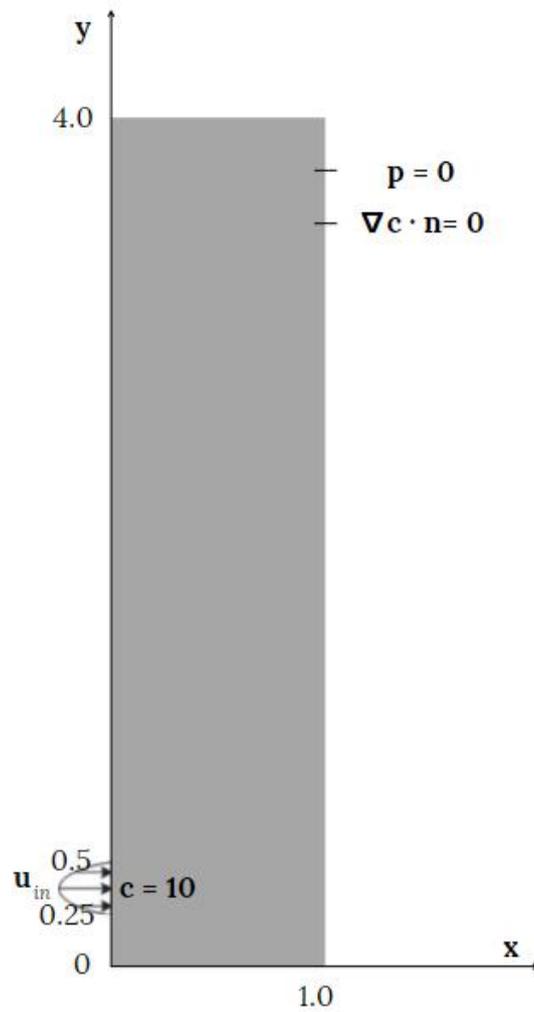


Figure 3.10: "bend channel" geometry.

The boundary conditions for the problem are prescribed by a parabolic velocity at the inlet, a zero-pressure condition at the outlet, and no-slip conditions on the remaining geometry.

The fluid properties and topology parameters are listed in Table (3.7),

| $\mathbf{u}_{max}$ [m/s] | $\rho$ [kg/m <sup>3</sup> ] | $\nu$ [m <sup>2</sup> /s] | $D$ [m <sup>2</sup> /s] | $k_s$ [1/s] | $\alpha_{max}$ [1/s] | $q$ | $V_r$ | maxIt |
|--------------------------|-----------------------------|---------------------------|-------------------------|-------------|----------------------|-----|-------|-------|
| 1                        | 1                           | 1                         | 10 <sup>-5</sup>        | 0.1         | 1e4                  | 1   | 0.4   | 50    |

Table 3.7: Parameter settings in the topology optimization of the bend channel.

The numerical experiment is studied with a triangular discretization of more than  $2 \times 10^4$  elements. Five different simulations, varying the weighting coefficient ratio  $\xi = \beta_2/\beta_1$ , are developed for this configuration, namely,  $\xi = 0, 1, 10, 100, \infty$ , with a value of  $\beta_1$  fixed to one for the firsts four scenarios, while zero in the last, and a value of  $\beta_2$  varying from one to zero.



# Chapter 4

## Results and Discussion

### 4.1 TO for fluid flows

#### 4.1.1 2D testing

##### The steady flow optimization

The solution of the example (3.4) has been already proposed by [32], and is showed in Figure (4.1).

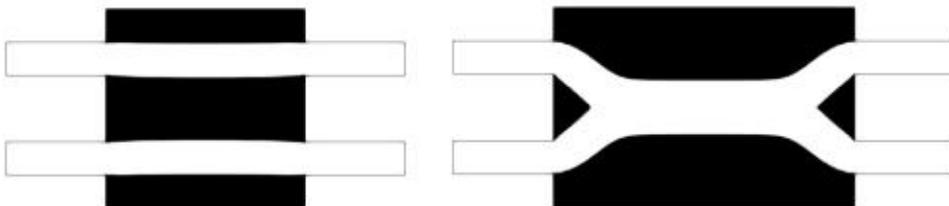


Figure 4.1: *Optimal layouts for  $L = 1.0m$  (left) and  $L = 1.5m$  (right). From Okubo Jr. [32].*

In the first simulation ( $L = 1m$ ) our models results are reported with varying the iterations in Figures (4.2) and (4.3). Each optimization scheme shows a different optimization process, both in the layouts and in convergence ve-

locity. The only algorithm that could reach the configuration proposed by the literature, with two straight channels connecting inlets and outlets, was the MMA, with a final compliance of  $J = 17.51$  and volume 33.99% of the starting optimization domain,  $V_0$ , achieved in 27 iterations.

The OC method provides an almost equivalent solution in terms of compliance ( $J = 18.83$ ), and a faster convergence (9 iterations), is not able to satisfy exactly the volume constraint, converging to a final volume of 34.67% $V_0$ ). Moreover, in the OC final configuration the two resulting channels shows a central bending towards the domain center, probably effect of the optimization process, that got stuck in a local minimum (see Figure(4.2)).

Finally, the GOC algorithm gives a completely different final layout, that merge the two entering channels into a bigger one. This is probably due to the optimization process of the algorithm, which operates only small changes per iteration, and converge more likely in local minima. The GOC final configuration present a compliance  $J = 34.47$ , with a final volume of 32.01% of  $V_0$ .

The optimization performances are showed in Figure (4.4). From the graphs in Figure (4.4), it is evident how the OC and MMA methods search immediately for feasible solution ( $Vol/V_0 < 0.34$ ), while the GOC algorithm just requires the solution feasibility at the end of the optimization process.

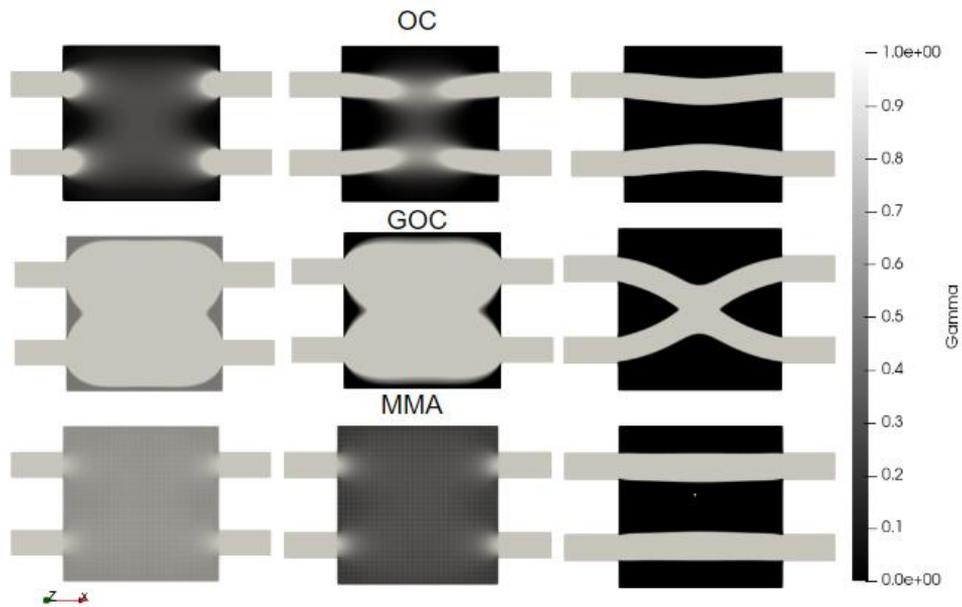


Figure 4.2: Snapshots of optimization procedure for the steady double pipe example in Figure (3.4) for  $L = 1\text{m}$ . From the left, configurations at iteration 2, 5, and final result

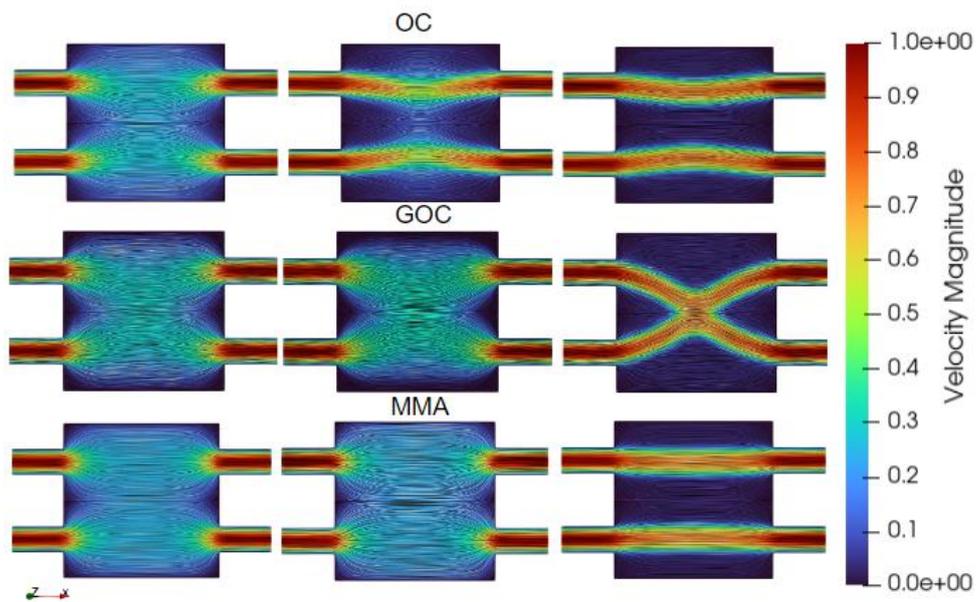


Figure 4.3: Snapshots of flow field at the optimization procedure for the steady double pipe example in Figure (3.4) for  $L = 1\text{m}$ . From the left, configurations at iteration 2, 5, and final result.

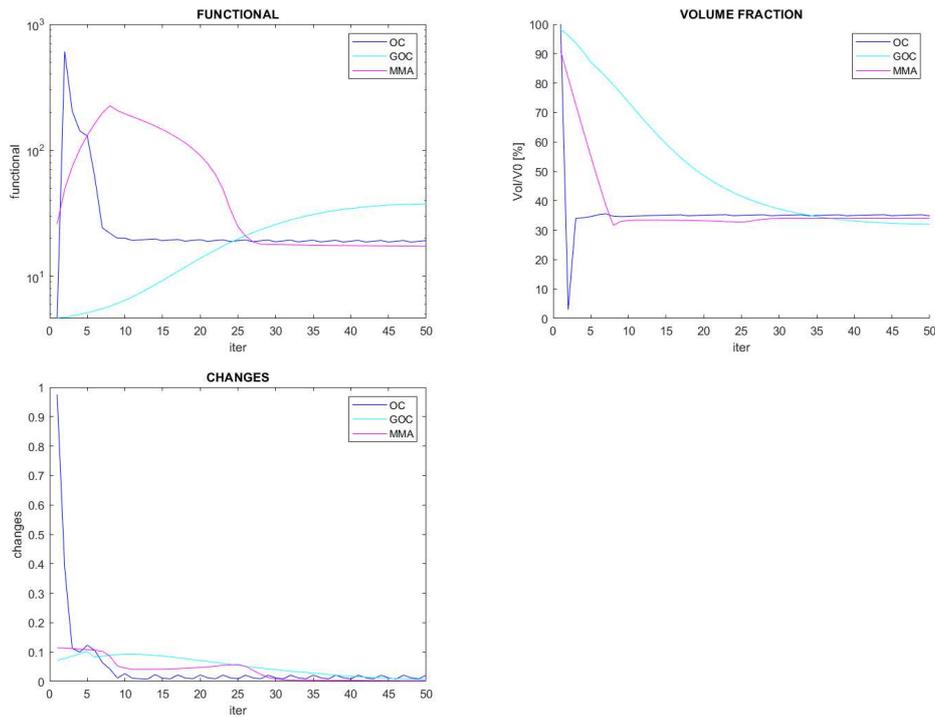
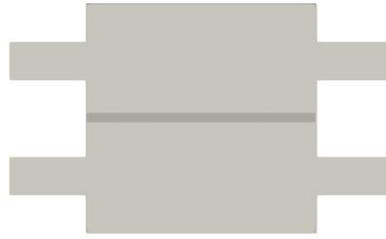
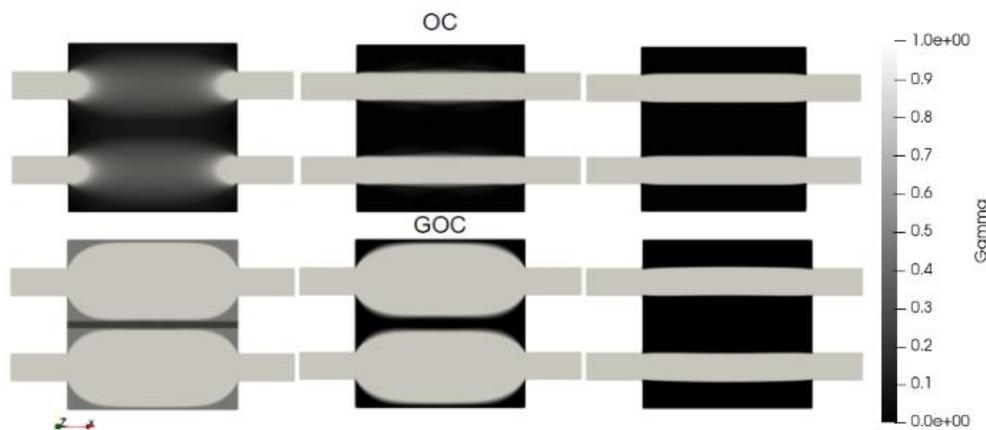


Figure 4.4: Performance comparison of the optimizer schemes.

### Effect of the initial configuration

Let us check now the effect of an initial configuration in this simple problem. The intuition suggests that an optimum could be found with the double pipe configuration. Results (4.4) actually showed that the double pipe layout provide a better compliance than the other minima found by OC and GOC methods, but can be important to check whether the OC and GOC methods converge to same solution by starting from a different initial condition. For this reason we try the OC and GOC methods with an initial low porosity layer between the two channels (see Figure (4.5), to see whether they keep or remove the separation in the optimization process. With this initial settings both the algorithms converge exactly to the double pipe configuration Figure(4.6), with a final compliance of  $J = 17.38$  for the Optimality Criteria, and  $J = 17.55$  for the Generalized Optimality Criteria, confirming that this is really the optimum configuration, independently on the used scheme.

Figure 4.5: *Initial material configuration.*Figure 4.6: *Snapshots of optimization procedure for the steady double pipe example in Figure (3.4) for initial condition of Figure(4.5). From the left, configurations at iteration 2, 5, and final result.*

The results of the second scenario ( $L = 1.5\text{m}$ ) are reported in Figures (4.7) and (4.8). In this case, the literature proposes a best layout formed by a merge of the two entering channels (see Figure 4.1). Similar configurations are in fact reached by the OC and GOC algorithms, while the MMA preserve the two separate channels in its final layout. The best result in terms of compliance is obtained with the OC method,  $J = 48.92$ , with a slightly unfeasible configuration (43.50% of the initial volume  $V_0$ ). In contrast to the case  $L = 1\text{m}$ , the MMA has here the worst layout configuration,  $J = 74.57$ , at a volume of 34% of  $V_0$ . Finally, GOC final configuration gives  $J = 54.00$  and a volume of 32.22% of  $V_0$ . Performance results are showed in Figure (4.9).

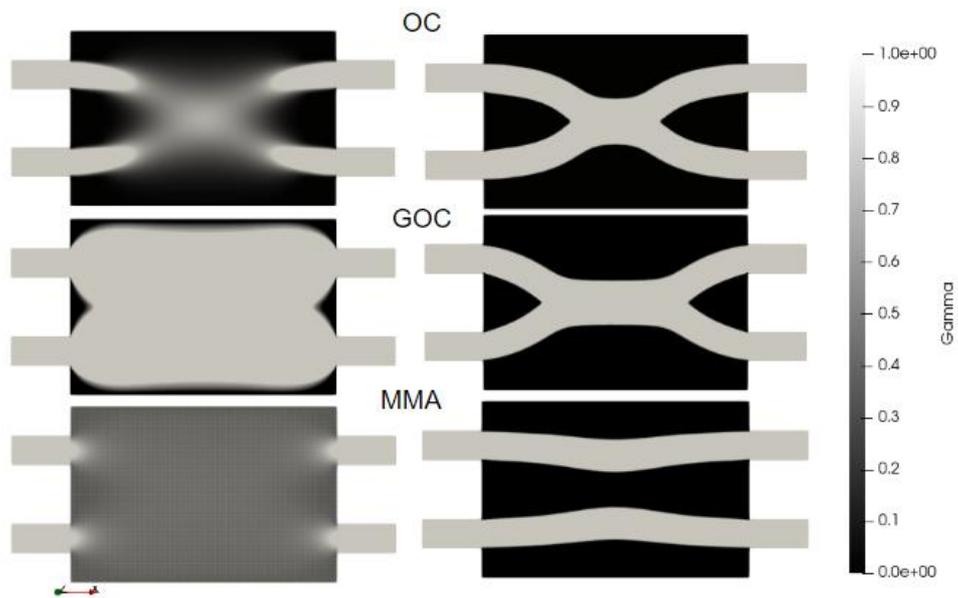


Figure 4.7: Snapshots of optimization procedure for the steady double pipe example in Figure (3.4) for  $L = 1.5\text{m}$ . From the left, configurations at iteration 5 and final result.

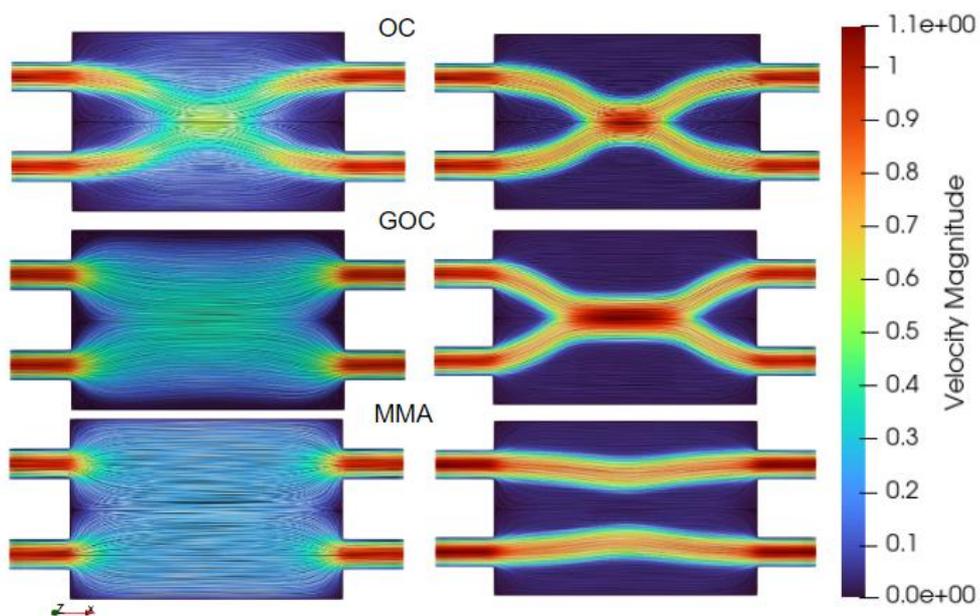


Figure 4.8: Snapshots of flow field at the optimization procedure for the steady double pipe example in Figure (3.4) for  $L = 1.5\text{m}$ . From the left, configurations at iteration 5 and final result.

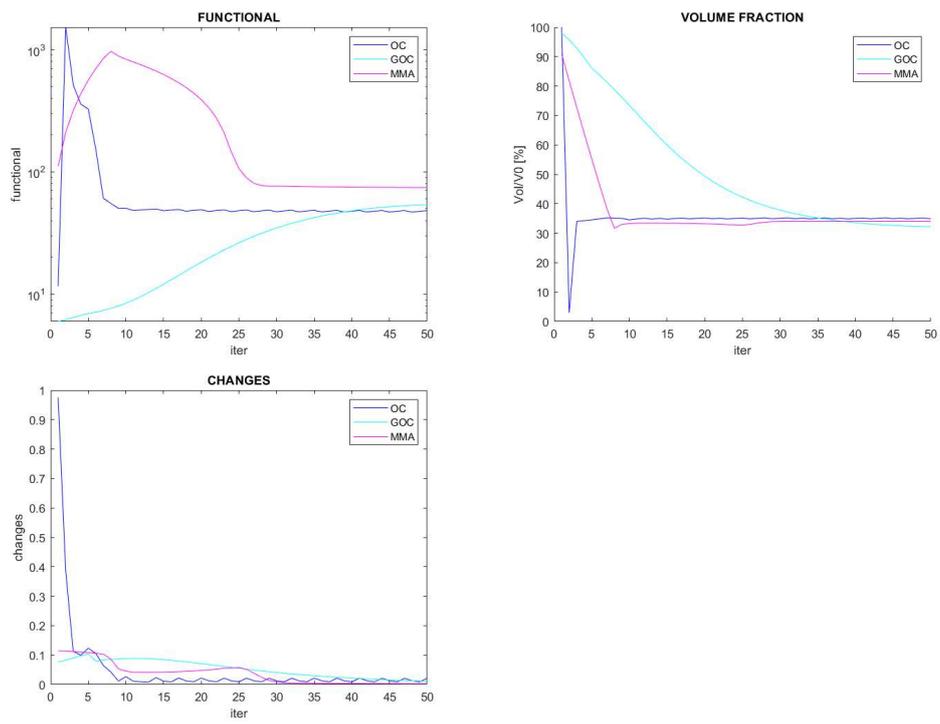


Figure 4.9: Performance comparison of the optimizer schemes.

### The Unsteady flow optimization

The solution of the unsteady double pipe example of Figure (6.14) has been proposed by [45] as shown in Figure (4.10).



Figure 4.10: *Optimal layout, from [32].*

The time-dependent flow field at the first iteration (free channel) is reported in Figure (4.11). The optimization results for the three simulations are presented in Figure(4.12). As in the steady double pipe example (3.4), the OC algorithm converges in few iterations (9), but slightly exceeding the volume constraint (34.86% of  $V_0$ ). The MMA result is instead identical to the reference one of Figure(4.10), and perfectly satisfy the imposed constraint ( $V = 34\%$  of  $V_0$ ). Lastly, the performance plots of Figure (4.13), shows that the GOC scheme could not reach convergence in the imposed 30 iterations, providing a final unfeasible solution ( $V = 37\%V_0$ ).

The best compliance is found by the OC algorithm ( $J = 0.40$ ), followed by MMA ( $J = 0.46$ ) and GOC ( $J = 0.49$ ).

The resulting flow path associated to the final MMA configuration is shown in Figure (4.14).

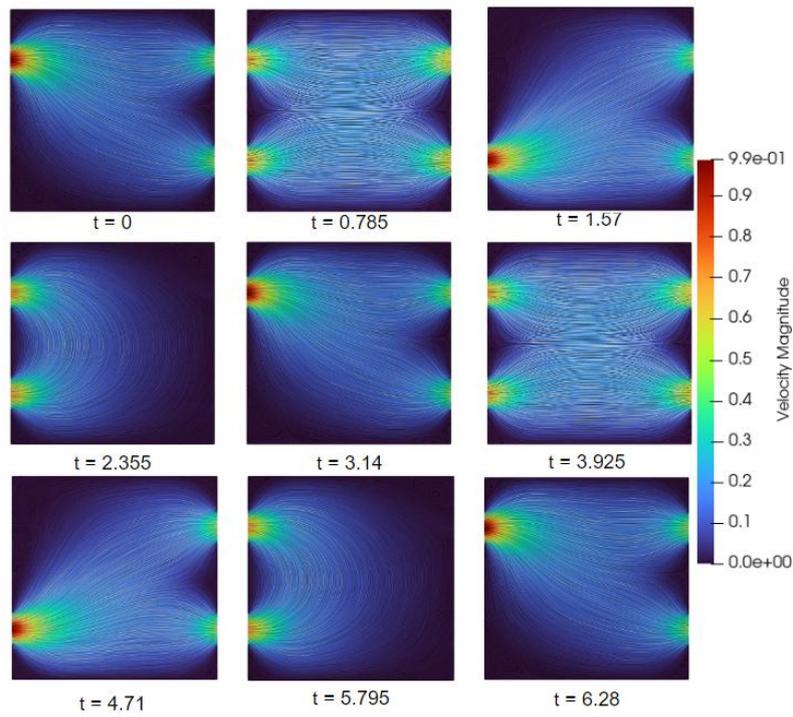


Figure 4.11: Snapshots of flow field the initial configuration for the unsteady double pipe example in Figure (6.14).

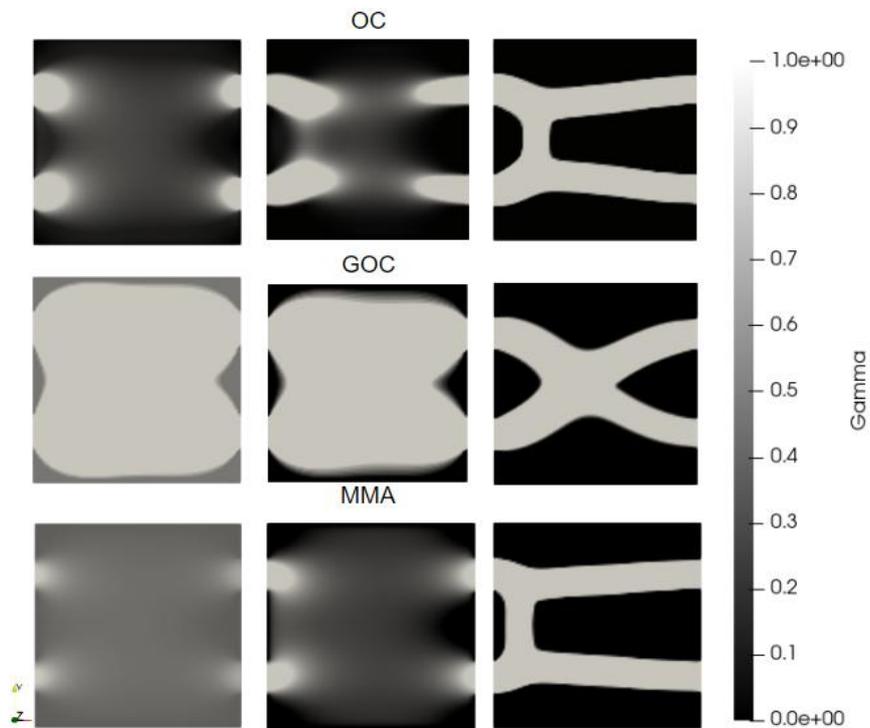


Figure 4.12: Snapshots of optimization procedure for the unsteady double pipe example in Figure (6.14). From the left, configurations at iteration 2, 5 and final result.

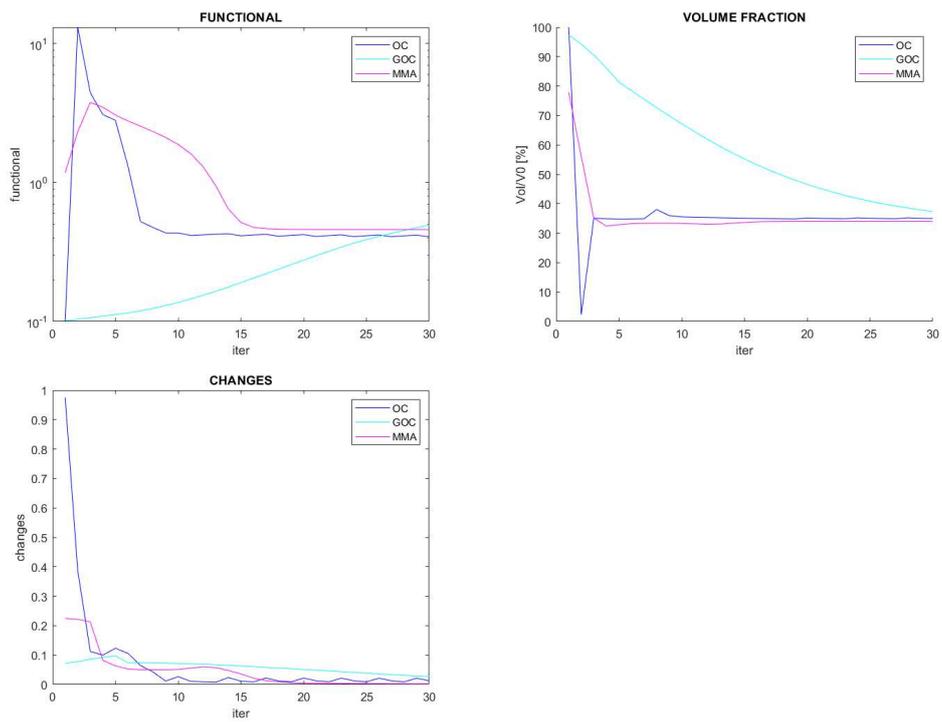


Figure 4.13: Performance comparison of the optimizer schemes.

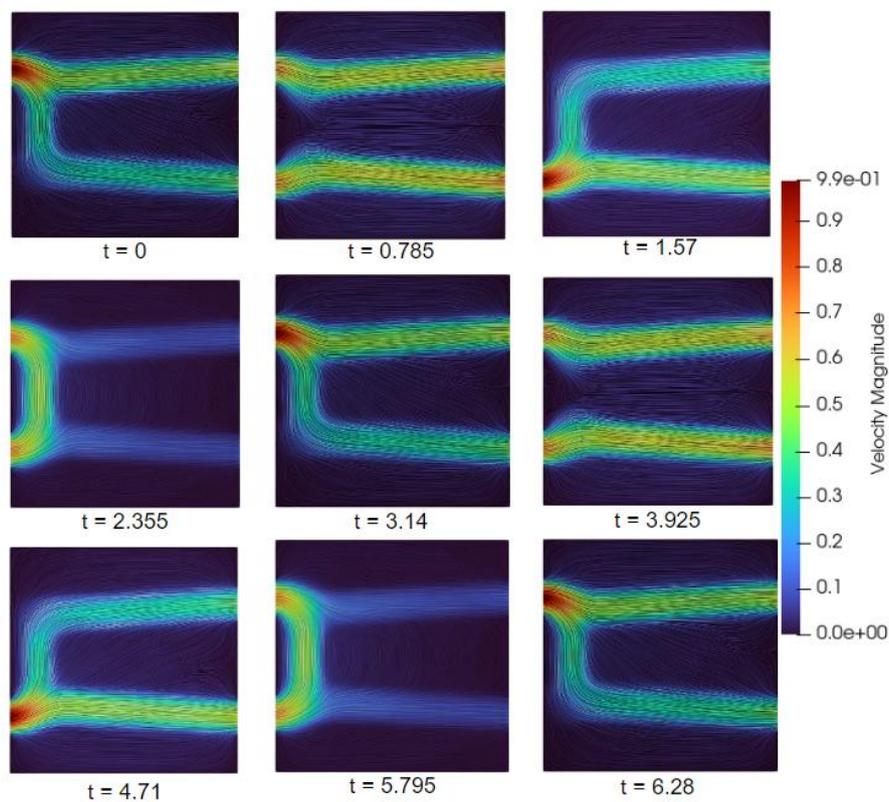


Figure 4.14: Snapshots of flow field at the final MMA configuration of Figure(4.12) for the unsteady double pipe example in Figure (6.14).

### The 4 Terminal device

The solution of the unsteady double pipe example of Figure (3.6) has been proposed by [16] as shown in Figure (4.15).

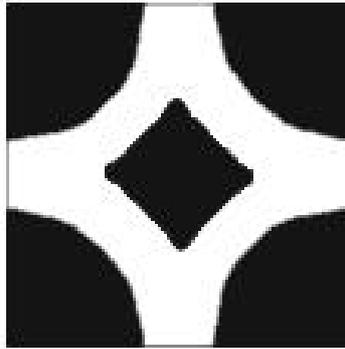


Figure 4.15: *Optimal layout for the 4 terminal device [16].*

The results of the optimization process are reported in Figures (4.16) and (4.17). The solution of the OC method almost matches the reference configuration in Figure (4.15), with a final compliance of  $J = 13.75$  at the 41.28% of  $V_0$ , and a convergence in 7 iterations. However, it is important to remark that all the three schemes are able to change the topology of the initial domain, by filling a box of solid material at the center of the configuration. The MMA method converges in 15 iterations to  $J = 2175$  and a volume of 39.99% of  $V_0$ , while the GOC method gives  $J = 28.36$  and  $V = 38\%V_0$  in more than 40 iterations. The optimization progress for the three algorithm is showed in Figure (4.18).

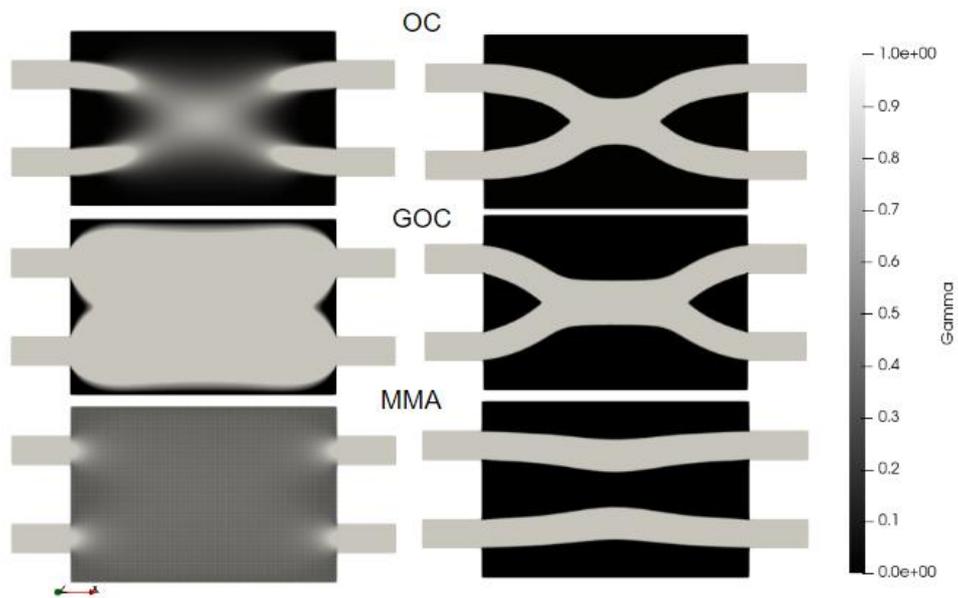


Figure 4.16: Snapshots of optimization procedure for the 4 device example in Figure (3.6). From the left, configurations at iteration 5 and final result.

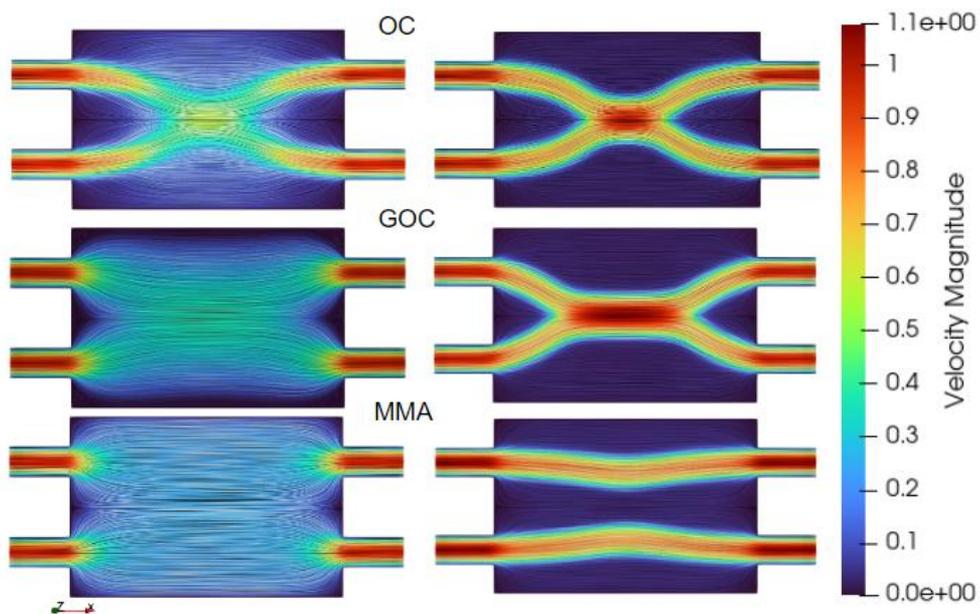


Figure 4.17: Snapshots of flow field at the optimization procedure for the 4 device example in Figure (3.6). From the left, configurations at iteration 5 and final result.

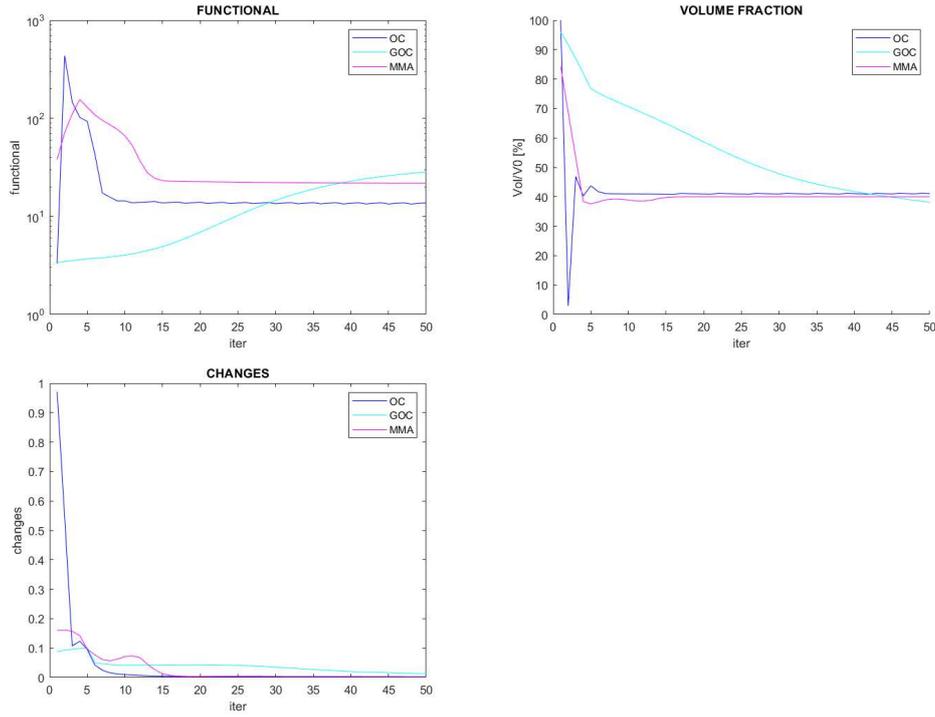


Figure 4.18: Performance comparison of the optimizer schemes.

### The Bend channel

The reference layouts for the two Reynolds numbers are shown in Figure (4.19). The paper of Deng [16] actually study this configuration with an additional term on the objective functional, given by the pressure work at the inlet, i.e.,

$$J = \int_0^T \int_{\Omega} J_{diss} + \int_0^T \int_{\Gamma_i} \beta p d\Gamma_i,$$

with  $J_{diss}$  the dissipation power given by Equation (2.2.6),  $\Gamma_i$  the inlet boundary, and  $\beta = 0.1$ . The final configurations of the present model may thus show some discrepancies, but we do not expect substantial differences in the model response with  $Re$  varying.

The resulting optimal layouts for the two scenarios are showed in Figures (4.20) and (4.21).

In the case  $Re = 1$ , the optimal configurations are all extremely similar, with compliance and volume of 20.26 and 25.26% $V_0$  for the OC method, 22.41 and 23.71% $V_0$  for the GOC method, and 21.01 and 25% $V_0$  for the MMA.

The case  $Re = 100$  present some instabilities on the final configurations, with a non clear solid/fluid interface. The results with the three algorithms are again extremely similar, with compliance and volume of  $3.45 \times 10^6$  and 25.08% $V_0$  for the OC method,  $1.24 \times 10^6$  and 32.8% $V_0$  for the GOC method, and  $3.12 \times 10^6$  and 25% $V_0$  for the MMA.

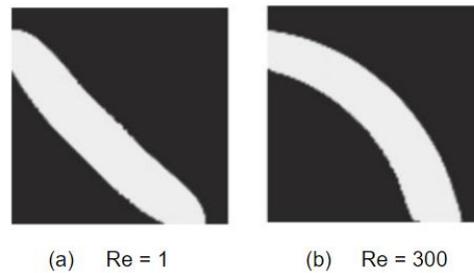


Figure 4.19: Optimal design at different Reynolds numbers [16], with an objective

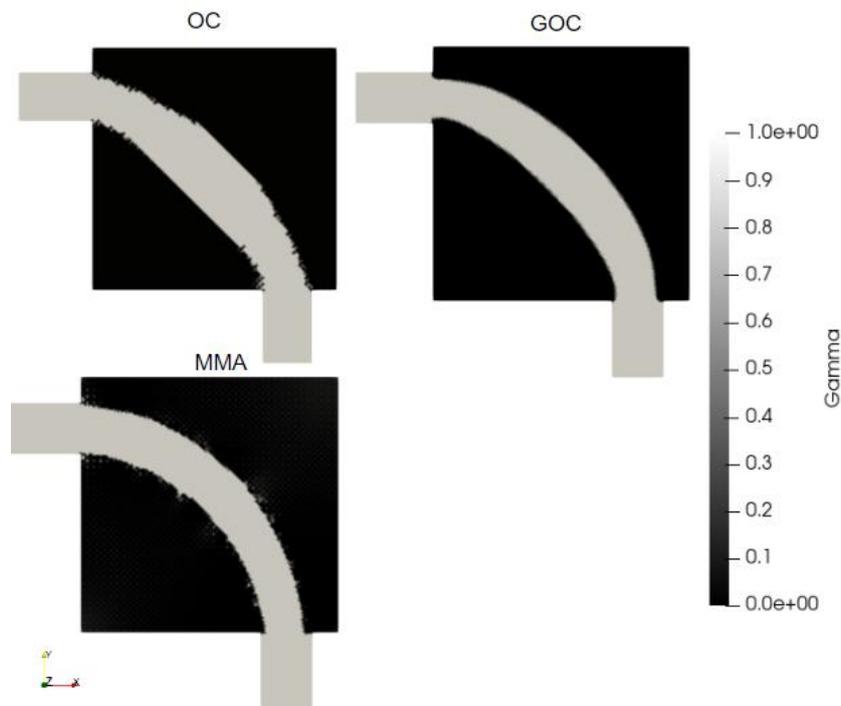


Figure 4.20: Optimal layouts for the bend channel example in Figure (3.7) at  $Re = 1$ . From the left, configurations at iteration 5 and final result.

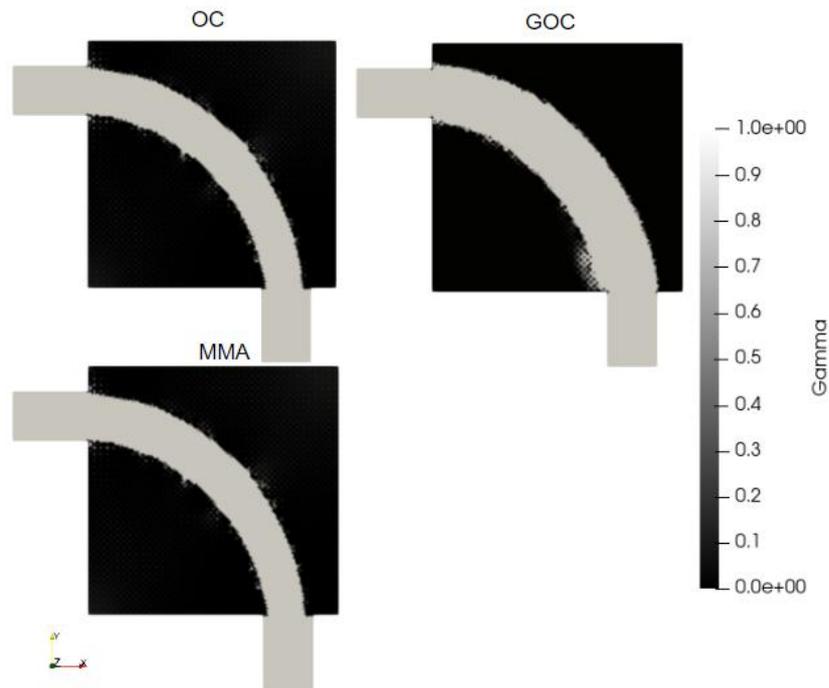


Figure 4.21: Optimal layouts for the bend channel example in Figure (3.7) at  $Re = 300$ .

### Choice of the functional

It is worth to stress how all these solutions are functional-dependent. Each choice of the functional has different effect on the evolution of the domain. Let us consider for example the problem in (3.6), by introducing with a body force  $\mathbf{f}_y = -10N$ , i.e., we consider the effect of gravity. The reference result in the literature is obtained in [16] for the maximization of the work of the body force, as shown in Figure (4.22). The result is clearly different with the minimization of the dissipation power (Figure 4.23). In the latter case, the gravity force slows down the fluid velocity and decrease the shear-wall stress, forcing the outflow in the upper outlet rather the the bottom, as expected by [16] Results are of  $J = 56.59$  and  $V = 41.10\%V_0$  for the OC method,  $J = 56.73$  and  $V = 38.1\%V_0$  for the GOC method, and  $J = 54.39$  and  $V =$

40.00% for the MMA.



Figure 4.22: Optimal layout, from [45].

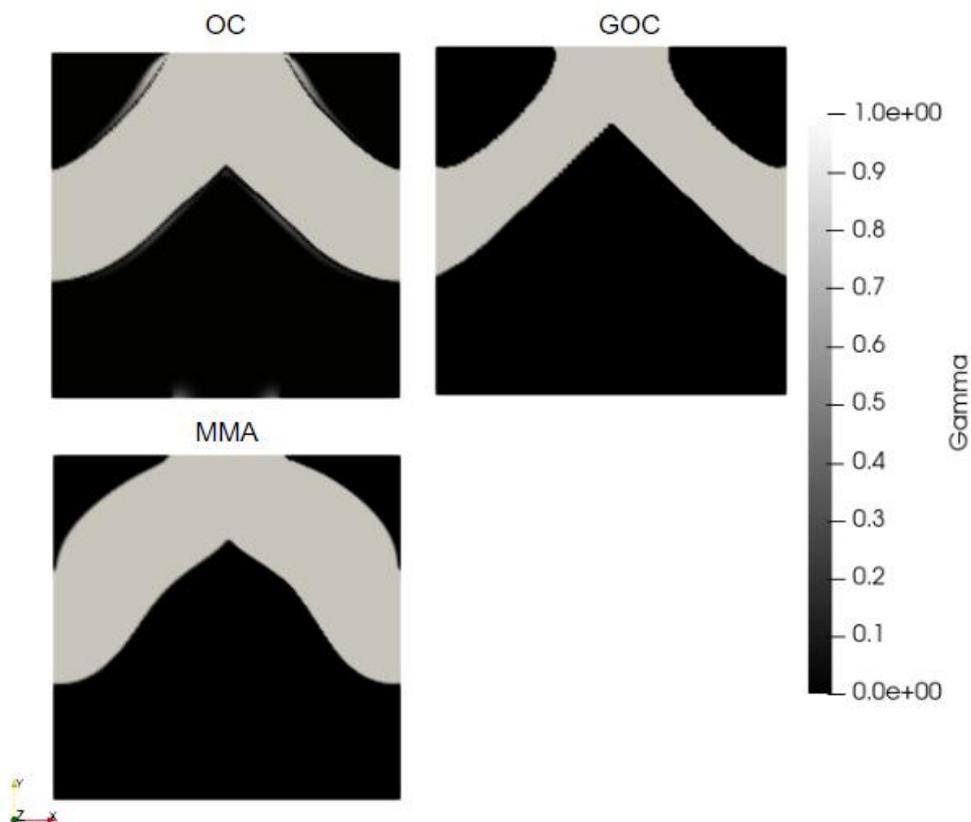


Figure 4.23: Optimal layouts for the 4 terminal device under gravity (4.22).

The gravity force is here used to slow the fluid motion and decrease the shear stress on the walls, forcing the fluid to exit from the upper outlet rather than the bottom one.

## 4.1.2 3D validation

### The Bend Channel

The solution of the three-dimensional bend channel example of Figure (3.8) has been proposed by [43] as shown in Figure (4.24).

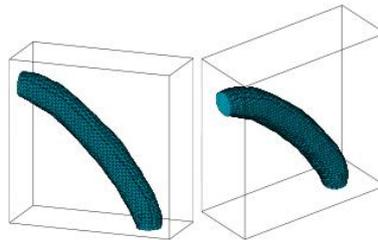


Figure 4.24: Optimal layout of the three-dimensional bend channel [43].

The results of the optimization are shown in Figure (4.25) for the scenario  $Re = 1$ , and in Figure (4.26) for the scenario  $Re = 50$ . In the case  $Re = 1$ , the final results read as:  $J = 87.15$ ,  $V = 10.73\%V_0$  for the OC method,  $J = 68.84$ ,  $V = 12.11\%V_0$  for the GOC method, and  $J = 74.51$ ,  $V = 9.99\%$  for the MMA.

On the other hand, results for the case  $Re = 50$ , are of  $J = 1.81 \times 10^5$ ,  $V = 12.41\%V_0$  for the OC method,  $J = 6.2 \times 10^4$ ,  $V = 21.68\%V_0$  for the GOC method, and  $J = 4.29 \times 10^5$ ,  $V = 10.0\%$  for the MMA.

The change of the Reynolds number produce an increment to the curvature of the resulting channel, particularly visible for the results of the GOC method.

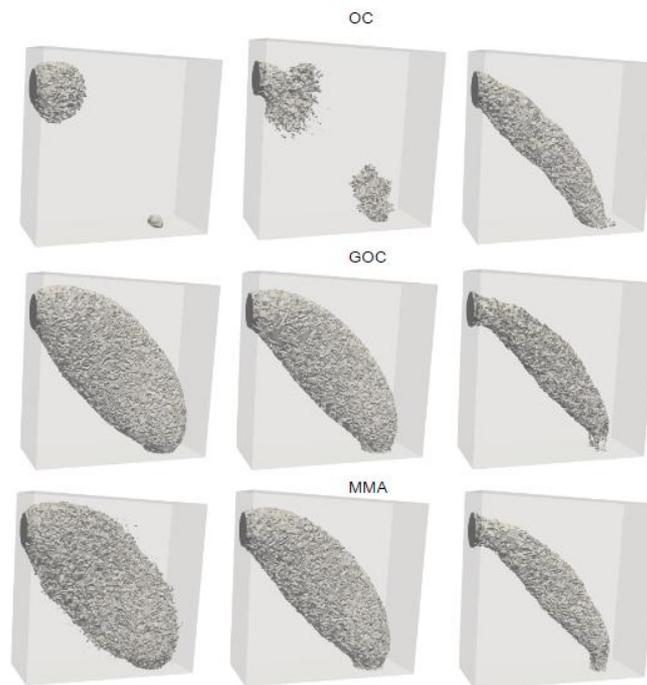


Figure 4.25: Snapshots of optimization procedure for the three-dimensional bend channel device example in Figure (3.8) at  $Re = 1$ . From the left, configurations at iteration 2, 5, and final result.

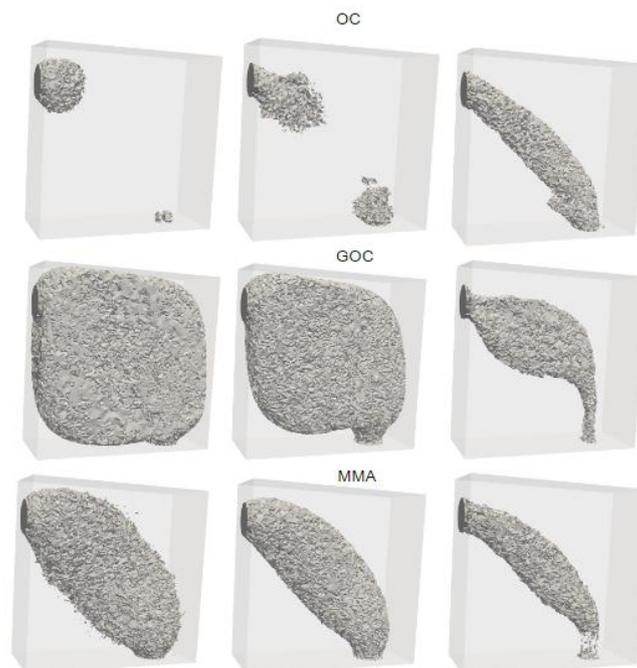


Figure 4.26: Snapshots of optimization procedure for the 6 device example in Figure (3.8) at  $Re = 50$ . From the left, configurations at iteration 2, 5, and final result.

## 6 Terminal device

The solution of the 6 terminal device example of Figure (3.9) has been proposed by [43] as shown in Figure (4.27). The optimization process is shown in Figures (4.28) (4.29). The OC is the only method that was able to reach the true best configuration, with  $J = 3.21 \times 10^4$ , at  $V = 10.44\%V_0$ . The MMA algorithm stops instead at a local minimum that only includes an incipient formation of such central hole, as proved by the streamlines plot in the final configuration (see Figure (4.29), reaching a final state with  $J = 8.31 \times 10^4$  and  $V = 9.90\%V_0$ . Finally, the GOC method does not present at all the hole, achieving the worst final compliance,  $J = 8.84 \times 10^4$  at a volume  $V = 7.85\%$ . The final configurations for the entire geometry are shown in Figure (4.30), while the performance graphs are in Figure (4.31).

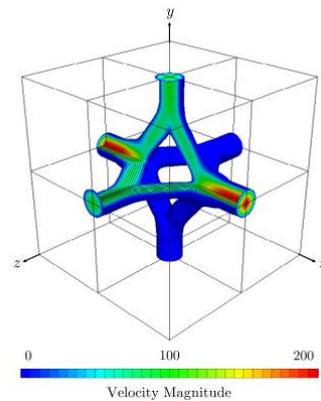


Figure 4.27: Optimal layout of the 6 terminal device [43].

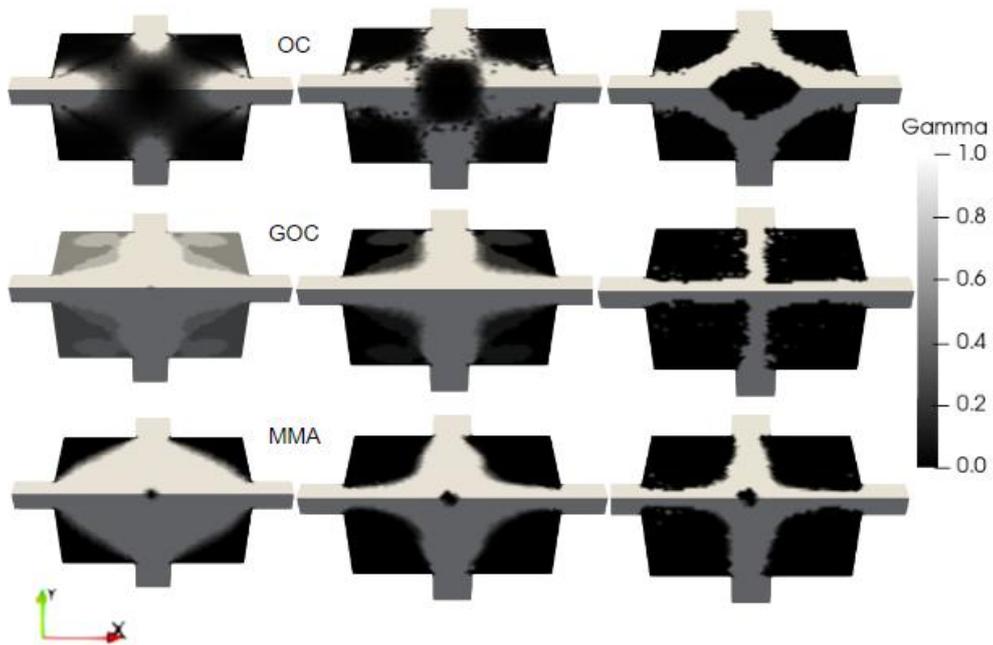


Figure 4.28: Snapshots of optimization procedure for the 6 device example in Figure (3.9). From the left, configurations at iteration 2, 5, and final result.

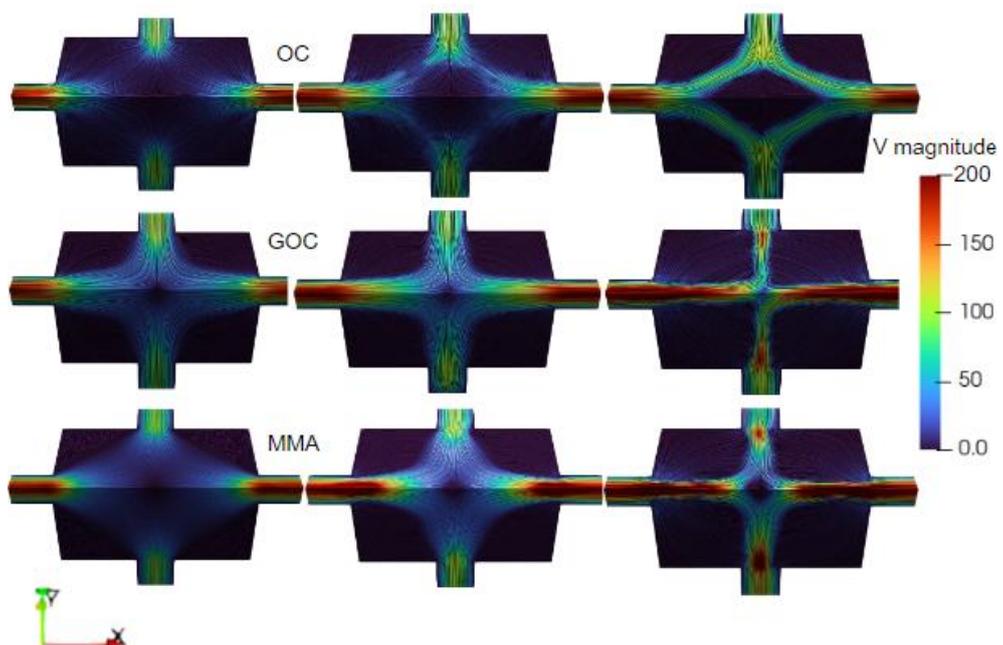


Figure 4.29: Snapshots of flow field at the optimization procedure for the 6 device example in Figure (3.9). From the left, configurations at iteration 2, 5, and final result.

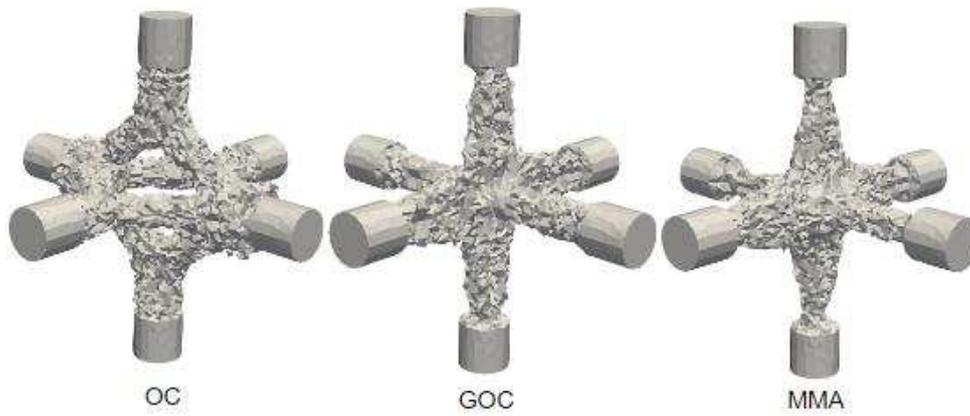


Figure 4.30: Resulting optimal configurations for the 6 device example in Figure (3.9).

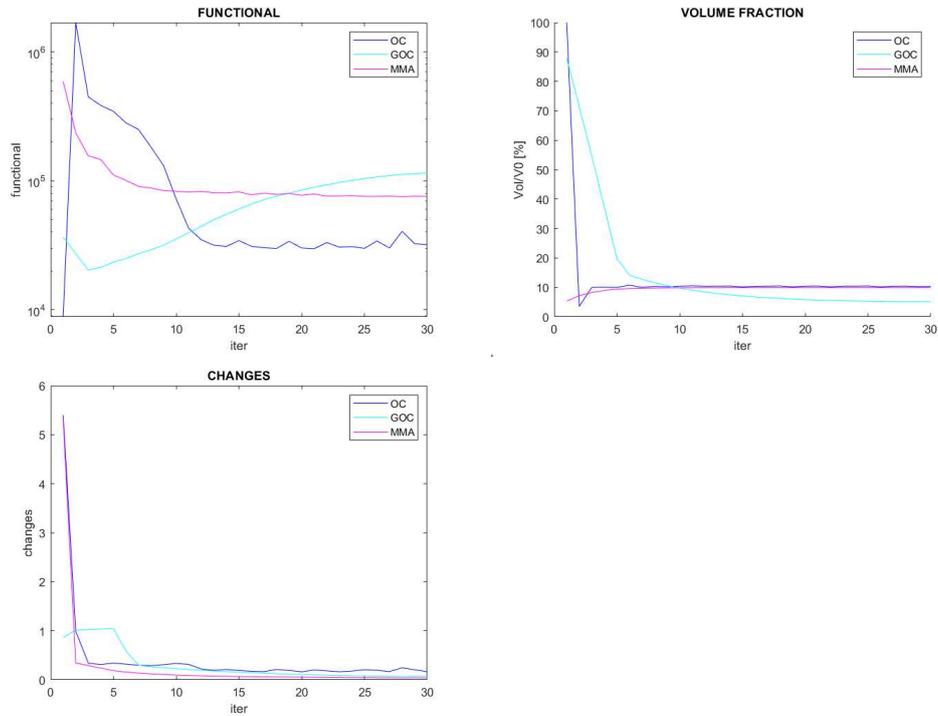


Figure 4.31: Performance comparison of the optimizer schemes.

### 4.1.3 Some final considerations: Checkerboard problem

The checkerboard pattern is one of the main computational and theoretical issues for a topology optimization algorithm. A checkerboard is defined as a

periodic pattern of high and low values of pseudo-densities,  $\gamma$ , arranged in a checkerboards fashion. This behaviour is undesirable as it results from numerical instabilities. The checkerboards possess artificially low porosity and prevent the fluid from entering specific regions of the domain, but does not correspond to an optimal distribution of material.

For the developed algorithm this problematic has proven to be particularly relevant when  $Re > 100$ , and some examples are shown in Figure (4.32).

This issue could be overcome with a proper density filter that force a continuous distribution of the values of  $\gamma$  inside the domain [44].

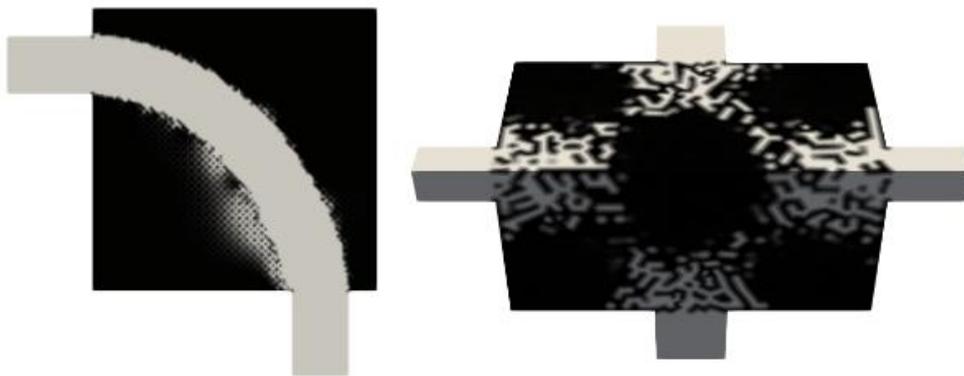


Figure 4.32: Examples of checkerboard patterns in the developed examples.

#### 4.1.4 Optimizer schemes comparison

Apart from the the double pipe and the 6-terminal device problems, all the schemes (OC, GOC, and MMA) converge to topology configurations similar to the optimal layouts reported in the literature.

However, some important differences must be stressed. Among the three, the Optimality Criteria has proved to be the more solid and efficient (4 best layouts out of 7 for the 2D scenario, and 2 out of 3 for the 3D), with final configurations very close to the guessed optimum solution. Further, the convergence is very fast, reaching the optimum in very few iterations (approximately less

than 10 iterations in both the 2D and the 3D scenarios).

The Generalized Optimality Criteria showed a very slow domain update, mirroring in a slow convergence or breaks in local minimum, when a good initial solution is not known.

Finally, the Method of Moving Asymptotes presents a rather fast convergence velocity, approximately 15/20 iterations for both the 2D and 3D scenarios, achieving satisfactory optimal solutions (best layouts out of 7 for the 2D scenario, and 2 out of 3 for the 3D). On the other hand, MMA solutions seem to be more dependent on the initial configuration than the OC, requiring a more accurate parameter calibration for each study.

Each of these schemes should be validated over different sets of optimization parameters and initial configurations, comparing all the local best layouts to find the true optimal configuration. Nonetheless, the interesting result is that, starting from a completely fluid domain, the Optimality Criteria algorithm was able to reach a final configuration extremely close to the optimal one, always providing at least a good sense of what should be the optimal layout of the considered problem.

## 4.2 TO for fluid flow and scalar transport problems

### 4.2.1 Testing: Bend Channel

**Case  $\xi = 0$**

The case  $\xi = 0$  exactly corresponds to the minimization of the power dissipation. The optimization process for the three optimization schemes is shown

in Figure (4.33). After 50 iterations all the schemes reach a similar configuration, and the best result is given by the MMA algorithm ( $J = 6.77$ ), that reached convergence in 10 iterations, with a volume fraction of 29.93% of  $V_0$ . The GOC algorithm could not reach convergence in the 50 iterations, exiting the simulation at  $J = 1.27$  with an unfeasible configuration (59.94% of the initial volume). Lastly, the GOC algorithm was able to converge in 9 iterations to a final configuration of 40.07% of the initial volume and a compliance of  $J = 7.25$ .

The pollute concentration was not included in the functional in this scenario, and the final values of the pollute concentration at the outlet (by  $L_2$  norm on the outlet) are 0.43 for the OC, 0.30 for the GOC, and 0.47 for the MMA. Performance results are shown in figure (4.34).

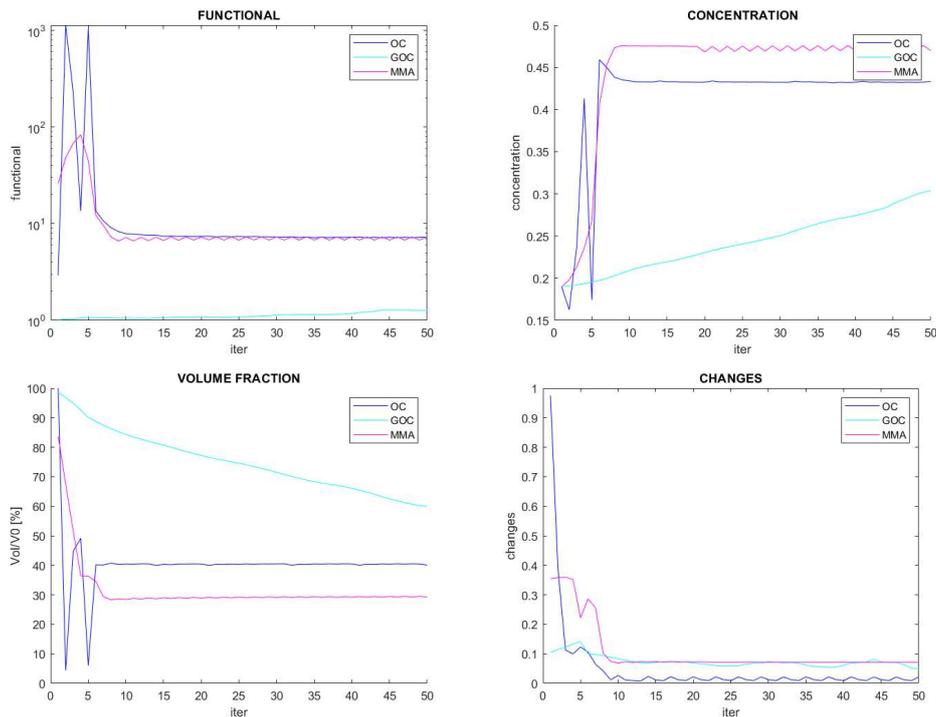


Figure 4.34: Snapshots of optimization procedure and effects on the flow field and pollute distribution for the bend channel example in Figure (3.10) for  $\xi = 0$ . Left column, Iteration 2, right column, final iteration.

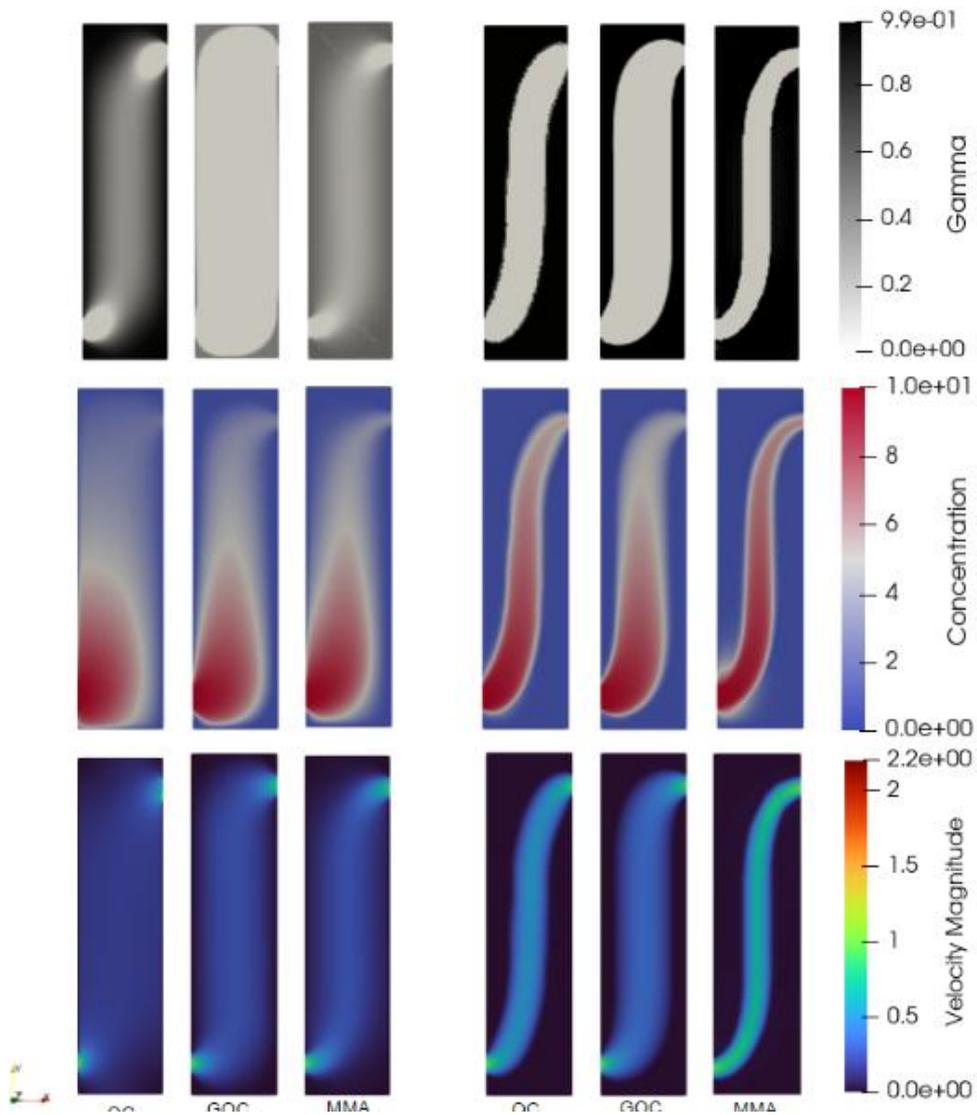


Figure 4.33: Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration for  $\xi = 0$

**Case  $\xi = 1$** 

The case  $\xi = 1$  corresponds to the case  $\beta_1 = \beta_2$ , equalizing the weights of the two components of the objective functional. The optimization process for the three optimization schemes is shown in Figure (4.35). After 50 iterations, the best compliance is now given by the OC algorithm ( $J = 8.10$ ), that reached convergence in 10 iterations, with a volume fraction of 40.29% of the initial volume. The GOC algorithm does not reach the convergence in 50 iterations, achieving  $J = 1.40$ , but showing an unfeasible configuration (57.76% of the initial volume). Lastly, the OC algorithm was able to converge in 9 iterations to a final configuration of 31.01% of the initial volume and a compliance of  $J = 19.88$ .

The pollute concentration was now included in the functional, and its final concentration at the outlet (by  $L_2$  norm on the outlet) are of 0.43 for the OC, 0.31 for the GOC, and 0.46 for the MMA. Performance results are shown in figure (4.36). From the results, the optimization process for all the three schemes, OC, GOC, and MMA, return almost the same results of the scenario with  $\xi = 0$ , proving that the energy dissipation is also in this case the most important driver of the optimization process, and only higher  $x_i$  should influence the model solution.

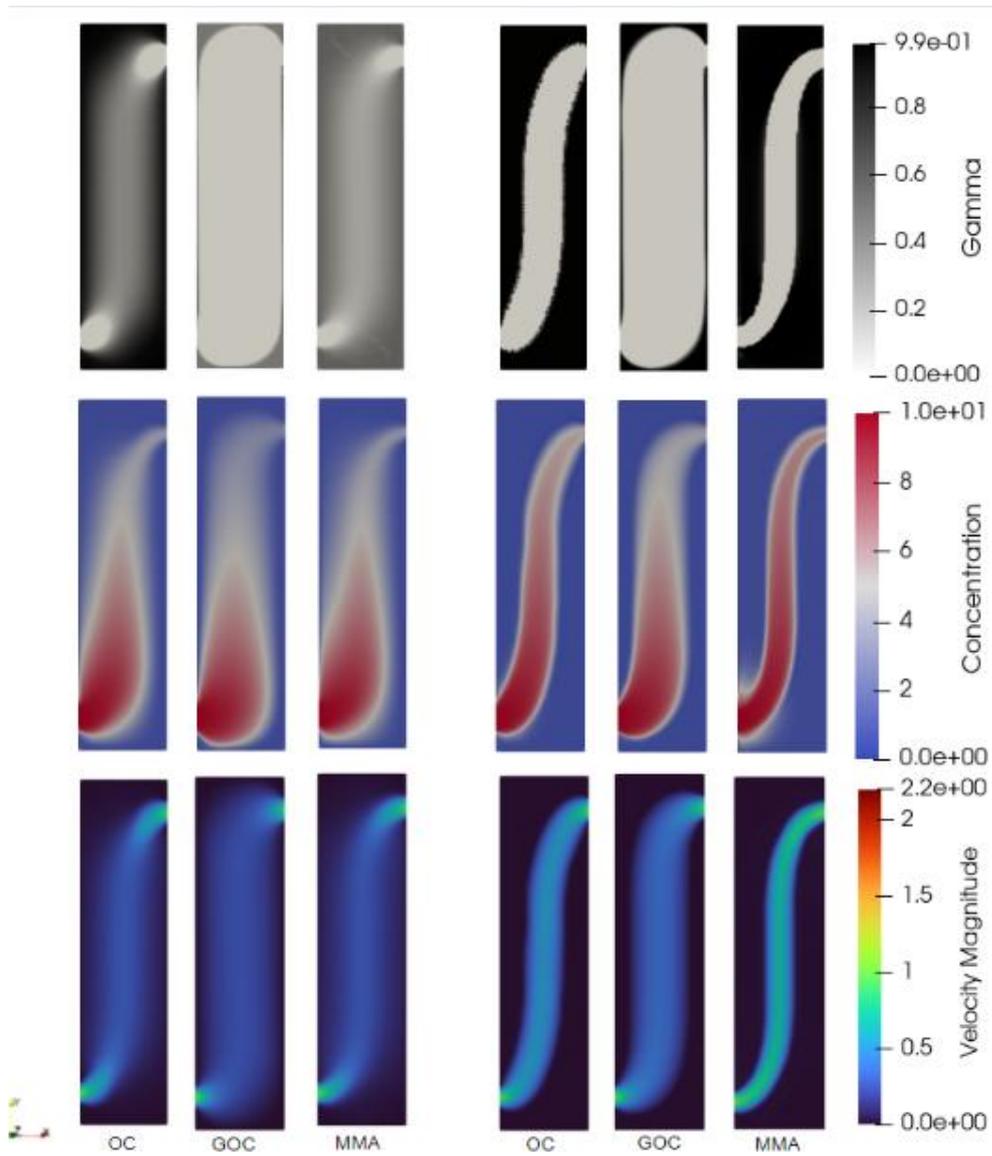


Figure 4.35: Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration for  $\xi = 1$ .

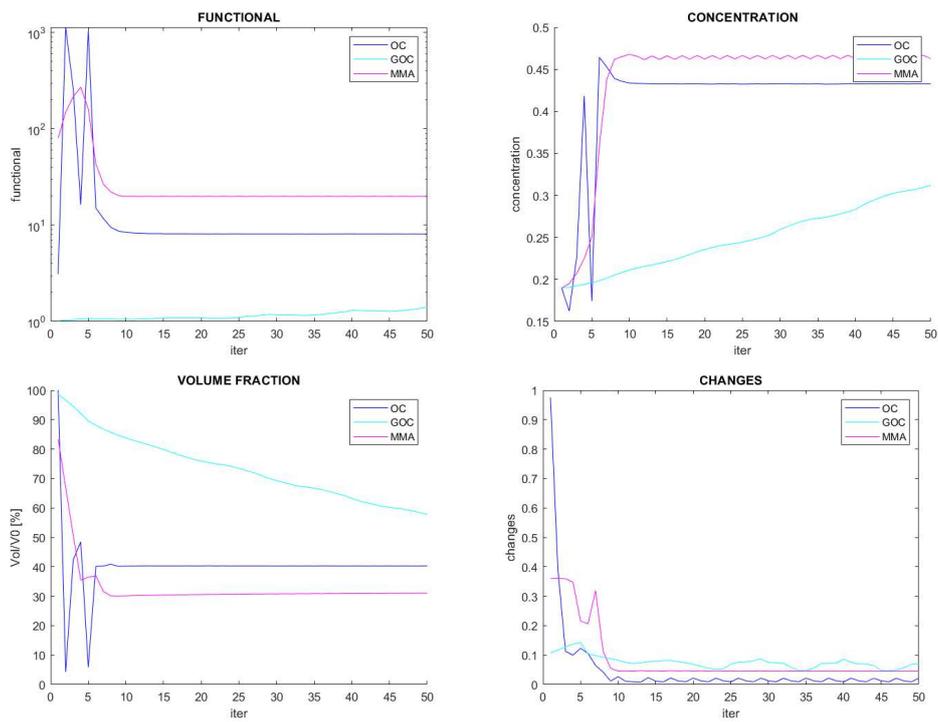


Figure 4.36: Snapshots of optimization procedure and effects on the flow field and pollute distribution for the bend channel example in Figure (3.10) for  $\xi = 1$ . Left column, Iteration 2, right column, final iteration.

**Case  $\xi = 10$** 

The case  $\xi = 10$  weights the minimization of the pollute concentration at the outlet 10 times more than the minimization of the energy dissipation. The optimization process for the three optimization schemes is shown in Figure (4.37). The result is in this scenario quite different with respect to the cases with  $\xi = 0$  and  $\xi = 1$ . After 50 iterations, the OC and GOC algorithms continue to converge to a single bend channel layout, while the MMA scheme moves to a multi-channel configuration, splitting the flow at the inlet in two narrow conducts, whose path follows the walls. Moreover an incipient formation of a channel in the middle of the domain is visible (see Figure 4.37). All the three solutions present some uncertainties from a physical perspective. The OC final configuration is a bend channel slightly larger and with different curvatures than the one for the case  $\xi = 1$  or  $\xi = 0$ , and provides the best compliance result, with  $J = 2.33$  and a final volume of 43.16% of  $V_0$ . Figure (4.37) shows some "dust" inside of the OC resulting channel, which is probably used by the algorithm to slow the fluid and increase the pollute dissipation without an energy-consuming channel division. The  $L_2$  norm of the concentration at the outlet is now of 0.41. This configuration is clearly non physical and only given by the numerical approximation, and should be excluded if a proper density filter was not applied.

The GOC algorithm finds instead a single, but very narrow channel, reaching now a very little fluid domain of 12.26% of the initial volume, lower in fact than the one imposed by the volume constraint. This configuration is likely not the best either for the pollution spreading and the energy conservation, since it forces the fluid to very high velocity close to the walls, and does not slow the fluid by longer paths of the channel or by a division into sub-channels, as one would expect intuitively. From the velocity magnitude analysis in Figure(4.37), it is clear that this layout allows a portion of the flow to move with

very low velocities even inside the regions with design variable  $\gamma = 0$ , in order to slow the fluid motion and decay more pollute by the sink rate. This configuration is again the effect of a numerical approximation, in this case the owing to the low value of  $\alpha_{max}$ , which does not prevent the fluid to pass through the computationally solid material. The final value for the objective functional with the GOC algorithm is  $J = 57.09$ , with the  $L_2$  concentration norm at the outlet of 0.25.

The MMA scheme provides in this case the worst configuration in terms of compliance,  $J = 35.52$ , with a volume of 21.55% of the initial one. The configuration shows a division of the initial flow into three channels, two attached to the walls, and a central one, non-complete, which stops before the center of the domain, and restarts right before the outlet. This layout is the extreme case of the solid permeability problem described for the GOC result, and is due to the low value of the damping parameter,  $\alpha_{max}$ . The pollute concentration at the outlet is in fact the minimal among the three schemes, with an  $L_2$  outlet norm of 0.17.

Performance results are shown in Figure (4.38). The optimization process for all the three schemes (OC, GOC, and MMA), now takes into account the effect of the pollute decayment, providing a decrease of the outlet concentration with respect to the previous scenarios.

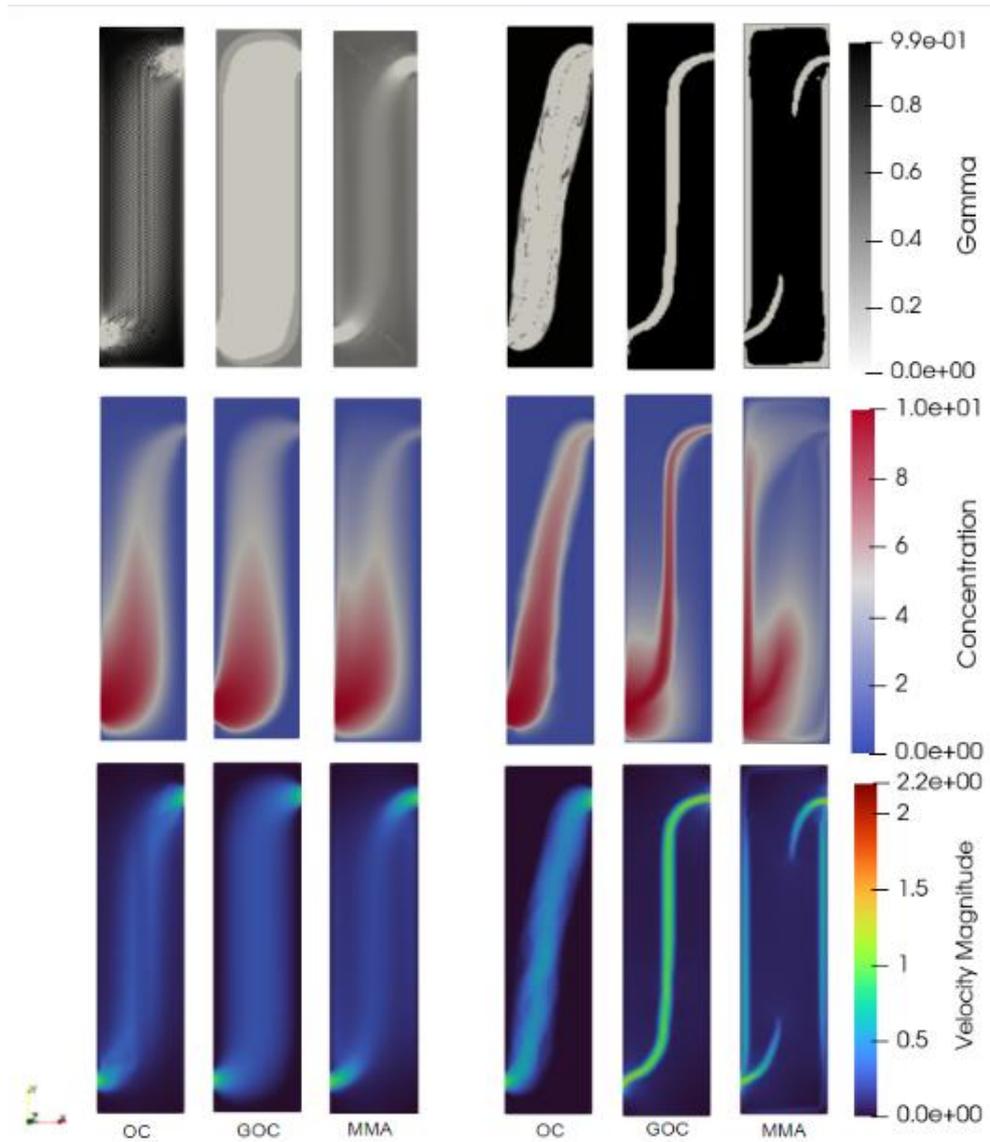


Figure 4.37: Snapshots of optimization procedure and effects on the flow field and pollutant distribution for the bend channel example in Figure (3.10) for  $\xi = 10$ . Left column, Iteration 2, right column, final iteration.

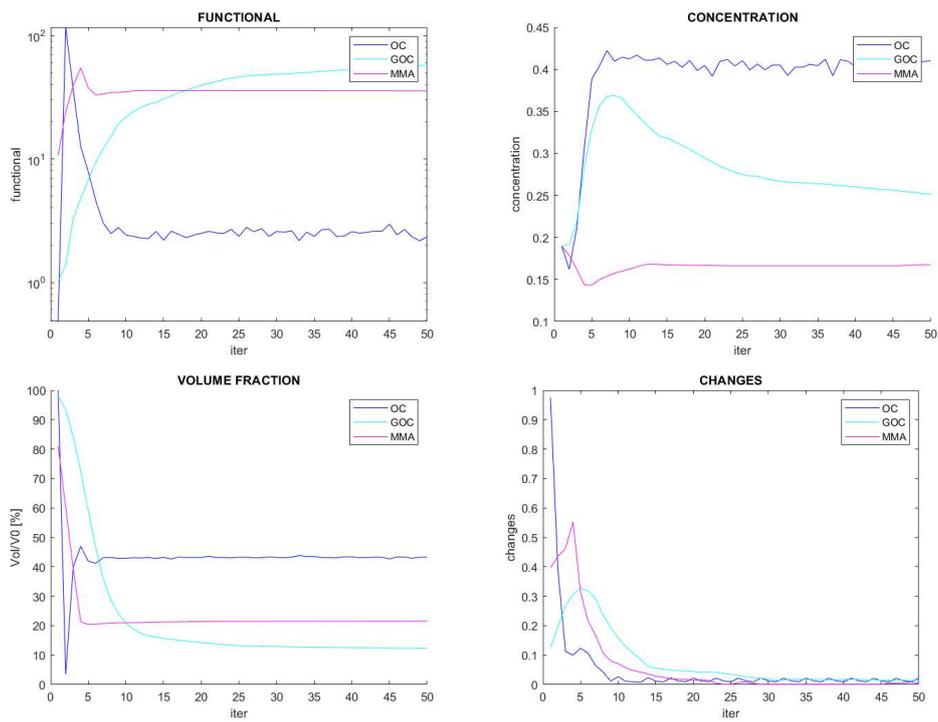


Figure 4.38: Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10) for  $\xi = 10$ .

**Case  $\xi = 100$** 

The case  $\xi = 100$  weights the minimization of the pollute concentration at the outlet 100 times more than the minimization of the energy dissipation. The optimization process for the three optimization schemes is shown in Figure (4.39). The results of the optimization show substantially the same features found in the case with  $\xi = 10$ . Each optimization scheme handles the weighted functional with different evolution processes.

The OC method still propose a single channel configuration, with a slightly different curvature than in the scenario with  $\xi = 10$ . The OC final layout is the best in terms of compliance,  $J = 0.78$ , with the 41.16% of the initial domain, but continue to preserve "density dust" in the formed channel, to slow the fluid.

The GOC scheme returns a final layout almost identical to the one found for  $\xi = 10$  (see Figure4.37), with a single, narrow channel, and a very slow velocity in the non-perfectly impermeable solid material. GOC final layout finds  $J = 16.62$  with only 11.01% of the initial domain occupied by fluid.

Lastly, the MMA scheme reaches in this configuration the exact division of the entering flow into two subchannels attached to the lateral walls, with a final compliance and volume of  $J = 4.01$  and  $17.96\%V_0$ , respectively. The increment of the  $\xi$  ratio produced the elimination of the nonphysical incomplete central channel (Figure4.37), reaching a first final configuration that, compared with Figure (4.33), is substantially different from the minimization of the energy dissipation.

Performance results are shown in figure (4.40). The resulting  $L_2$  concentration norm at the outlet is now of 0.4, 0.23, and 0.15 for the OC, GOC, and MMA algorithms, respectively, i.e., values in line with the solution with  $\xi = 10$ .

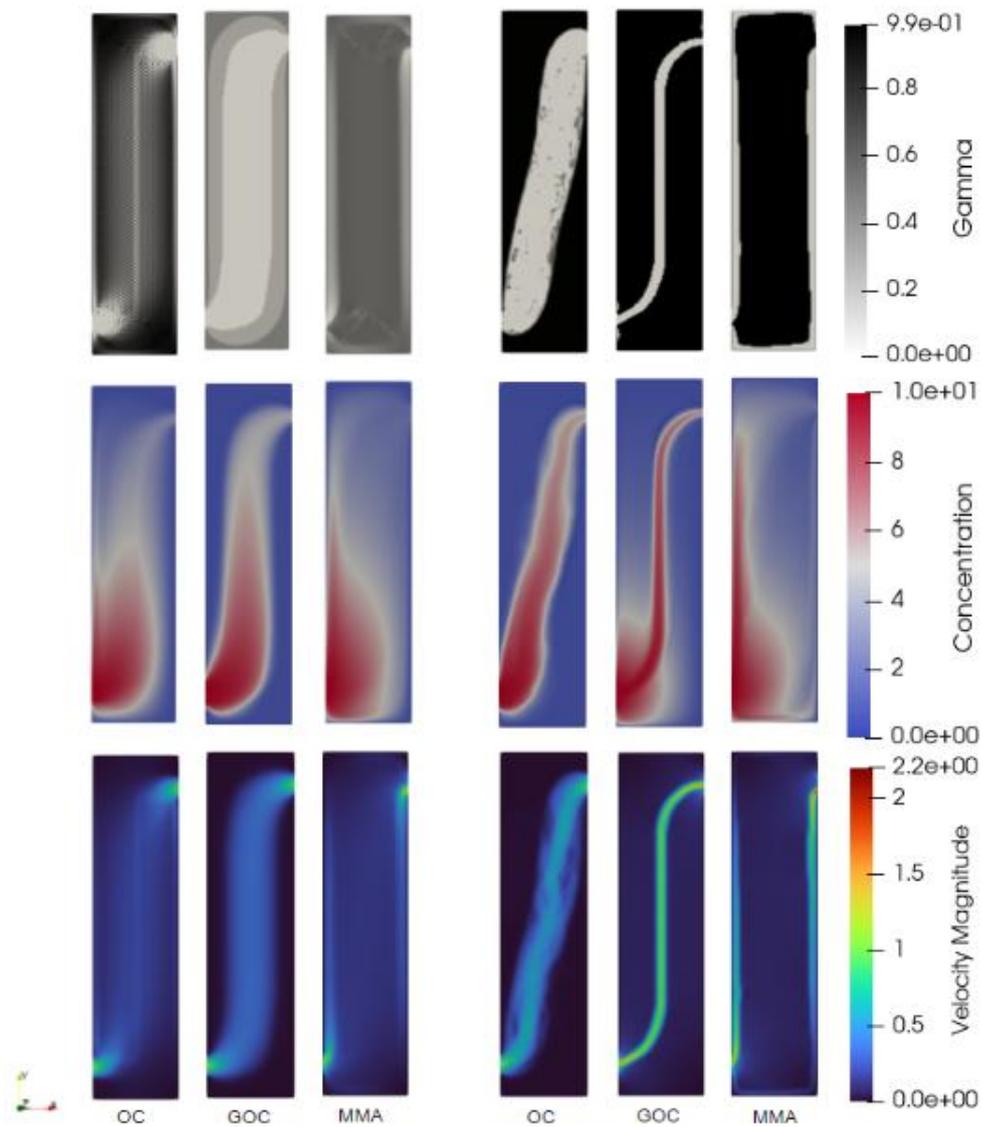


Figure 4.39: Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration.

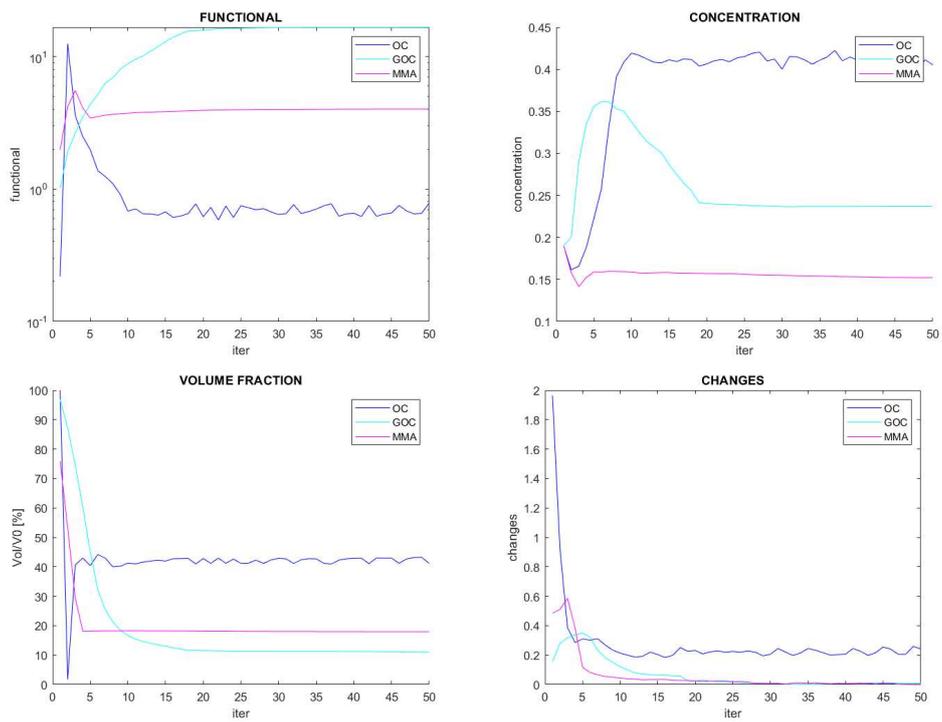


Figure 4.40: Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10).

**Case  $\xi = \infty$  ( $\beta_1 = 0$ )**

The case  $\xi = \infty$  considers only of the pollute removal in the objective functional, hence the  $L_2$  concentration norm at the outlet will exactly coincide with the functional value. The optimization process for the three optimization schemes is shown in Figure (4.41). Some important and interesting differences can be noted with respect to the previous scenarios.

The OC method moves from the single channel configuration shown in cases  $\xi = 0 \rightarrow 100$  to a nonphysical solution that exploit the slightly permeability of the fictitious solid material. The proposed solution can be described as an incomplete bend channel, with a small layer of porous material at the center, which divide the two sides of the channel, and causes a slow down of the fluid velocity and a reduction of the pollute mass. The final configuration for the OC algorithm provides the worst compliance among the three schemes ( $J = 0.28$ , and a volume of 41.05% of the initial domain).

The GOC and the MMA algorithms reaches almost the same configuration, practically equivalent to the one found by the MMA scheme for  $\xi = 100$  (see Figure 4.39), showing the formation of two channels, placed along the domain boundaries. This path is the longest connection route between the inlet and the outlet, which does not require the complete inversion of the flow direction of motion. It is important to note that with this almost permeable material, this configuration is not possible, since the fluid would simply move through the computationally solid material. The next paragraph will show what may happen with a larger value of the damping parameter,  $\alpha_{max}$ .

While the MMA configuration for  $\xi = 100$ , (Figure 4.39) still shows some fluid motion inside the solid material, the channels obtained here, for  $\xi = \infty$ , i.e.  $\beta_1 = 0$ , seem to be more sharp and defined, (Figure 4.41). Final layouts for MMA and GOC schemes give  $J = 0.14$  and  $J = 0.12$ , for final volumes of 19.78% and 24.41% of the initial domain, respectively.

Performance results are shown in figure (4.42).

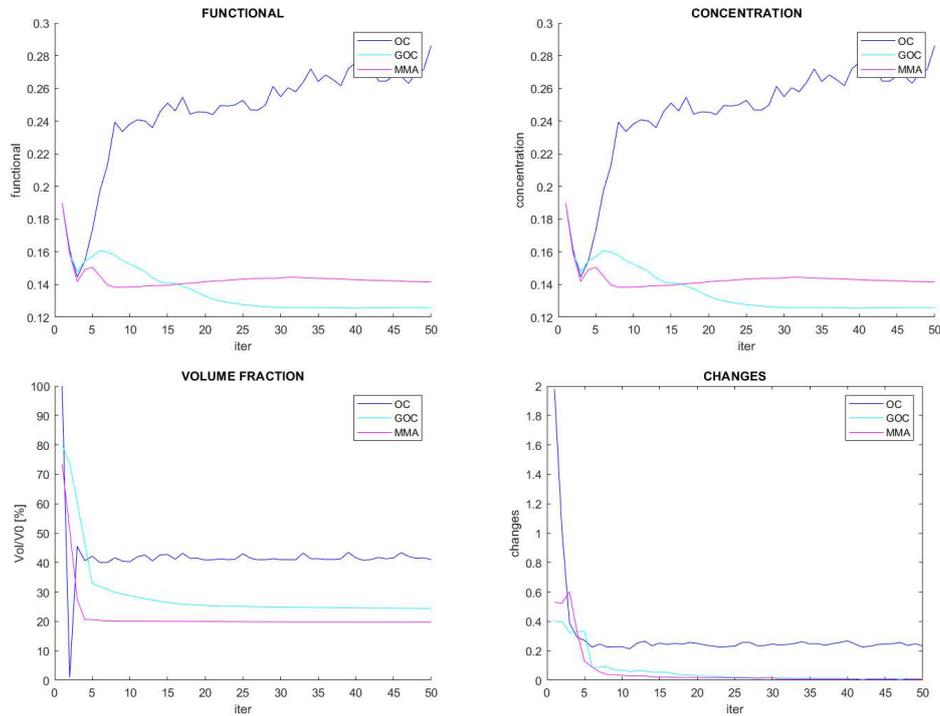


Figure 4.42: Performance comparison of the optimizer schemes for the bend channel example in Figure (3.10).

### 4.2.2 Instabilities and checkboard effect

The tests over the bend channel problem have proved that the developed Topology Optimization algorithm can find some heuristically good solutions, according to the prescribed computational physics, even for the flow-driven scalar transport problem of reductive pollute. In this fraim, the main problem is given by the non completely impermeable porous material at the solid region, which allows the fluid to move into unrealistic regions of the domain.

The main issue with the increment of the damping parameter,  $\alpha_{max}$ , is given by the formation of checkboard patterns in the solution.

While the fluid-flow topology optimization problem has been proved to be well-posed at least for Stokes flow (Borrvall and Petersson [14]), even without

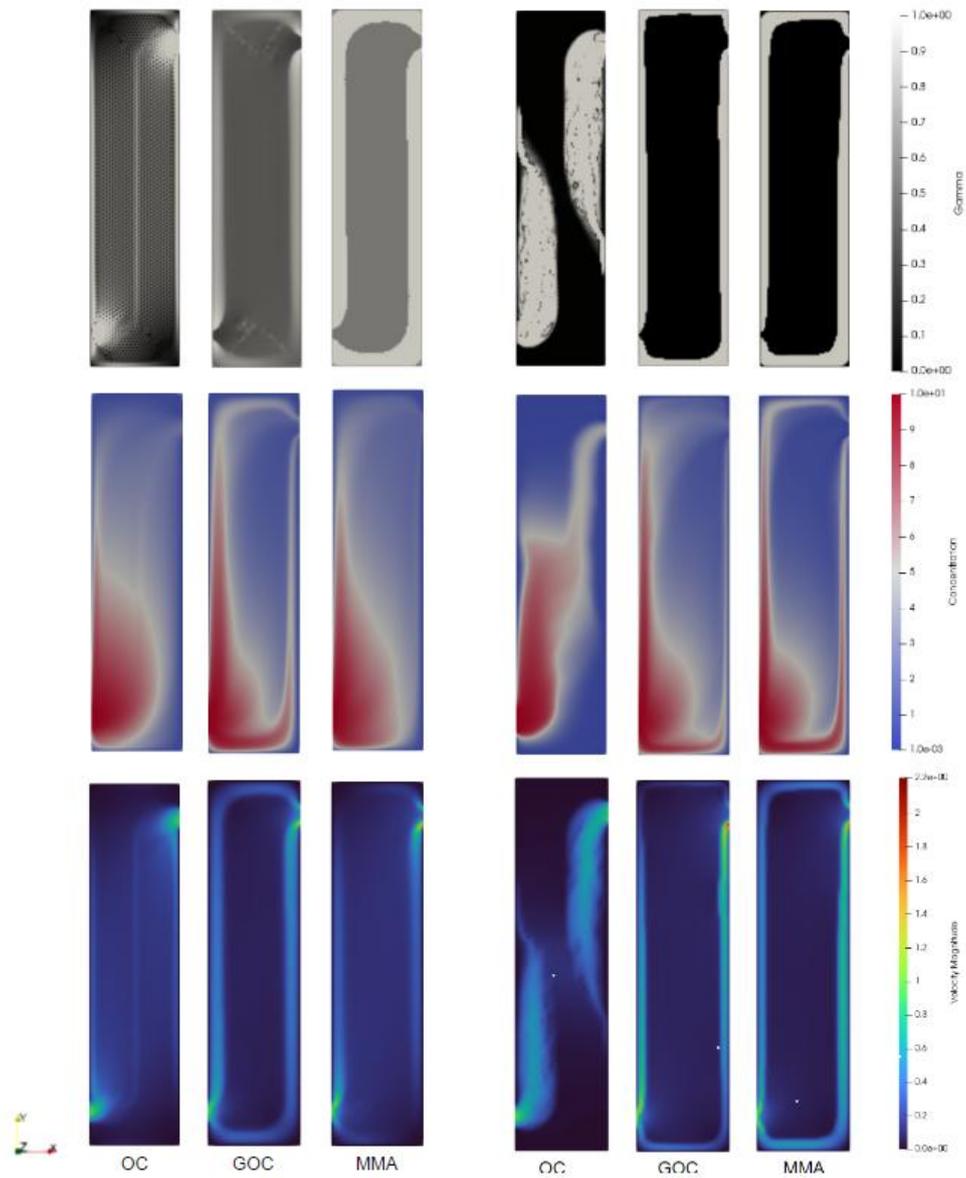


Figure 4.41: Snapshots of optimization procedure for the bend channel example in Figure (3.10). Left column, Iteration 2, right column, final iteration.

the use of a regularization technique (density filter), the optimization problem described at (3.5.3) does not ensure an optimal continuous solution of the material distribution. An example of the instabilities that may arise is given by considering the bend channel example (3.10) with  $\xi = 0$ , setting  $\alpha_{max} = 1e6$ . Snapshot of the optimization are shown in Figure 4.43.

The OC algorithm seems to find a stable solution even with this high value for  $\alpha_{max}$ , but it does not divide the channel, nor curve it to increase the residence time and pollute reduction. The final  $L_2$  concentration outlet norm of 0.40. On the other hand, both the GOC and MMA schemes converge to a check-board pattern, clearly dependent on the adopted mesh. Such configuration is extremely more efficient in the pollute distribution, ( $L_2$  norm of 0.16 for both). These solutions are anyway non physical and impossible to reproduce, thus future developments should then definitely introduce a density filter in the optimization process.

### 4.2.3 Optimization schemes: final considerations

Analyzing the results for the bend channel scenario, it is possible to deduce some features of the implemented Topology Optimization algorithm. Firstly, the tool is able to recognize the presence of an additional concentration-dependent term in the functional, and weight its effect on the optimization process. Secondly, the Generalized Optimality Criteria and the Method of Moving Asymptotes seem to have similar optimal configurations at the increment of the ratio between pollute decayment and minimization of the dissipation energy, while the Optimality Criteria converge quickly to local minimum, which seems to be almost constant with  $\xi$ .

With these considerations the Optimality Criteria algorithm poor to analyze

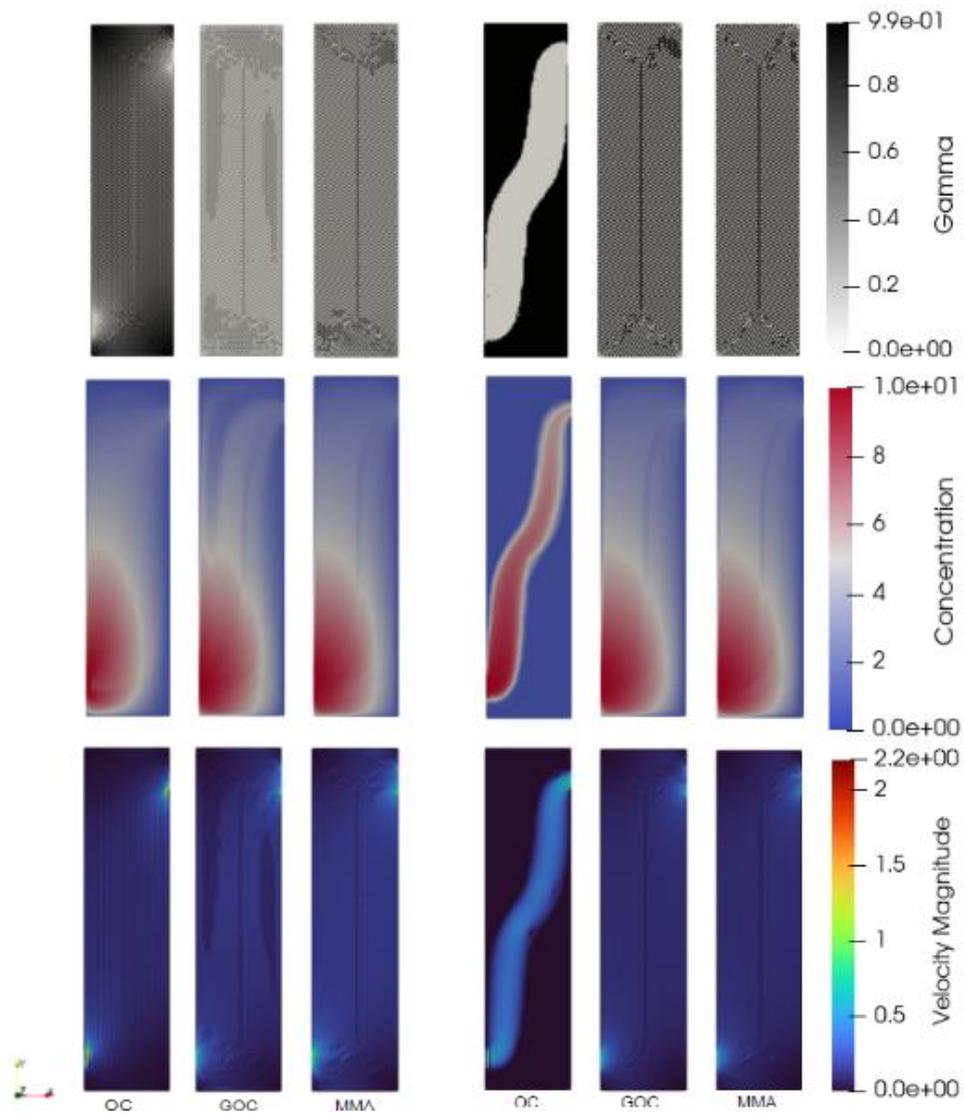


Figure 4.43: Snapshots of optimization procedure for the bend channel example in Figure (3.10) for  $\xi = 0$  and  $\alpha = 1e6$ . Left column, Iteration 2, right column, final iteration.

the transport scenarios, while the MMA shows inalterred convergence properties with  $\xi$ .



# Chapter 5

## Conclusions and future developments

A topology optimization algorithm is developed to solve the layout optimization problem for fluid flow and flow-driven scalar transport problems, based on a density method approach for the optimization, and a Finite Element method for the solving of the governing equations.

After an analysis of the current state of the art, a new derivation for the adjoint system of the Navier-Stokes equations is proposed, with the goal of adopting a simple approach against the current hard and poorly treated formulations in the literature. Further, we present a generalization to couple the advection-diffusion-reaction equations with the Navier-Stokes equations.

The complete code was implemented in the C++ language through an object oriented programming routine, applying the OpenMP directives for a shared memory parallelization of the heaviest computations.

With the goal of efficiency, the used technique for the solution of the linear systems, the GMRES algorithm, was accelerated by the use of an almost new preconditioner, based on the popular SIMPLE solver widely used in commercial softwares.

Lastly, the algorithm is validated by comparing well-known solutions found in the literature for fluid-flow problems, and with an *ad-hoc* example for the flow-driven transport problem. Among the three methods proposed, i.e. OC, GOC, and MMA, starting from a configuration completely filled by the fluid

domain, the Optimality Criteria is the most stable optimization method in purely fluid flow problems, but it was unable to reach the best configuration in the transport scenario. The Optimality Criteria is mostly applied to topology optimization in structural mechanics, and only works with a single constraint on the optimization, but it is able to converge in very few iterations, thus requiring extremely less computational time than the other methods. On the other hand, the GOC algorithm should extend the Optimality Criteria to more general settings, but provides very slow updates on the optimization domain, requiring much more optimization iterations when a good starting solution is not known.

Lastly, the Method of Moving Asymptotes is the most popular algorithm for fluid application, but the convergence to the best configuration is more dependent on the initial condition and parameters calibration. Nonetheless, comparing fluid-flow and transport scenarios, this method resulted the technique valid for most of applications, with good converging properties.

An important note is that checkboard patterns may arise at moderate or high Reynolds numbers,  $Re > 100$ , in particular for the flow-driven transport problem, invalidating the optimization process. The same problem has been found for the flow-driven scalar transport problem, making impossible the analysis of the problem with a true impermeable material. The algorithm would thus require the additional implementation of a density filter, to force a continuous porosity solution inside the domain.

A second important generalization of the implemented algorithm would then be the consideration of a material-dependent diffusion coefficient for the transport problem, to study more complex problem with a realistic non identical behaviour of species diffusion across fluid and solid regions.

Finally, another interesting improvement may come with the testing over different interpolation rule for the material porosity over the optimization param-

eter,  $\gamma$ , thus changing the "easiness" of movement from solid to fluid domain, and *viceversa*, to test the effect on the optimization process.



# Chapter 6

## Appendices

### 6.1 Appendix A: Boundary conditions

The aim of this project is to analyze the best shape of the solid material inside the domain to minimize (or maximize) the chosen functional, on different external conditions.

It is well known that for a boundary value problem, there are three possible types of boundary conditions, Dirichlet, Neumann and Robin, which briefly speaking, respectively, the value, the flux or a mixture of the two for the unknowns of the problem. For the Navier-Stokes problem, this corresponds to set the velocity value, its flux, or a combination of the two at the contour. The contour can be subdivided in parts with different boundary conditions, in values or type, with the only constraint to have just one condition for each node at the boundary, in order to avoid an overconstrained problem.

$$\mathbf{u}(\mathbf{r}) = \mathbf{g}_D(\mathbf{r}), \quad \mathbf{r} \in \Gamma_D, \quad (\text{Dirichlet BCs})$$

$$[p\mathbf{I} - \mu\nabla\mathbf{u}(\mathbf{r})] \cdot \mathbf{n} = \mathbf{q}_N(\mathbf{r}) \quad \mathbf{r} \in \Gamma_N \quad (\text{Neumann BCs})$$

$$\beta\mathbf{u} + [p\mathbf{I} - \mu\nabla\mathbf{u}(\mathbf{r})] \cdot \mathbf{n} = \mathbf{q}_C(\mathbf{r}) \quad \mathbf{r} \in \Sigma_C \quad (\text{Robin BCs})$$

For brevity, we will here report the features of just the Dirichlet and Neumann types, since the Robin condition is far less employed in simple problems like the ones presented in this dissertation. An important point to stress is that, since the pressure never appears in the governing equations outside from a derivative, it is necessary to impose at least one Neumann B.C., otherwise it would only be defined up to a constant and would generate oscillations in the solution. Another possibility to avoid oscillations (briefly discussed above), without this Neumann condition is to impose a further equation setting the mean pressure to a chosen value, like zero,

$$\int_{\partial\Omega} p d\mathbf{r} = 0. \quad (6.1.1)$$

For the same reason, it is compulsory to impose at least one Dirichlet B.C. for the velocity at least in the Stokes problem.

### **The Dirichlet B.C.**

The Dirichlet boundary conditions aim to fix the values of the unknowns at the chosen boundary  $\Gamma_D \subset \partial\Omega$ . In the discretized formulation, this corresponds to fix the nodal values of the velocity

$$\mathbf{u}_i = \mathbf{g}_i, \quad (6.1.2)$$

where  $\mathbf{g}_i$  is the value that has to be imposed at node  $i$  of the Dirichlet boundary (using above notations,  $\mathbf{g}_i = \mathbf{g}_D(\mathbf{x}_i)$ , with  $\mathbf{x}_i$  position of node  $i$ ). Other possibilities, like fixing just one of the velocity components are just trivial extensions of (6.1.2). In contrast, the pressure value is related to the flux of the velocity, hence the case of fixed pressure in the boundary will be treated

with the Neumann boundary conditions.

The Dirichlet conditions is usually called "essential", since to enforce them, equation  $i$  of the Galerkin formulation, with  $i$  index of a Dirichlet node, has to be completely substituted with something like (6.1.2). More specifically such constraints have to be imposed in the strong formulation, changing the trial space of the solution candidates.

From a computational point of view, one simply substitutes the FEM equations at the Dirichlet nodes with the prescribed values, usually with a "lifting function" or a "penalty method" approach ([19]).

### Neumann B.C.

The Neumann boundary conditions impose the velocity flux at the boundary. This condition is usually called "Natural" because it appears naturally in the FEM formulation. As shown in (2.4.1), to impose the Neumann conditions is sufficient to set the value of  $[\frac{p}{\rho}\mathbf{I} - \nu\nabla\mathbf{u}] \cdot \mathbf{n} = \mathbf{q}_N$  in  $\Gamma_N$ .

### General external conditions

In a general flow problem, there are at least three typical boundary conditions employed in the majority of the cases ([46])

1.  $\mathbf{u}|_{\Gamma_{in}} = \mathbf{u}^{in}$ , Inflow condition;
2.  $\mathbf{u}|_{\Gamma_{rigid}} = 0$ , Solid wall/"no slip" condition;
3.  $([p\mathbf{I} + \mu\nabla\mathbf{u}] \cdot \mathbf{n})|_{\Gamma_{out}} = (p\mathbf{n} + \mu\partial_n\mathbf{u})|_{\Gamma_{out}} = 0$ , Outflow condition,

where,  $\Gamma_{in}$ ,  $\Gamma_{solid}$  and  $\Gamma_{out}$  are non intersecting portions of  $\partial\Omega$ . Usually the inflow and solid wall condition is specified with a Dirichlet type BC prescribing the velocity of the inflow or setting all the velocity components to zero. The treatment of the outflow boundary condition is instead a little more tricky.

Numerical simulation of flow problems usually requires the truncation of a conceptionally unbounded flow region to a bounded computational domain. The conditions adopted, aim at numerically replacing the behaviour of the field left out, which are exactly the inflow and outflow condition.

The outlet condition stated above,  $(p\mathbf{n} + \mu\partial_n\mathbf{u})|_{\Gamma_{out}} = 0$  has proven to be well suited for laminar parallel flows, and since it is naturally set without prescribing any boundary conditions, it is called the "do nothing" approach or "free outflow" condition [47].

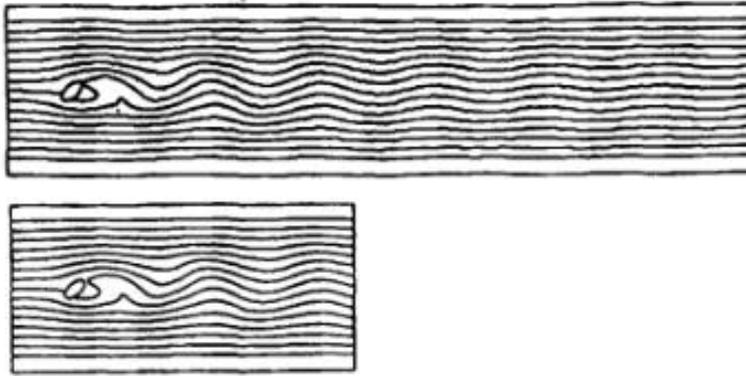


Figure 6.1: Effect of the "do nothing" outflow condition shown by streamline plots for flow around an inclined ellipse at  $Re = 500$  after 100 time steps. From [47]

This approach usually yields very satisfactory results (see Figure 6.1). However, it is important to analyze what lies underneath its formulation. With some trivial computations, calling  $S \subset \partial\Omega$  the portion of the boundary where one wants to apply such condition, we have

$$p\mathbf{n} + \mu\partial_n\mathbf{u} = 0 \implies \int_S p\mathbf{n}d\sigma = \mu \int_S \partial_n\mathbf{u}d\sigma = -\nu \int_S \partial_t\mathbf{u}d\sigma = 0, \quad (6.1.3)$$

where we have implicitly assumed the incompressibility of the fluid, i.e.,

$$\nabla \cdot \mathbf{u} = 0 \implies \partial_n\mathbf{u} + \partial_t\mathbf{u} = 0, \quad (6.1.4)$$

with  $t$  tangential direction. It is to note that  $\mathbf{u} = 0$  on  $S$  by applying the outflow boundary apart from the inflow condition, and, thus, near to the solid wall boundary, where the velocity is identically zero due to the "no slip" condition. Therefore, the mean pressure over  $S$  must be zero:

$$\int_S p d\sigma = 0. \quad (6.1.5)$$

This means that, the "do nothing" outflow is equivalent to apply a mean zero pressure on  $S$ .

Some problem arise when multiple outflow regions have to be imposed. The example in Figure (6.2) shows the streamlines for a Poiseuille flow with same inflow condition and same variational formulation, but domains with one leg of the pipe longer than the other in the second formulation. It is clear that, in the second case, the flux across the upper boundary increases.

To overcome this problem, a solution consist to apply different mean pressures to each outflow boundary, given that

$$\begin{aligned} p\mathbf{n} + \mu\partial_n\mathbf{u} = \mathbf{q}_N &\implies \int_S p\mathbf{n} + \mu\partial_n\mathbf{u}d\sigma = \int_S p\mathbf{n}d\sigma - \int_S \mu\partial_t\mathbf{u}d\sigma = \int_S p\mathbf{n}d\sigma \\ \bar{p} = \frac{1}{|S|} \int_S p\mathbf{n}d\sigma &\implies \int_S p\mathbf{n} + \mu\partial_n\mathbf{u}d\sigma = |S|\bar{p}. \end{aligned} \quad (6.1.6)$$

More insights on this formulation and some other outflow conditions approach are given in [46] or [47].

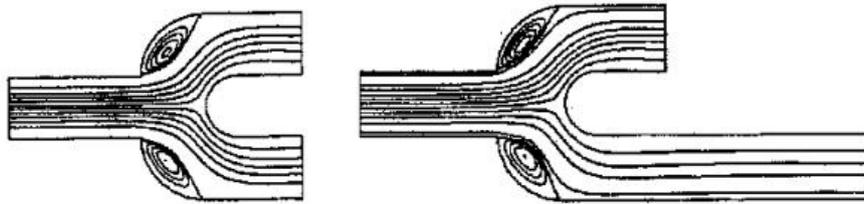


Figure 6.2: Effect of the "do nothing" outflow condition shown by streamline plots for flow through a bifurcating channel at  $Re = 20$ . From [46]

An analysis of the pressure boundary condition for fluid flow problems [48] showed that, when a Neumann condition is imposed to a non-straight boundary that is connected only to wall boundaries, we have

$$\int_{\Gamma_N} \left( \frac{1}{\rho} p \mathbf{I} - \nu \partial_n \mathbf{u} \right) \cdot \mathbf{n} d\sigma \simeq |\Gamma_N| \bar{p} + o\left(\frac{C}{r^d}\right), \quad (6.1.7)$$

with  $C$  constant depending on the mean outgoing flux and  $r$  the mean curvature radius of the boundary.

This condition is hence exact for straight surfaces  $r \rightarrow \infty$ . In that case mean pressure condition can be applied trivially as Neumann BCs.

### The Symmetry BC

Finally, symmetry BCs aim at prescribing standard symmetry conditions, i.e.,

$$\begin{aligned} \mathbf{u} \cdot \mathbf{n} &= 0, \\ \nabla \mathbf{u}_{\parallel} &= 0, \end{aligned}$$

which defines three conditions for the three dof of each node at the boundary (normal component and gradient of parallel component in normal direction set to zero). The first one can be imposed simply via lifting functions as the Dirichlet BCs, while for the other two conditions is sufficient to notice that

$$\nabla \mathbf{u}_{\parallel} = \nabla(\mathbf{u} - \mathbf{u}_n) = \nabla \mathbf{u} - \nabla[(\mathbf{u} \cdot \mathbf{n})\mathbf{n}] = 0.$$

Accordingly, the corresponding term in variational formulation (2.4.4) reduces to

$$\int_{\Sigma_S} \left( \frac{1}{\rho} p \mathbf{I} - \nu \nabla \mathbf{u} \right) \mathbf{n} \cdot \mathbf{v} d\sigma = \int_{\Sigma_S} \left[ \left( \frac{1}{\rho} p \mathbf{I} - \nu (\mathbf{u} \cdot \mathbf{n}) \mathbf{n} \right) \mathbf{n} \cdot \mathbf{v} d\sigma \right].$$

This term has thus been discretized and added to the above formulation in

case of Symmetry BCs. Noting that, in case of symmetry face aligned with one of the Cartesian axis, symmetry conditions are exactly coincident to the imposition of velocity normal component to zero and free outlet condition (Neumann BC with  $g = 0$  in  $\Sigma_S$ ) for the other components.

## 6.2 Appendix B: the elements choice

The well-posedness for the FEM linear system of the Navier-Stokes problem (2.4.1) is guaranteed if the *discrete inf-sup condition* (also known as Babuska-Brezzi condition [49]) holds. In the practice, this condition prescribes that, for  $p \neq 0$ , there must be a  $v$  such that

$$p^T Bv \geq \beta \|v\| \|p\|, \quad (6.2.1)$$

with  $\beta > 0$ . Condition (6.2.1) guarantees that the system matrix

$$K = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad (6.2.2)$$

is invertible. This immediately fails, if a nonzero pressure  $p$  for which  $Bp = 0$  exists. In the discrete cases this is equivalent to the maximum rank to  $B$  (full column rank). In particular, by (6.2.1), every possible nonzero pressure  $p$  must not be orthogonal to every  $Bv$ . Thus, the space of all  $Bv$ 's must have at least the dimension of the pressure space [50]. In our case,  $B$  represents the divergence operator, and this is the reason why often the velocity elements are defined by polynomials with one degree higher than those used for the pressure elements. Unfortunately, dimension of spaces are not sufficient to prove the discrete inf-sup condition (6.2.1), and it is necessary to check each

particular couple of spaces  $(\mathcal{V}_h, \mathcal{Q}_h)$ . The general idea is then that the two spaces cannot be chosen independently and that, loosely speaking, we need to have enough degrees of freedom in  $\mathcal{V}_h$  with respect to  $\mathcal{Q}_h$ .

Defining  $P_0, P_1, P_2$  as constant, linear and quadratic polynomials on triangles, respectively, and  $Q_0, Q_1, Q_2$  for constant, bilinear, and biquadratic elements on quadrilaterals, we have the following results [20]:

1. Velocities in  $P_1$ , pressures in  $P_0$ : failure
2. Velocities in  $P_1$ , pressures in  $P_1$ : failure
3. Velocities in  $P_2$ , pressures in  $P_1$ : **well posed**
4. Velocities in  $Q_1$ , pressures in  $Q_1$ : failure
5. Velocities in  $Q_2$ , pressures in  $Q_1$ : **well posed**.

Therefore, the simplest choice for a triangular discretization would be the  $P_2/P_1$  approach. We have, however, other possibilities to enrich the velocity space preserving the simplicity of the use of linear shape functions (see Figure 6.3). Above all, the  $P_1$  - iso -  $P_2/P_1$  simply enriches the velocity space by uniform refinement the triangulation (connects the midpoints of each triangle) [21].

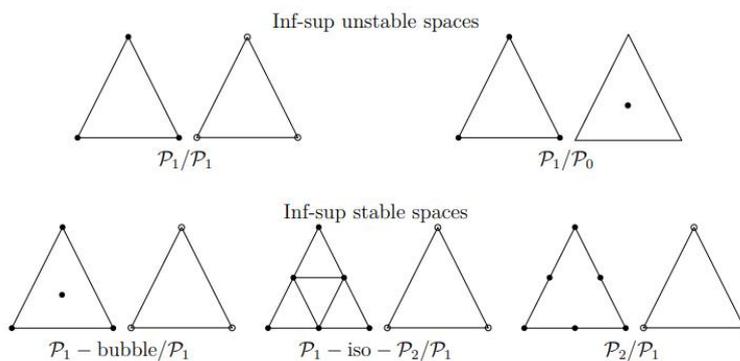


Figure 6.3: Some inf-sup stable and unstable spaces.

## 6.3 Appendix C: Solver efficiency

The efficiency of the implemented code has been tested against PARDISO commercial software (academic library [38]). The PARDISO library provides functions for both direct or iterative solvers. Clearly these methods are created to be fast with every pattern of the entering system matrix, and not specifically for the block matrix of the Naviers Stokes problem.

The behavior of the different implemented solvers has been tested on several problems and three different numbers of degrees of freedom, with an exit rule of residual norm tolerance lower than  $1e-5$  for all the iterative schemes.

### 2D COMPARISON: PARDISO VS SIMPLE vs SCHUR vs JACOBI VS GMRES

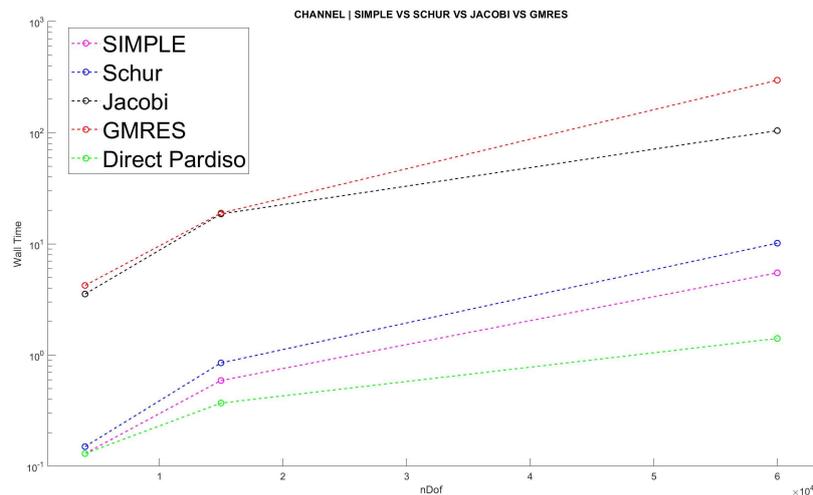


Figure 6.4: Mean Wall time Vs dof for the various preconditioners, and PARDISO direct solver.

### 3D COMPARISON: SIMPLE vs PARDISO vs DIRECT PARDISO

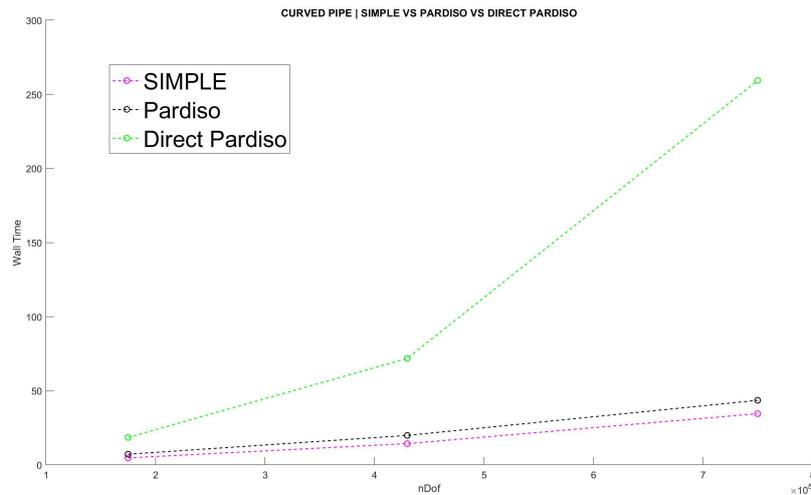


Figure 6.5: Mean Wall time Vs dof for the various preconditioners, and PARDISO direct solver.

About the 3D comparison with PARDISO some important remarks can be done. First of all, the direct solver of PARDISO is faster than any other scheme, when applied to 2D problems with less than  $1e5$  degrees of freedom. However, its performance worsens as we test the algorithm to 3D problems with a large number of dof. Therefore, the SIMPLE preconditioner, which is the best among the implemented preconditioners, has been tested against the iterative version of PARDISO, which adopts a SQMR algorithm (Simplified Quasi-Minimal-Residual). From the time analysis of the tested problems, the SIMPLE-preconditioned GMRES, performs better than PARDISO (max dof test =  $7.5e5$ ). An explanation can be that the PARDISO solver is optimized for the solution of any sparse matrix linear system. *Vice versa*, the SIMPLE procedure is thought specifically to be applied to linear system deriving from the discretization of the Stokes or Navier-Stokes problems.

### SMP scaling

Some results about the scaling of the SIMPLE-preconditioning scheme by varying dof with various number of threads are reported in Figures (6.6) and

(6.7).

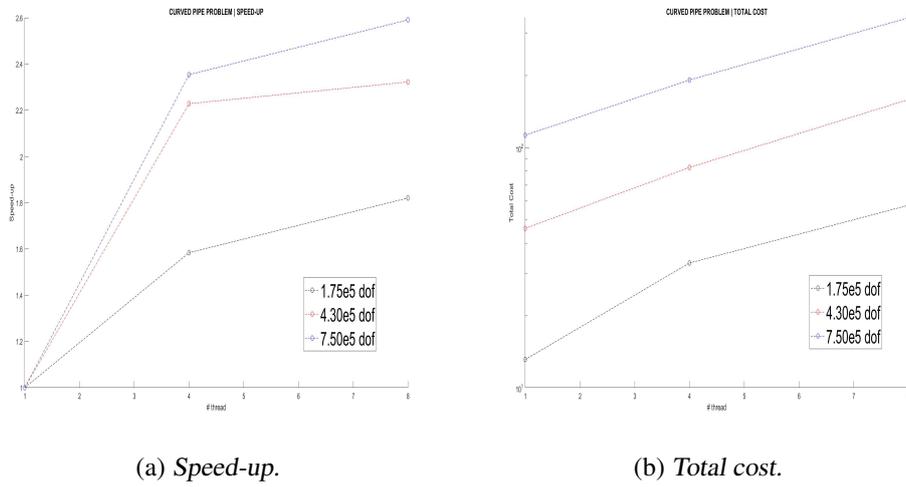


Figure 6.6: Result at thread varying for the SMP implementation of the SIMPLE-based GMRES.

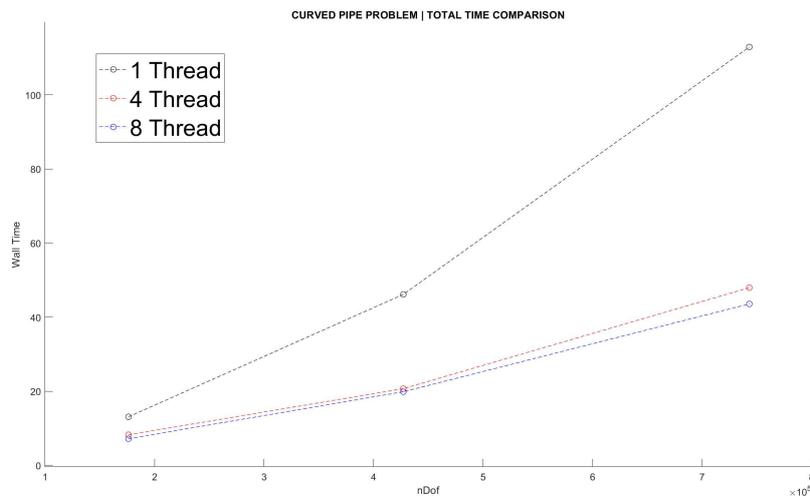


Figure 6.7: Time comparison over 3 different degrees of freedom for different number of threads.

## 6.4 Appendix D: Validation

### 6.4.1 Advection Diffusion Reaction

Let assume a constant diffusion coefficient  $D = 1.0\text{m}^2/s$  and  $k = 0.51/s$ . The developed FEM model had been tested against hand-made theoretical solutions. Given formulation (2.4.7), it is possible to choose any possible velocity field  $\mathbf{u}$  and solution  $c(\mathbf{x}, t)$ . Further, we can compute from the governing equation an appropriate set of boundary conditions and source/sink term  $r$ , which could produce such solution  $c(\mathbf{x}, t)$ . Explicit calculus is showed only for the first example.

Validation has been developed for several different problems and cases. Here, results of two critical scenarios are showed, namely, unsteady two-dimensional and steady three-dimensional. Each problem has been tested with the relative errors adopting three different mesh (generated by COMSOL [42]), namely EXTRA COARSE, NORMAL, and EXTRA FINE. The elemental length (element maximum edge) are, respectively, maximum 0.4, 0.134, and 0.04 times the characteristic length of the problem in 2 dimensional problems, and 0.38, 0.2, and 0.07 times the characteristic length of the problem in 3 dimensional problems.

Considering  $\epsilon = c - c_h$ , with  $c_h$  the approximate solution, absolute errors are evaluated using an  $L^2(\Omega)$  norm for steady scenarios and an  $L^2([0, T], L^2(\Omega))$  norm for the unsteady ones, i.e.,

$$\|\epsilon\|_{L^2(\Omega)} = \left( \int_{\Omega} (c - c_h)^2 d\mathbf{x} \right)^{\frac{1}{2}}$$

$$\|\epsilon\|_{L^2([0, T], L^2(\Omega))} = \left( \int_0^T \int_{\Omega} (c - c_h)^2 d\mathbf{x} \right)^{\frac{1}{2}},$$

while the relative errors are simply

$$\begin{aligned} \|\epsilon\|_{L^2(\Omega)}^R &= \frac{\|(c - c_h)\|_{L^2(\Omega)}}{\|c\|_{L^2(\Omega)}} \\ \|\epsilon\|_{L^2([0,T],L^2(\Omega))}^R &= \frac{\|(c - c_h)\|_{L^2([0,T],L^2(\Omega))}}{\|c\|_{L^2([0,T],L^2(\Omega))}}. \end{aligned}$$

### Ex:1 Unsteady 2D

Let consider solution

$$c(\mathbf{x}, t) = \frac{(x^2 + y^2)}{2}t,$$

subject to the velocity field  $\mathbf{u} = [y, -x]$ , on  $\Omega = \{(x, y) \in \mathbf{R}^2, x \in [0, 2], y \in [0, 1]\}$ . By simple computations

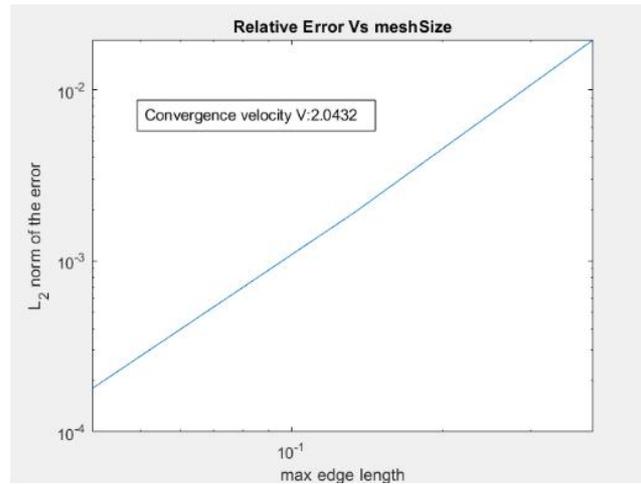
$$\partial_t c = \frac{x^2 + y^2}{2},$$

$$\mathbf{u} \cdot \nabla c = yxt - xyt = 0$$

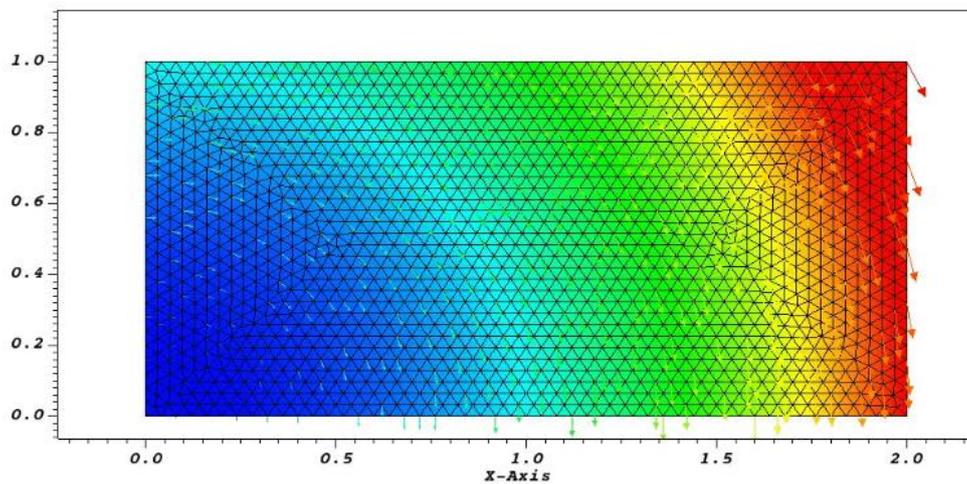
$$D\Delta c = Dt + Dt = 2Dt$$

$$r = \partial_t c + \mathbf{u} \cdot \nabla c - D\Delta c + kc \implies r = \frac{x^2 + y^2}{2}(1 + k) - 2Dt.$$

Results in terms of error convergence and graphical visualization are shown in Figure(6.8).



(a)  $L_2$  relative error (in space and time) convergence result against mesh size (MATLAB).



(b) VISIT representation of solution at time 2.4s. Color table from red  $c = 5$  to blue  $c = 0$ .

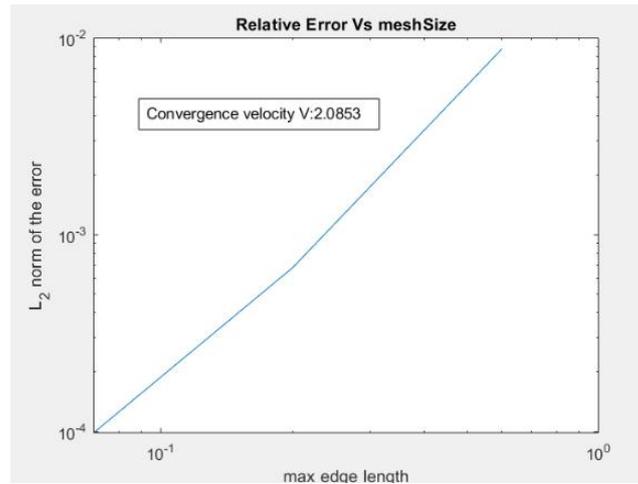
Figure 6.8: Validation with COMSOL solver for the unsteady 2D example.

### Ex:2 Steady 3D

Let consider now solution  $c(\mathbf{x}, t) = e^{-(x+y)} \sin(z)$ , subject to the velocity field  $\mathbf{u} = [1, -1, 1]$  on a parallelepiped of dimension  $1 \times 1 \times 2$  meters. With the same operations of previous case, the source term results in

$$r(\mathbf{x}) = e^{-(x+y)} [\cos(z) + (k - D)\sin(z)].$$

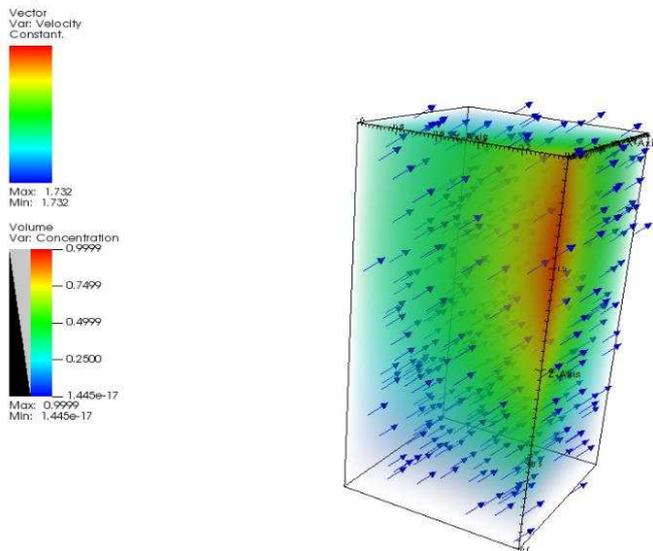
Error convergence graph and graphical results are shown in Figure(6.9).



(a) VISIT representation of solution at time 2.4s. Color table from red  $c = 5$  to blue  $c = 0$ .

DB: 101.vtu  
Cycle: 101

Time: 1e+16



(b)  $L_2$  relative error (in space and time) convergence result against mesh size (MATLAB).

Figure 6.9: Validation with COMSOL solver for the steady 3D example.

## 6.4.2 Navier-Stokes

Navier-Stokes solver has been validated both against theoretical solutions and COMSOL numerical software.

Validation through theoretical solutions has been developed as for the advection-diffusion-reaction problem, by mesh refinements and analysis of relative error

scaling.

Since the solution of a commercial software suffers from approximation, it is not possible to operate in the same way for scenarios without theoretical solution. For this reason, the validation through COMSOL numerical solver emerges via an empirical approach. Both the newly implemented solver and the commercial one should provide a vanishing error as the discretization is improved, and therefore we expect the same for the difference between the two approximate solutions.

In these cases, the test thus simply check that

- i) solutions difference (in norm) stay bounded and decreases as the mesh is refined, and
- ii) graphical results present the same characteristics (in terms of streamlines, pressure distribution, etc.).

The code has been validated via comparison against Poiseuille and Womersley theoretical solutions and several COMSOL analysis.

### Poiseuille flow

Consider a Poiseuille flow within a tube of radius  $R = 0.5\text{m}$  and length  $L = 4$  meters. Imposing a pressure difference  $\Delta P = 4 \text{ Pa}$ , the theoretical solution takes the form [51]

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ \frac{\Delta P}{L} 4\mu(R^2 - r^2) \end{bmatrix}, \quad (6.4.1)$$

with  $r = (x^2 + y^2)^{\frac{1}{2}}$ . The distance from the cylinder axis, with BCs

- $p = 2 \text{ Pa}$ ,  $z = 0 \text{ m}$ , bottom face;
- free outlet  $p = 0 \text{ Pa}$  (Top face);

- Wall no-slip condition on lateral surface;

Results are obtained for  $\rho = 0.5 \text{ Kg/m}^3$  and  $\mu = 0.05 \text{ Pa}\cdot\text{s}$  as showed in Figure (6.11).

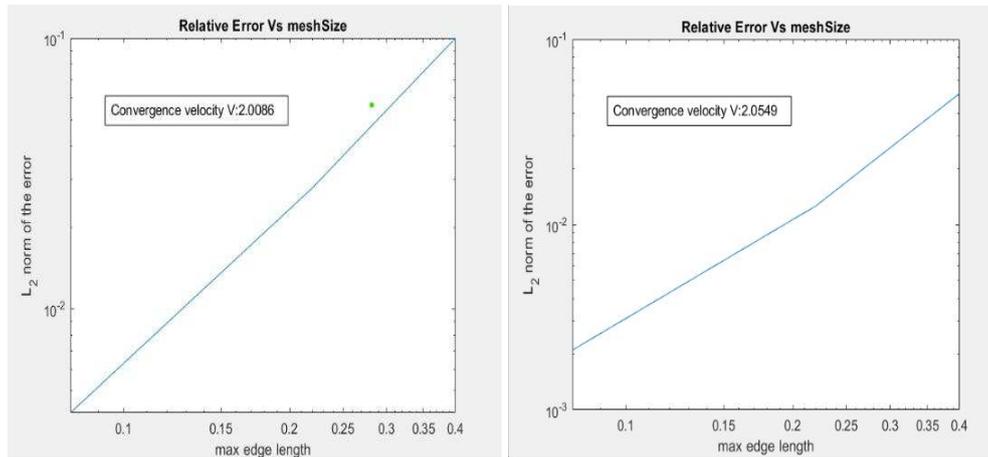


Figure 6.10: *L<sub>2</sub> error for pressure (left panel) and velocity (right panel).*

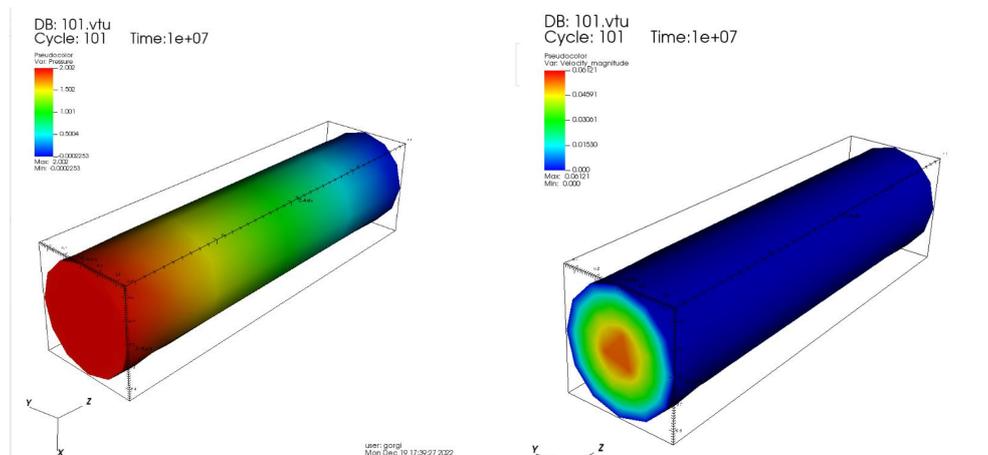


Figure 6.11: *Pressure and Velocity results.*

### Womersley flow

For a cylinder subject to a sinusoidal oscillating pressure  $\Delta P(t) = \Delta P_0(1 + \gamma_p \sin(\omega t))$ , the theoretical Womersley solution is [52]

$$\mathbf{u}(\mathbf{x}) = \begin{bmatrix} 0 \\ 0 \\ w(r, t) \end{bmatrix}, \quad (6.4.2)$$

with

$$w(r, t) = \frac{\Delta P_0}{4\mu L}(R^2 - r^2) + \text{Re} [\hat{w}(r)e^{-i\omega t}]$$

$$\hat{w}(r) = \frac{\Delta P_0 \gamma_p}{\rho L \omega} \left[ \frac{J_0(r\sqrt{i\omega/\nu})}{J_0(R\sqrt{i\omega/\nu})} - 1 \right].$$

Here,  $\gamma_p$  and  $\omega$  are the amplitude and the frequency of the pressure difference oscillations, respectively.  $J_0$  is a first-order Bessel function [53]. The solver proved good convergence properties even in this scenario, as showed in Figures (6.12) and (6.13).

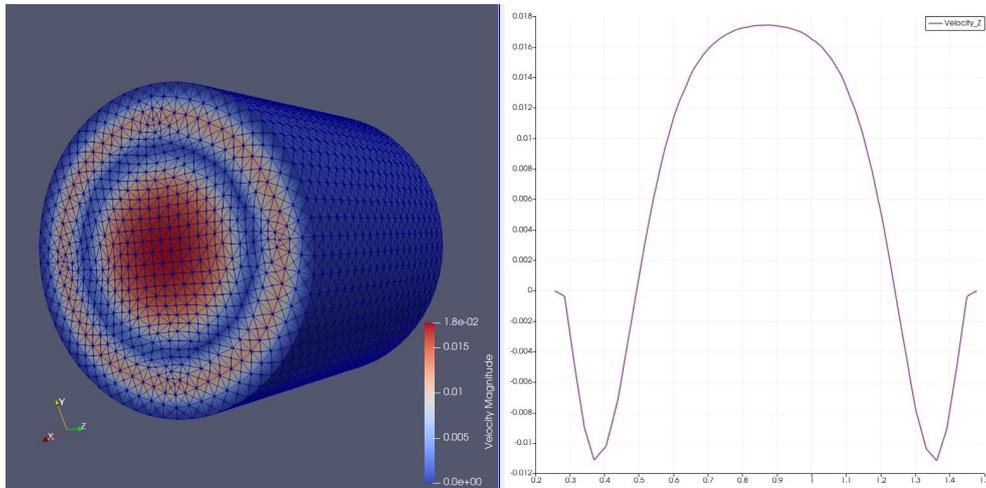


Figure 6.12: PARAVIEW representation of approximate Womersley solution at  $Wo = 12.5$ ,  $time = 1.1s$ .

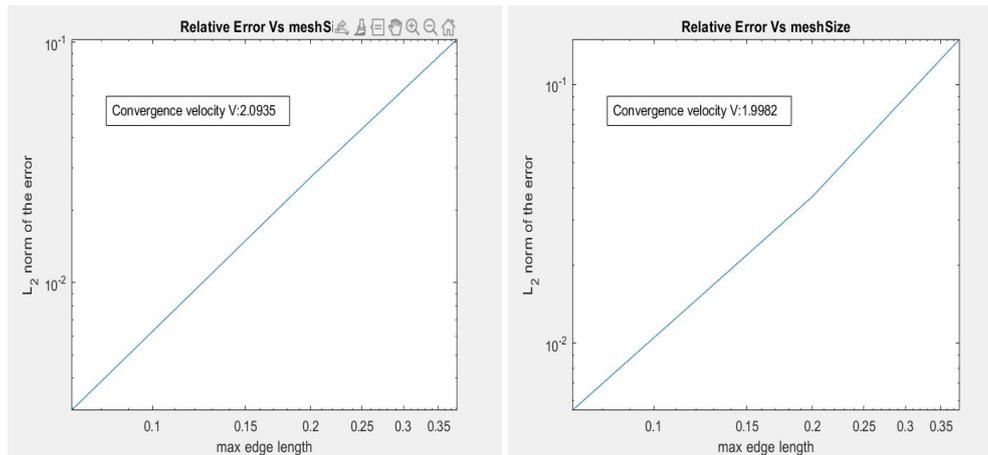


Figure 6.13:  $L_2$  error for pressure (left panel) and velocity (right panel).

### The 2D double pipe flow

The geometry for the first non theoretically solved case studied is described in Figure 6.14. The problem presents a simple  $1 \times 1$  (meters) box, with two inlet channels (on the left) and two outlet channels (on the right).

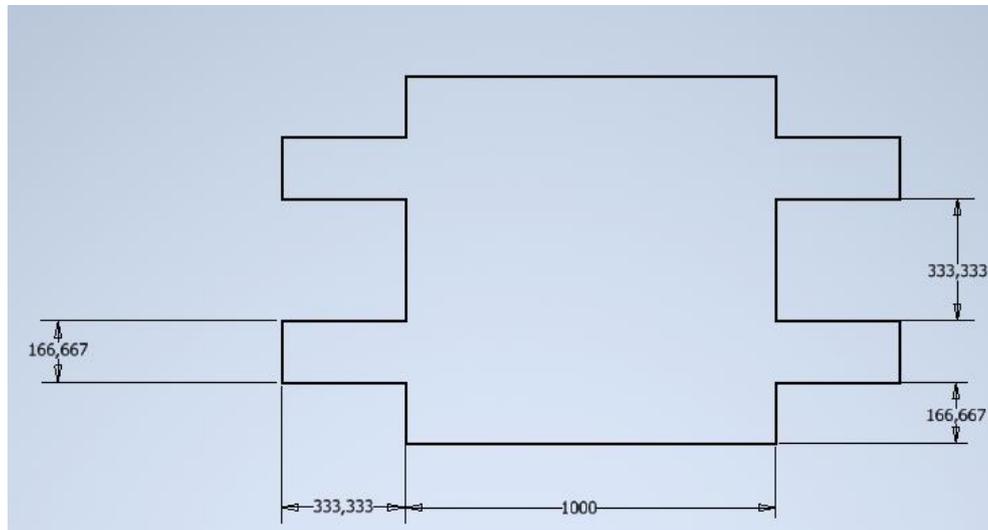


Figure 6.14: "Double Pipe" geometry in mm by INVENTOR [54]

The imposed velocity as inlet is a simple parabolic profile normalized to unity for both the entering channels. Outlets are then treated with a "free outlet" condition (6.1). All other boundaries are considered as no slip wall bound-

aries ( $\mathbf{u} = 0$ ). For the time dependent analysis a zero-initial condition was set in the whole domain, for a total time study of 8.0s.

This configuration has been tested with three different sets of physical parameters, to check behaviours at different Reynolds numbers, and three different mesh sizes for the convergence analysis. Comparison between our and COMSOL solution are presented in Figures (6.15), (6.16), and (6.17).

| MESH SIZE<br>[m] | REYNOLDS NUMBER                   |              |  |              |  |              |
|------------------|-----------------------------------|--------------|--|--------------|--|--------------|
|                  | Re = 1<br>( $\rho = 1, \mu = 1$ ) |              | Re = 100<br>( $\rho = 10, \mu = 0.1$ ) |              | Re = 1000<br>( $\rho = 10, \mu = 0.01$ ) |              |
|                  | $\epsilon_P$                      | $\epsilon_V$ | $\epsilon_P$                           | $\epsilon_V$ | $\epsilon_P$                             | $\epsilon_V$ |
| 0.130            | 15.50%                            | 25.30%       | 13.00%                                 | 25.00%       | 17.90%                                   | 36.50%       |
| 0.045            | 2.30%                             | 5.10%        | 2.41%                                  | 5.11%        | 9.53%                                    | 11.83%       |
| 0.013            | 2.00%                             | 4.40%        | 1.12%                                  | 1.15%        | 1.60%                                    | 4.80%        |

|                                |                            |
|--------------------------------|----------------------------|
| $\rho$<br>[kg/m <sup>3</sup> ] | density                    |
| $\mu$<br>[N/M <sup>2</sup> *s] | dynamic viscosity          |
| $\epsilon_P$<br>[1]            | Pressure L2 relative error |
| $\epsilon_V$<br>[1]            | Velocity L2 relative error |

Figure 6.15: Relative difference between COMSOL and developed FEM solutions in  $\|\cdot\|_{L^2([0,T],L^2(\Omega))}$  norm at different Re and meshes.

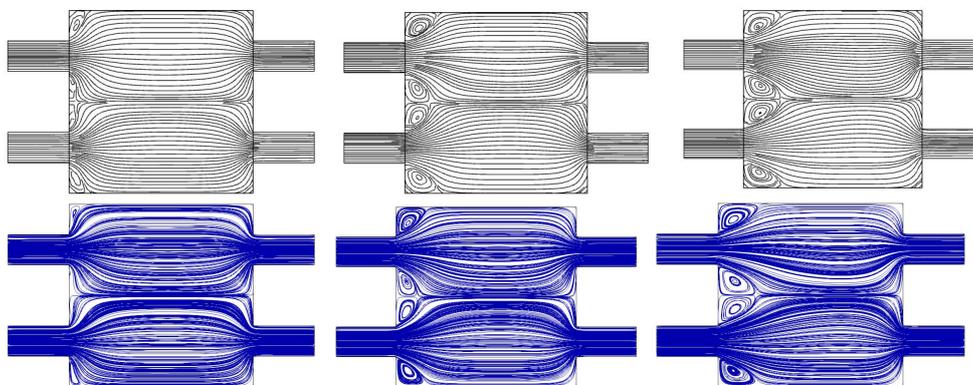


Figure 6.16: Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 100. Considered instants of time  $t=0.2s, t=0.5s, t= 8.0s$  from left to right.

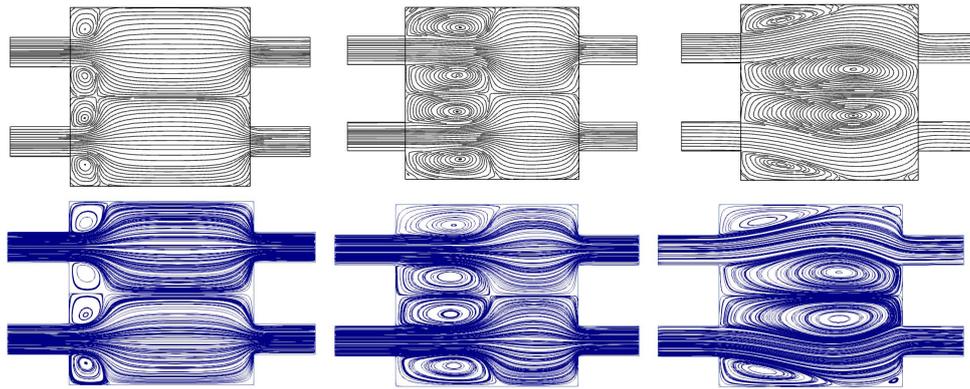


Figure 6.17: Streamline comparison between COMSOL (above) and implemented FEM solution (below) at  $Re = 1000$ . Considered instants of time  $t=0.5s$ ,  $t=1.5s$ ,  $t=8.0s$  from left to right.

### The 3D flow

For 3D comparison has been considered a cavity with one inlet and two outlets (see Figure 6.18).

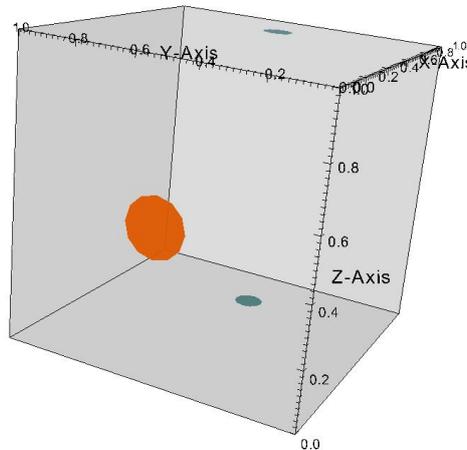


Figure 6.18: Cavity geometry. In orange inlet boundary (circle of radius  $0.1m$  centered in  $(0, 0.5, 0.5)$ ) and in blue outlet boundaries (circles of radius  $0.05m$ , centered in  $(0.75, 0.5, 0)$  and  $(0.75, 0.5, 1.0)$ ).

The imposed velocity at the inlet is parabolic with the maximum velocity equal to  $1\text{ m/s}$ , while the outlets are fixed by a "free outlet" condition. All other boundaries are considered as no slip wall boundaries. The time dependent analysis starts from a zero initial condition and evolves up to a final time

$$T = 8.0s.$$

As for the previous configuration, this settings have been tested with three different combinations of physical parameters and three different mesh sizes. Comparison between our and COMSOL solution are presented in Figures (6.19), (6.20), and (6.21).

| MESH SIZE<br>[m] | REYNOLDS NUMBER                   |              |  |              |  |              |
|------------------|-----------------------------------|--------------|--|--------------|--|--------------|
|                  | Re = 1<br>( $\rho = 1, \mu = 1$ ) |              | Re = 100<br>( $\rho = 10, \mu = 0.1$ ) |              | Re = 1000<br>( $\rho = 10, \mu = 0.01$ ) |              |
|                  | $\epsilon_P$                      | $\epsilon_V$ | $\epsilon_P$                           | $\epsilon_V$ | $\epsilon_P$                             | $\epsilon_V$ |
| 0.130            | 3.00%                             | 20.99%       | 0.50%                                  | 24.70%       | 0.60%                                    | 72.67%       |
| 0.067            | 0.10%                             | 13.90%       | 0.50%                                  | 16.40%       | 0.16%                                    | 32.40%       |
| 0.037            | 0.10%                             | 8.50%        | 0.50%                                  | 10.00%       | 0.15%                                    | 12.10%       |

|                                |                               |
|--------------------------------|-------------------------------|
| $\rho$<br>[kg/m <sup>3</sup> ] | density                       |
| $\mu$<br>[N/M <sup>2</sup> *s] | dynamic viscosity             |
| $\epsilon_P$<br>[1]            | Pressure L2<br>relative error |
| $\epsilon_V$<br>[1]            | Velocity L2<br>relative error |

Figure 6.19: Relative difference between COMSOL and developed FEM solutions in  $\|\cdot\|_{L^2([0,T],L^2(\Omega))}$  norm at different Re and meshes.

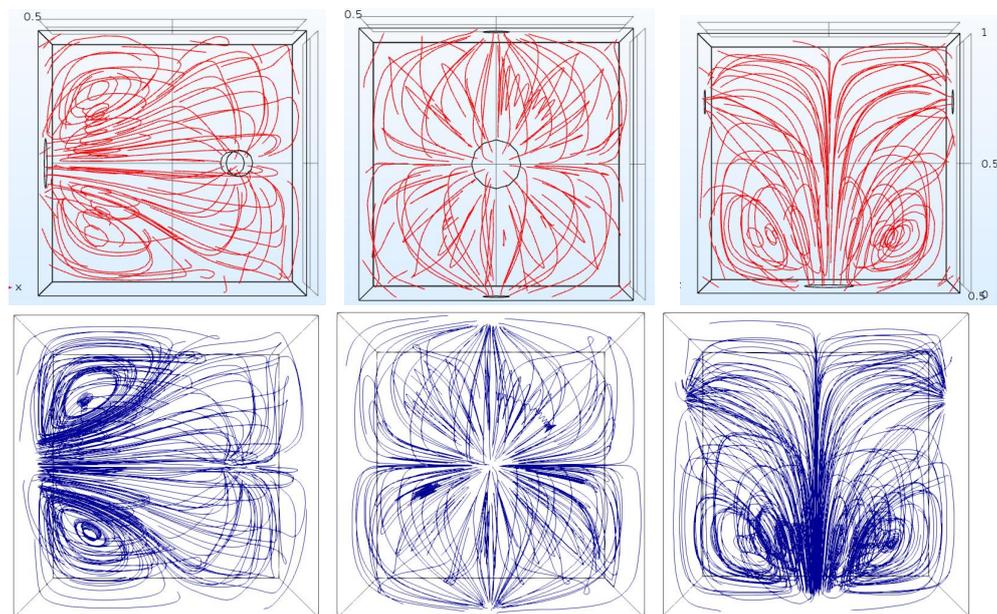


Figure 6.20: Streamline comparison between COMSOL (above) and implemented FEM solution (below) at Re = 100 for t = 8s. Plane xy (left panel), yz (middle panel), and zx (right panel).

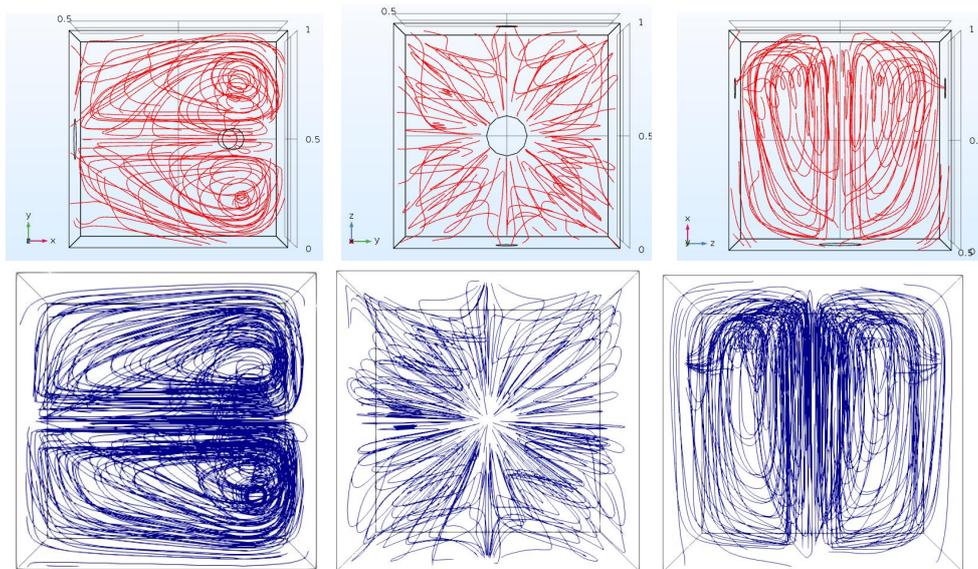


Figure 6.21: Streamline comparison between COMSOL (above) and implemented FEM solution (below) at  $Re = 1000$  for  $t = 8s$ . Plane  $xy$  (left panel),  $yz$  (middle panel), and  $zx$  (right panel).



# Bibliography

- [1] P.M. Du DZ. Pardalos and W. Wu. “History of Optimization”. In: *Encyclopedia of Optimization 2* (2008), pp. 1538–1542. DOI: [https://doi.org/10.1007/978-0-387-74759-0\\_268](https://doi.org/10.1007/978-0-387-74759-0_268).
- [2] Christodoulos A Floudas and Panos M Pardalos. *Encyclopedia of optimization*. Springer Science & Business Media, 2008.
- [3] UT Zimmermann and T Lindner. *Train schedule optimization in public rail transport*. Springer, 2003.
- [4] Stephen J. Wright. *optimization Encyclopedia Britannica*. <https://www.britannica.com/science/optimization>. Accessed 25 March 2023.
- [5] Lucas Stelle Chemim, Christine Sotsek Nicolle, and Mariana Kleina. “Layout optimization methods and tools: A systematic literature review”. In: *Gepros: Gestão da Produção, Operações e Sistemas* 16.4 (2021), p. 59.
- [6] Yu-Deok Seo, Hyun-Jung Kim, and Sung-Kie Youn. “Shape optimization and its extension to topological design based on isogeometric analysis”. In: *International Journal of Solids and Structures* 47.11 (2010), pp. 1618–1640. ISSN: 0020-7683. DOI: <https://doi.org/10.1016/j.ijsolstr.2010.03.004>. URL: <https://www.sciencedirect.com/science/article/pii/S002076831000079X>.
- [7] X. Chen. “Topology optimization of microfluidics—a review”. In: *Microchemical Journal* 127 (2016), pp. 52–61.

- [8] C.S Alexandersen J.; Andreassen. “A Review of Topology Optimisation for Fluid-Based Problems”. In: *Fluids* 5.29 (2020). DOI: <https://doi.org/10.3390/fluids5010029>.
- [9] T. Weinberg D. Kim N. H. Dong and Dalidd. “Generalized optimality criteria method for topology optimization”. In: *Applied Sciences* 11.7 (2021).
- [10] Krister Svanberg. “The Method of Moving Asymptotes - A new Method for Structural Optimization”. In: *International journal for numerical methods in engineering* 24.2 (1987), pp. 359–373.
- [11] Martin Philip Bendsøe and Ole Sigmund. *Topology optimization: theory, methods, and applications*. Springer Science & Business Media, 2003.
- [12] Kazuo Yonekura and Yoshihiro Kanno. “A heuristic method using Hessian matrix for fast flow topology optimization”. In: *Journal of Optimization Theory and Applications* 180 (2019), pp. 671–681.
- [13] A. Stück. *Adjoint Navier-Stokes methods for hydrodynamic shape optimisation*. Technische Universität Hamburg, 2012.
- [14] & Petersson J. Borrvall T. *Topology optimization of fluids in Stokes flow*. Vol. 41. 1. International Journal for Numerical Methods in Fluids 2003; 41:77–107, 2003, pp. 77–107.
- [15] Laurits Højgaard Olesen, Fridolin Okkels, and Henrik Bruus. “A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow”. In: *International Journal for Numerical Methods in Engineering* 65.7 (2006), pp. 975–1001.
- [16] Z. Liu Y. Deng Y. Wu. *Topology Optimization Theory for Laminar Flow*. Springer Singapore, 2018. DOI: <https://doi.org/10.1007/978-981-10-4687-2>.

- [17] David Makhija and Kurt Maute. “Level set topology optimization of scalar transport problems”. In: *Structural and Multidisciplinary Optimization* 51 (2015), pp. 267–285.
- [18] Yuchen Guo et al. “Topology optimization of passive micromixers based on Lagrangian mapping method”. In: *Micromachines* 9.3 (2018), p. 137.
- [19] Mario Putti. *Notes on Numerical Methods for Differential Equations*. Lecture Notes of course "Numerical Methods for Continuous System". Padova, 2021.
- [20] Gilbert Strang. *18.086 Mathematical methods for engineers II, Spring 2005*. 2005.
- [21] Alfio Quarteroni. *Numerical Models for Differential Problems*. Springer Cham, 2017. DOI: <https://doi.org/10.1007/978-3-319-49316-9>.
- [22] Massimiliano Ferronato. *Iterative Methods for Sparse Linear Systems*. Lecture Notes of course "Numerical Methods for High Performance Computing". Padova.
- [23] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [24] Youcef Saad and Martin H Schultz. “GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems”. In: *SIAM Journal on scientific and statistical computing* 7.3 (1986), pp. 856–869.
- [25] Youcef Saad. “A Flexible Inner-Outer Preconditioned GMRES Algorithm”. In: ().
- [26] M Bellalij, K Jbilou, and H Sadok. “New convergence results on the global GMRES method for diagonalizable matrices”. In: *Journal of computational and applied mathematics* 219.2 (2008), pp. 350–358.

- [27] Malcolm F. Murphy, Gene H. Golub, & Andrew J. Wathen. “A note on preconditioning for indefinite linear systems [https://ora.ox.ac.uk/objects/uuid:36c706dd-45d0-4882-9ff3-cae603b964be/download\\_file?file\\_format=pdf&safe\\_filename=NA-99-07.pdf&type\\_of\\_work=Report](https://ora.ox.ac.uk/objects/uuid:36c706dd-45d0-4882-9ff3-cae603b964be/download_file?file_format=pdf&safe_filename=NA-99-07.pdf&type_of_work=Report)”. In: (1999).
- [28] Malcolm F. Murphy, Gene H. Golub, & Andrew J. Wathen. “Krylov subspaces p.164-165 <https://people.inf.ethz.ch/arbenz/ewp/Lnotes/chapter9.pdf>”. In: ().
- [29] Vuik Segal Rehmann. “Preconditioners for Incompressible Navier-Stokes Solvers p.249-250”. In: (2010).
- [30] J. Bell. “Fréchet derivatives and Gâteaux derivatives. Department of Mathematics”. In: (2014).
- [31] Ulbrich M. Hinze M. Pinnau R. and & Ulbrich S. *Optimization with PDE Constraints*. Vol. 23. Springer, 2008.
- [32] C. M. Okubo Junior. “Topology optimization of compressible flows using a discrete adjoint approach”. In: (2022).
- [33] E. Zeidler. *Nonlinear Functional Analysis and its Applications. I, Fixed-Point Theorems*. Springer, 1986.
- [34] K. Svanberg. “The Method of Moving Asymptotes-Modelling aspects and solution schemes”. In: (1998).
- [35] O. Sigmund. “A 99 line topology optimization code written in Matlab”. In: (2018), pp. 120–127. DOI: <https://doi.org/10.1007/s001580050176>.
- [36] S. V. Patankar. “Numerical heat transfer and fluid flow”. In: (1980).

- [37] M. Ur Rehman Segal A. and C. Vuik. “Preconditioners for incompressible Navier-Stokes solvers”. In: *Numerical Mathematics Theory, Methods and Applications*.3.3 (2010), pp. 245–275.
- [38] S. Hamm M. Bollhöfer O. Schenk R. Janalik and K. Gullapalli. *State-of-The-Art Sparse Direct Solvers, Parallel Algorithms in Computational Science and Engineering. Modeling and Simulation in Science, Engineering and Technology, 1-32*. Birkhäuser, 2020, 2020.
- [39] Luca Bergamaschi. “Iterative Methods for Sparse Linear Systems p.15”. In: ().
- [40] C. Vuik. “New insights in GMRES-like methods with variable preconditioners p.192-200”. In: (1994).
- [41] Rohit Chandra et al. *Parallel programming in OpenMP*. Morgan kaufmann, 2001.
- [42] COMSOL Multiphysics®. *AC/DC Module User’s Guide*. Version 6.1. Stockholm, Sweden, 2022.
- [43] Xiaoxia Dai et al. “Topology optimization of steady Navier-Stokes flow via a piecewise constant level set method”. In: *Structural and Multidisciplinary Optimization* 57 (2018), pp. 2193–2203.
- [44] Sunil Kumar Avinash Shukla Anadi Misra. “CHECKERBOARD PROBLEM IN FINITE ELEMENT BASED TOPOLOGY OPTIMIZATION”. In: (2013).
- [45] Z. Liu Y. Deng Y. Wu. *Topology Optimization Theory for Laminar Flow*. Springer Singapore, 2018, pp. 1–4. DOI: <https://doi.org/10.1007/978-981-10-4687-2>.
- [46] Rolf Rannacher. “Finite Element Methods for the Incompressible Navier-Stokes Equations”. In: (1999).

- [47] Rolf Rannacher John G. Heywood and Stefan Turek. “Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations”. In: (1996).
- [48] Rolf Rannacher Heywood John G. and Stefan Turek. “Artificial boundaries and flux and pressure conditions for the incompressible Navier–Stokes equations”. In: *International Journal for numerical methods in fluids* 22.5 (1996), pp. 325–352.
- [49] Michel Fortin and Franco Brezzi. *Mixed and hybrid finite element methods*. Vol. 2. New York: Springer-Verlag, 1991.
- [50] LONG CHEN. “INF-SUP CONDITIONS”. In: (2014).
- [51] Yasuo Koizumi, Tomio Okawa, and Shoji Mori. *Fundamentals of Thermal and Nuclear Power Generation*. Elsevier, 2021.
- [52] Alexander Y. and Leopold G. In: *Physics of Fluids* 2081 (2003). DOI: <https://doi.org/10.1063/1.158012>.
- [53] Eric W. Weisstein. *Bessel Function of the First Kind*. URL: <https://mathworld.wolfram.com/BesselFunctionoftheFirstKind.html>.
- [54] Autodesk. *Autodesk Inventor*. Version 22.0. San Rafael, CA: Autodesk, 2018.
- [55] Fridolin Okkels Laurits Højgaard Olesen and Henrik Bruus. “A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow”. In: (2005).
- [56] R.B. Haber A. Gersborg-Hansen O. Sigmund. “Topology optimization of channel flow problems”. In: (2005).
- [57] Stephen M. Robinson. “Stability Theory for Systems of Inequalities, Part II: Differentiable Nonlinear Systems”. In: (1976).

- [58] S. Kurcyusz J. Zowe. “Regularity and Stability for the Mathematical Programming Problem in Banach Spaces”. In: (1979).
- [59] K. Yosida. *Functional Analysis*. Springer, 1980.
- [60] *Fluid Deformation and Energy Dissipation Derivations*. [https://aguaclara.github.io/Textbook/Fluid\\_Deformation\\_and\\_Energy\\_Dissipation/FDED\\_Derivations.html](https://aguaclara.github.io/Textbook/Fluid_Deformation_and_Energy_Dissipation/FDED_Derivations.html).
- [61] MATLAB. *version R2021b*. Natick, Massachusetts, 2021.
- [62] TYRONE REES CHEN GREIF and DANIEL B. SZYLD. “GMRES WITH MULTIPLE PRECONDITIONERS”. In: ().
- [63] T. Dbouk. “A review about the engineering design of optimal heat transfer systems using topology optimization”. In: (2017).
- [64] Abdelwahed M. and Hassine M. “Topology optimization of time dependent viscous incompressible flows”. In: *Abstract and Applied Analysis 2014* (2014).
- [65] Wu J. Deng Y. Liu Z. and Wu Y. “Topology optimization of steady Navier–Stokes flow with body force”. In: *Computer Methods in Applied Mechanics and Engineering* 255 (2013), pp. 306–321.
- [66] János Lógó, Hussein Ismail, et al. “Milestones in the 150-year history of topology optimization: A review”. In: *Computer Assisted Methods in Engineering and Science* 27.2-3 (2020), pp. 97–132.
- [67] Carlos H Villanueva and Kurt Maute. “CutFEM topology optimization of 3D laminar incompressible flow problems”. In: *Computer Methods in Applied Mechanics and Engineering* 320 (2017), pp. 444–473.