Pontifícia Universidade Católica
do Rio de Janeiro

Javier Aliaga Rivera

Sensitivity Analysis for Design and Control of Thermal-Fluid Systems Using Computational Fluid Dynamics-Based Adjoint Method

Tese de Doutorado

Thesis presented to the Programa de Pós–graduação em Engenharia Mecânica, do Departamento de Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Mecânica.

Advisor: Prof. Marcos Sebastião de Paula Gomes

Rio de Janeiro
May 2022

Pontifícia Universidade Católica
do Rio de Janeiro

Javier Aliaga Rivera

Sensitivity Analysis for Design and Control of Thermal-Fluid Systems Using Computational Fluid Dynamics-Based Adjoint Method

Thesis presented to the Programa de Pós–graduação em Engenharia Mecânica da PUC-Rio in partial fulfillment of the requirements for the degree of Doutor em Engenharia Mecânica. Approved by the Examination Committee:

Prof. Marcos Sebastião de Paula Gomes
Advisor
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Ivan Fabio Mota de Menezes
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Angela Ourivio Nieckele
Departamento de Engenharia Mecânica – PUC-Rio

Prof. Antonio André Novotny
Laboratório Nacional de Computação Científica - MTC

Prof. Emilio Carlos Nelli Silva
Departamento de Engenharia Mecatrônica e Sistemas Mecânicos - USP

Rio de Janeiro, May 31st, 2022

Javier Aliaga Rivera

Graduated in Chemical Engineering by Universidad Nacional del Centro del Perú.

Bibliographic data

Aliaga Rivera, Javier

Sensitivity Analysis for Design and Control of Thermal-Fluid Systems Using Computational Fluid Dynamics-Based Adjoint Method / Javier Aliaga Rivera; advisor: Marcos Sebastião de Paula Gomes. – 2022.

137 f: il. color. ; 30 cm

Tese (doutorado) - Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Mecânica, 2022.

Inclui bibliografia

1. Engenharia Mecânica – Teses. 2. Mecânica dos Fluidos Computacional. 3. Controle de sistemas fluido-térmicos. 4. Método da Adjunta Discreta. 5. Método da Adjunta Continua. 6. Otimização Topológica. 7. Otimização Paramétrica. 8. Método dos Elementos Finitos. 9. Método dos Volumes Finitos. I. de Paula Gomes, Marcos Sebastião. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Mecânica. III. Título.

CDD: 004

To my family, for their support
and encouragement.

Acknowledgments

First and foremost, I would like to express my deep gratitude to my advisor, Professor Marcos Sebastião de Paula Gomes for the stimulus and partnership to carry out this work.

Additionally, I am grateful for all the wonderful professors and fellow students whose encouragement and knowledge I have benefited from during my tenure as a postgraduate student.

Finally, I am indebted to my family, who have provided me with encouragement and support in all aspects of my life. Spacial thanks goes to my mom Gloria, grandparents Javier and Venedicta for their unconditional support and love.

To CNPq and PUC-Rio, for the aids granted, without which this work does not could have been accomplished.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

Abstract

Aliaga Rivera, Javier; de Paula Gomes, Marcos Sebastião (Advisor). **Sensitivity Analysis for Design and Control of Thermal-Fluid Systems Using Computational Fluid Dynamics-Based Adjoint Method**. Rio de Janeiro, 2022. 137p. Tese de Doutorado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

In this work, we perform a sensitivity analysis for the design and control of thermo-fluid systems using computational fluid dynamics based adjoint method. To illustrate the method, the fluid is modeled by the incompressible Navier-Stokes equation adding the Brinkman penalization approach to represent solid material in specific regions of the domain and the heat transfer is modeled by a convection-diffusion equation at steady state. The system of equations is discretized using the finite element method and the finite volume method implemented in Matlab and OpenFOAM, respectively. We consider three types of design (control) parameters with respect to which the sensitivities were determined: first, the velocity components at the inlet boundaries, second, the position and size of the sources acting as flow blockage, and finally, the pseudo-density that determines the material distribution in the computational domain. The sensitivity analysis begins by comparing the continuous and discrete adjoint variables for specific cost functions. We then check the sensitivities by comparing them with sensitivities obtained by the finite difference method. We obtain good agreement in all cases and prove the robustness of the method. Next, two case studies are presented. The optimal location and size of discrete sources in the domain; and the topology optimization of a heat exchanger, considering heat sources in specific regions of the domain as well as convection boundary conditions, aiming to minimize a given cost function while limiting the power dissipation in the channel. Several numerical studies have also been performed for different configurations and operating conditions. The different cases show the performance of the method by applying it to the optimal design with little computational effort.

Keywords

Computational Fluid Dynamics; Thermal-Flow Control; Discrete Adjoint Method; Continuous Adjoint Method; Topology Optimization; Parametric Optimization; Finite Element Method; Finite Volume Method.

Resumo

Aliaga Rivera, Javier; de Paula Gomes, Marcos Sebastião. **Análise de Sensibilidade para o Projeto e Controle de Sistemas Fluido Térmicos Usando o Método Adjunto Baseado em Dinâmica de Fluidos Computacional**. Rio de Janeiro, 2022. 137p. Tese de Doutorado – Departamento de Engenharia Mecânica, Pontifícia Universidade Católica do Rio de Janeiro.

Neste trabalho, realizamos uma análise de sensibilidade para o projeto e controle de sistemas termofluidos usando o método da adjunta baseado em dinâmica de fluidos computacional. Para ilustrar o método, o fluido é modelado pela equação incompressível de Navier-Stokes adicionando a abordagem de penalização de Brinkman para representar o material sólido em regiões específicas do domínio e a transferência de calor é modelada por uma equação de convecção-difusão em estado estacionário. O sistema de equações é discretizado usando o método dos elementos finitos e o método dos volumes finitos implementados no Matlab e OpenFOAM, respectivamente. Consideramos três tipos de parâmetros de projeto (controle) com relação aos quais a sensibilidade é determinada: primeiro, os componentes de velocidade nos limites de entrada, segundo, a posição e o tamanho das fontes que atuam como bloqueio de fluxo e, finalmente, a pseudodensidade que determina a distribuição de material no domínio computacional. A análise de sensibilidade começa comparando as variáveis adjuntas contínuas e discretas para funções de custo específicas. Em seguida, verificamos as sensibilidades comparando-as com as sensibilidades obtidas pelo método de diferenças finitas. Obtivemos boa concordância em todos os casos e comprovamos a robustez do método. A seguir, são apresentados dois estudos de caso. A localização e tamanho ideais de fontes discretas no domínio; e a otimização da topologia de um trocador de calor, considerando fontes de calor em regiões específicas do domínio, bem como condições de contorno de convecção, visando minimizar uma determinada função de custo enquanto limita a dissipação de energia no canal. Vários estudos numéricos também foram realizados para diferentes configurações e condições de operação. Os diferentes casos mostram o desempenho do método aplicando-o ao projeto ótimo com pouco esforço computacional.

Palavras-chave

Mecânica dos Fluidos Computacional; Controle de sistemas fluido-térmicos; Método da Adjunta Discreta; Método da Adjunta Continua; Otimização Topológica; Otimização Paramétrica; Método dos Elementos Finitos; Método dos Volumes Finitos.

Table of contents

List of figures

List of tables

List of algorithms

List of Abreviations

ANN–Artificial Neural Network

CA–Continuous Adjoint

CFD–Computational Fluid Dynamics

DA–Discrete Adjoint

DNS–Direct Numerical Simulation

FD–Finite Difference

FEM–Finite Element Method

FVM–Finite Volume Method

GA– Genetic Algorithm

HVAC – Heating, ventilation, and air conditioning

HCA–Hybrid Cellular Automata

MMA–Method of Moving Asymptote

OC–Optimality Criteria

POD–Proper Orthogonal Decomposition

PDE–Partial Differential Equation

RAMP–Rational Approximation of Material Properties

RANS–Reynolds Averaged Navier Stokes

RSM–Reynolds Stress Mode

SIMP–Solid Isotropic Material with Penalization

SQP–Sequential Quadratic Programming

SIMPLE–Semi-Implicit Method for Pressure-Linked Equations

"The art of structure is where to put the holes"

— **Robert Le Ricolais**, *French-American engineer and philosopher (1894-1977).*

1
Introduction

Here, the term conjugate scalar fluid flow systems refers to processes involving the transport of scalar quantities such as temperature and/or concentration of chemical species within fluids due to thermal and/or mixing interactions between them. This phenomenon is observed in many industrial heat/mixing devices such as heat exchangers, heating, ventilation and air conditioning systems (HVAC), dilution of environmental pollutants, natural degradation of chemicals, etc.

Optimizing the design and control of conjugate scalar flow systems is an active area of research for consistent and cost-effective improvement of such systems. When dealing with a simple system, it is possible to use experience and intuition to improve the system with minor changes through a trial-and-error process. However, if we want to deal with highly complex systems, we need a more structured optimization approach. A general optimization problem is concerned with minimizing a particular cost function bounded by the state system described by partial differential equations. To this end, information is needed about the variation in a set of design (or control) parameters that produces variation in the cost function, i.e., sensitivity.

1.1
Literature Survey

In the literature, we find two main categories of optimization methods: a) Stochastic optimization methods such as Genetic Algorithms (GA) (Xue; Zhai; Chen, 2013) and b) Gradient-based optimization methods such as Adjoint methods (Othmer, 2008). The GA is superior in finding global optimal conditions, but still requires too many computational fluid dynamics (CFD) simulations proportional to the number of design variables. On the other hand, the adjoint method allows the computation of the complete sensitivities with the effort of only two solvers, the so-called primary and adjoint equations, regardless of the dimension of the search space. Although local optima may occur, the computational cost of the adjoint method is less than the GA method, but still substantial. The Proper Orthogonal Decomposition (POD) method (Kutz et al., 2016) reduces the CFD simulation overhead, but the method itself may not be accurate due to the nonlinear nature of fluid flows and unrealistic boundary conditions. Finally, the CFD-based artificial neural network (ANN) (Prieler et al., 2018) method appears to be an attractive

area of research, but still requires too many CFD simulations of multiple representative cases to train the ANN.

Currently, the CFD-based adjoint method remains a suitable approach for optimization in structural mechanics and fluid dynamics, and now the concept is evolving in many directions, such as error estimation, sensitivity analysis, and uncertainty quantification in most engineering fields.

Depending on the type of design variables, the adjoint method is mainly used for shape, topology, flow control and parametric optimization problems.

In shape optimization, the contour boundaries of the system are morphed to improve the objective function. The computational domain changes at each optimization cycle using a re-meshing or mesh deformation tool based on surface sensitivities. However, in industrial cases, re-meshing and mapping the results can significantly increase the computational cost. In this regard Soto; Löhner; Yang (2004) has presented a complete CFD design methodology. In steady flows, shape optimization is applied in vehicle ducts to minimize the total pressure drop through a tube or to achieve a uniform distribution of species in a given region (Helgason; Krajnović, 2012; Helgason; Krajnović, 2014; Helgason, 2015), and also in airfoils to improve lift and drag coefficients (Schramm; Stoevesandt; Peinke, 2018). In addition, we find works in unsteady flows for optimal control (Nadarajah; Jameson, 2007), in non-viscous (Wang; Mavriplis; Anderson, 2010) and viscous flows (Srinath; Mittal, 2010; Yamaleev; Diskin; Nielsen, 2010). In 2018, Skinner; Zare-behtash provided an overview of the prevailing optimization approaches integrated with aerodynamic theory for the purpose of shape optimization, and a comprehensive but straightforward insight for non-experts and a reference detailing the current state for experienced practitioners. Kenway et al. (2019) proposed an open benchmark and reported a comprehensive evaluation of the various approaches to adjoint implementation.

Topology optimization has its roots in the optimization of load-bearing structures. In this respect, in recent decades compact Matlab implementation codes have been presented for minimizing the compliance of statically loaded structures (Sigmund, 2001; Andreassen et al., 2011), also using isoparametric polygonal elements (Talischi et al., 2012b) and in three-dimensional domains (Liu; Tovar, 2014). Fanni; Shabara; Alkalla (2013), performed a comparison between different topology optimization algorithms, such as the Method of Moving Asymptotes (MMA), Sequential Quadratic Programming (SQP), Optimality Criteria (OC), and Hybrid Cellular Automata (HCA). The Method of Moving Asymptotes provides good control of the convergence, stability, and speed of the optimization process. In thermal conduction systems,

topology optimization has been shown to yield unconventional tree-like optimal structures of highly conductive material for efficient heat transfer (Subramaniam; Dbouk; Harion, 2018), e.g., in electrical projector housings (Menge et al., 2018). In flow problems, Peskin (1972) studied the flow of a viscous incompressible fluid in a region with immersed boundaries that move with the fluid and exert forces on the fluid. Then, Goldstein; Handler; Sirovich (1993) modeled a no-slip flow boundary with an external force field. They used techniques related to Peskin's immersed boundary approach to introduce solid surfaces into a simulated flow field. Years later, we found the seminal work on topology optimization by Borrvall; Petersson (2003) for Stokes flow and based on lubrication theory. Then it was extended to Navier-Stokes equations by Gersborg-hansen; Sigmund; Haber (2005). These approaches had similarity to the Brinkman penalization term introduced in the momentum equation to calculate whether a fluid cell is favorable or counterproductive to the flow. Nowadays, the Brinkman penalization approach is the subject of ongoing research, for example, Spietz; Hejlesen; Walther (2017) has presented an iterative Brinkman penalization for simulating an impulsive flow through a sphere and a circular disk. Sharaborin; Rogozin; Kasimov (2021) coupled the Volume of Fluid and Brinkman Penalization methods to simulate incompressible multiphase flows. They developed numerical algorithms for direct simulation of three-dimensional incompressible multiphase flows in the presence of multiple fluids and solids. Fuchsberger et al. (2022) incorporated obstacles into a fluid flow problem using a Navier-Stokes-Brinkman penalization approach. Going back to topology optimization, in 2008 Othmer derived the adjoint equations and boundary conditions based on a continuous adjoint formulation, demonstrating the versatility of the approach with respect to changes in the objective function. Since then, several studies have been developed, e.g. parallel optimization framework in C++ using MMA (Aage; Lazarov, 2013), implementation of the level set method for surface representation and evolution (interface between solid and fluid) (Karpouzas; Villiers, 2014). In the laminar regime, Nogueira (2016) has developed the design of rotors of radial flow machines. Behrou; Ranjan; Guest (2019) developed a methodology for adaptive explicit no-slip boundary conditions that adaptively removes elements in the solid regions using a density-based method and mass flow constraints. Alonso et al. (2019) used density-based optimization to the design of Tesla-type centrifugal pumps without blades. Gaymann; Montomoli; Pietropaoli (2019) applied a density-based method to the design of diode valves at medium to high Reynolds numbers. In unsteady flows, Nørgaard et al. (2019) applied

automatic differentiation to topology optimization in an unsteady oscillating pressure pump. Skaar (2017) used topology optimization for biomedical flows. For turbulent flows, Dilgen et al. (2018) applied automatic differentiation to obtain accurate sensitivities using the one-equation Spalart-Allmaras model and the two-equation $k-\omega$ model. As Kontoleontos et al. (2013) did for shape optimization, they showed that the "frozen turbulence" assumption leads to inaccurate sensitivities for topology optimization. Yoon (2016) applied the $k-\epsilon$ model to design 2D flow components that minimize turbulent energy, i.e., noise. For work with non-Newtonian fluids, Kian (2017) applied topology optimization to channel design, and Dong; Liu (2019) presented a bi-objective formulation for the design of asymmetric fixed-geometry microvalves. The main advantage of topology optimization is that design changes can be made without changing the mesh, which saves a lot of computational effort. On the other hand, an inherent feature of topology optimization is the ragged surface of the resulting geometries. Therefore, in many cases, it is most efficient to start with topology optimization and then switch to shape optimization to solve the optimization problem (Ruberto, Ingegneria Aeronautica, Politécnico Di Milano, 2017). In recent research by Karpouzas (2019), shape and topology optimization methods were combined in a way that preserves the strengths of both methods. "In coupled thermo-fluid systems, complex channels, fractals, ribs, and rows of fluid manifolds are examples of the structures in the final geometries, so it is still a complex object of study" (Pietropaoli; Montomoli; Gaymann, 2019). With the advent of additive manufacturing techniques, relatively complex geometries can be produced, and the ability to increase the structural complexity and efficiency of mechanical components is expected to have a relevant impact on cooling system technology. With this in mind, a great deal of research has been conducted on this topic, e.g. for application in microchannels at low Reynolds numbers (Dede, 2009; Koga, 2010; Yoon; Koo, 2019), in automotive exahust systems (Hinterberger; Olesen, 2011), in turbines to minimize stagnation pressure dissipation while maximizing heat transfer between fluid and solid regions (Pietropaoli; Montomoli; Gaymann, 2019), in forced convection flow in cooling channels (Goeke; Wünsch, 2017), in the heat transfer process by natural convection in a rectangular cavity (Ruberto, Ingegneria Aeronautica, Politécnico Di Milano, 2017), and in unsteady flows (Kavvadias et al., 2015). Subramaniam; Dbouk; Harion (2019), worked with multi-objective functions combining pressure drop reduction and thermal power maximization in incompressible flows from low to medium Reynolds numbers. The two objectives were linearly combined using the weighted sum method, and several optimal designs were produced for different combinations

of these weighting factors.Yu et al. (2020) studied thermal-fluid-structural problems for designing a three-dimensional heat sink with load transfer in a parallel solver. In addition, the Darcy number was studied because it was found that even if the Darcy number is low enough to penalize the near-zero velocity, it cannot eliminate the thermal convection term in the solid region. Chen; Xia; Zhao (2020) implemented a topology optimization to design highly conductive fins in a multi-tube thermochemical heat storage system. Tawk; Ghannam; Nemer (2019) developed a density-based approach to optimize heat exchangers with two separate fluids and a solid. Li et al. (2019) presented extensive numerical and experimental comparisons in the design of liquid-cooled heat sinks using a multi-objective density-based method. Dong; Liu (2020) investigated topology optimization to air-cooled microchannel heat sinks with discrete heat sources. Hu; Zhang; Li (2020) performed an in-depth comparison of an optimized microchannel heat sink with a reference straight-channel heat sink.

In parametric optimization, the system is described by some parameters that may refer to the geometry, material, physical properties, size and location of the source and inlets, among others. The value of these parameters varies until an optimized configuration is reached. Particularly interesting studies relate to heat and mass transfer to create the desired closed environments using the inverse design method, as shown by Liu (2017) and Nabi; Grover; Caulfield (2017). Liu et al. (2016) used the CFD-based adjoint method to determine the optimal location, size, and parameters of air inlets in a two-dimensional cavity. In doing so, they calculated the sensitivities over the positions of nodal meshes in the framework of FEM (Schneider; Jimack, 2008), which implies the use of mesh adjustments. On the same topic, Zhao et al. (2017) used an adaptive step size to adjust the design parameters, which had a significant impact on the computation time but affected the convergence stability. In addition, Kouhi et al. (2017), presented a framework based on the discrete adjoint method for evaluating the sensitivities of a number of parameters related to chemically reacting flows and Zhao et al. (2018) developed an area-constrained topology and cluster analysis to consolidate multiple air drain holes into a small number of inlets and also determine their size and location. This method was applied in a two-person office and in a crowded single-aisle aircraft cabin.

In flow control, the focus of optimization is on the load history of the systems. In certain cases, control can be performed in real time. In this case, the optimization process is called active control. For example, Gunzburger; Svobodny (1990) studied the problem of minimizing viscous drag

on a body by adding or removing mass through the boundary. Later, Joslin et al. (1997), studied the injection or suction of fluid through a single opening at the boundary for the problem of suppressing boundary layer instability. Meanwhile, numerical methods for automatic detection and containment of one or more point sources of contaminants in closed conduits, indoor environments, and environmental streams are also of global interest (Katopodes, N. D., & Piasecki, 1996; Warnock, 2013; Wang, 2015). Rimer (2016), developed intelligent public infrastructure systems to control hazardous releases while protecting passengers in civil infrastructure systems.

1.2 Problems and Opportunities of Research

In the literature review on optimization with the adjoint method in fluid dynamics, we have found gaps, both in theoretical developments and applications, that are worth exploring. In the following, we address the issues considered in this work:

(i) Many engineering optimization applications require the use of unstructured meshes to accurately describe the geometry of the design domain and define the boundary conditions. Polygonal meshes have been shown to be advantageous over triangular and quadrilateral meshes in CFD simulations because they provide a better approximation of the velocity and pressure fields. In addition, there is less susceptibility to numerical instabilities such as checkerboard patterns because there are more degrees of freedom for velocity than for pressure for elements with many sides. (Talischi et al., 2012a). Pereira et al. (2016), presented a Matlab implementation of topology optimization for flow problems in the educational computational PolyTop/PolyMesher code. Polygonal finite elements were used to obtain a stable low-order discretization of the Stokes equations for incompressible viscous flows. Unfortunately, the applications of Stokes flows are rather limited. Motivated by the modularity and flexibility offered by the PolyTop/PolyMesher philosophy, we therefore envision the possibility of exploring such a framework for other phenomena, e.g., for the stationary Navier-Stokes equations in conjunction with scalar transport equations for optimization problems.

(ii) Alexandersen; Andreasen (2020), provided a review of the literature on topology optimization of fluid-based problems related to: pure fluid flow, species transport, conjugate heat transfer, fluid-structure interaction,

microstructure, and porous media. The review covered 186 papers from 2003 to January 2020, inclusive, and found that FEM is the most commonly used discretization method with 76%. Surprisingly, FVM, the preferred discretization method for computational fluid dynamics, appears in only 7% of the papers. This could be because FEM is the preferred solid mechanics method from which topology optimization starts; discrete adjoint approaches are simpler with FEM than with FVM; also, FEM has gained stability and accuracy. In addition, the review found that the vast majority of studies with 85% used a steady-state laminar flow model, only 7% considered a transient laminar flow model, and 3% treated turbulent flow. The last two cases are most likely due to the complex formulation and the huge increase in computational cost. Therefore, both FEM and FVM should be investigated when determining sensitivities using the adjoint method, and the strengths of both methods should be considered.

(iii) When we implement an adjoint method, we must decide whether to use the discrete or the continuous approach. The respective advantages and disadvantages of these two alternatives are the subject of ongoing discussion (Gunzburger, 2002). In the continuous approach, first the primary equations are linearized, then the adjoint equations are derived from the linearized primary equations, and finally the adjoint equations are discretized. The discrete method starts with the discretization of the primary equations, which are linearized and then finally transposed. The discrete formulation is able to provide the exact sensitivities and the treatment of the boundary conditions is quite simple, unlike the continuous formulation. However, the continuous adjoint formulation can be easily implemented in a C++ framework such as OpenFOAM. Moreover, it allows the derivation of the sensitivities of different design variables and can be quickly adapted to a wide range of objective functions. In this context, the continuous adjoint boundary conditions as derived by Othmer (2008) need to be investigated for their accuracy and stability with respect to their implementation. Then, a comparison between the continuous and discrete adjoint variables for specific cost functions needs to be explored.

(iv) After reviewing the literature on forced convection heat exchanger design using topology optimization (Goeke; Wünsch, 2017; Ruberto, Ingegneria Aeronautica, Politécnico Di Milano, 2017; Kavvadias et al., 2015; Subramaniam; Dbouk; Harion, 2019; Tawk; Ghannam; Nemer,

2019; Li et al., 2019; Yu et al., 2020; Chen; Xia; Zhao, 2020; Dong; Liu, 2020; Hu; Zhang; Li, 2020), all of them consider a heat source in the whole domain and none of them consider convection boundary conditions. It can be seen that there is a scientific gap in considering heat sources in certain regions of the domain and convection boundary conditions.

(v) The optimal location, strength, size, and shape of sources in a flow domain are critical for controlling flow, indoor thermal comfort, and improving air quality. In the literature, we have found papers that focuses only on the optimal location, strength, size, and shape of inlets (Liu et al., 2016; Schneider; Jimack, 2008; Zhao et al., 2017; Zhao et al., 2018). In this context, we have found the potential and practical applications in source position optimization, namely, shaping of impermeable inner walls in the fluid flow region. We can model interesting solid shapes with a number of small neighboring sources.

1.3 Main Contributions of the work

In what follows, each point discussed in Section 1.2 refers to each contribution to this thesis:

(i) We implemented a Matlab code for the optimization of conjugate scalar flow problems based on the educational PolyTop/PolyMesher code (Talischi et al., 2012a; Pereira et al., 2016) and on the FEM implementation guide code of (Carvalho; Valério, 2012);

(ii) Reviewing the literature, we found that the topic of optimization of fluid dynamic systems under turbulent conditions has been the focus of attention of researchers (Alexandersen; Andreasen, 2020). However, we performed simulations in a laminar regime to avoid the difficulties associated with RANS turbulence models, e.g., the wall function law, in order to better characterize the problem and understand the concept of adjoint equations in fluid dynamic problems. We have used both FEM and FVM separately to determine the sensitivities and show the advantage of using both methods together, i.e., FVM to solve the primal equations and FEM to solve the adjoint equations;

(iii) Nadarajah; Jameson (2000), presented a complete formulation for both the Continuous Adjoint (CA) and Discrete Adjoint (DA) approaches in automatic aerodynamic design using the Euler equations. They concluded that, as expected, the DA gradients had better agreement than

the CA gradients when compared to the finite difference (FD) method, but the difference was generally small. Therefore, they recommended using the DA approach to verify the CA formulation. In this work, we performed a comparison between the continuous and discrete adjoint variables in thermo-fluid systems for specific cost functions and considering the optimization of various parameters;

(iv) The design of heat exchangers using topology optimization, taking into account heat sources in specific regions of the domain and convection boundary conditions, are cases that complement what is presented in the state of the art on the subject of optimization of conjugate heat transfer systems. This is one of the objectives of the present work, a clear scientific contribution in this topic;

(v) A new contribution of this work is the determination of the sensitivity of an arbitrary objective function with respect to the position of sources in a fluid flow system, whether dependent or independent of the state variables. We also show possible applications using several examples. First, one-dimensional examples of a scalar transport equation are presented, followed by an extension to two dimensions. The study cases presented in this work, show that the proposed method complements the well-known shape and topology optimization in flow systems.

1.4 Objectives

1.4.1 General Objective

The objective of this work is to perform a sensitivity analysis for the design and control of thermal-fluid systems in laminar regime and steady state, in particular to apply the methodology to the optimization of the position of sources and the topology of heat exchangers.

1.4.2 Specific Objectives

- To compare the continuous and discrete adjoint methods using the finite volume and finite element methods;
- To derive the mathematical expressions for the sensitivities of the objective functions with respect to the design variables using the adjoint method and compare the results with the finite difference method;
- To validate the optimization method using inverse design with attainable design objectives.
- To perform several numerical studies to optimize the position of sources and topology of heat exchangers for various configurations and operating conditions.

1.5 Organization of the work

The main part of the work is divided into six chapters (including the introduction and conclusions):

Chapter 2 presents the governing equations and describes the adjoint optimization method in its discrete and continuous forms for generic objective functions. Then, in Chapter 3, the sensitivities are derived using the continuous adjoint method. Chapter 4 presents the numerical methods and algorithms. In Chapter 5, the FE code implemented in polygonal grids is verified and a sensitivity analysis of thermal-fluid systems is performed by examining the continuous and discrete adjoint variables in the context of FEM and FVM. In Chapter 6, we present the analysis of two case studies, namely source design optimization and topology optimization of a heat exchanger with line heat source and convection boundary condition. Finally, in Chapter 7, conclusions are drawn and some suggestions for future work are presented.

2
Conceptual Framework

First, we consider an optimization problem involving a scalar transport equation in a fluid flow system. In this work, a scalar transport equation is used to describe heat and mass transfer, mainly through diffusion and convection mechanisms. Next, we assume that the objective function $\mathcal{J}$ is defined as follows:

$$\mathcal{J} = \mathcal{J}(\boldsymbol{u}, p, \phi, \varrho) \tag{2-1}$$

and the system of governing equations by:

$$\mathbf{R}(\boldsymbol{u}, p, \phi, \varrho) = \mathbf{0} \tag{2-2}$$

where $\boldsymbol{u}, p$ and ϕ represent velocity, pressure, and scalar state variables, respectively, and ϱ represents design variables, e.g., inlet boundary, sources, material distribution, geometry, etc.

Thus, the constrained optimization problem can be expressed as follows:

find $\boldsymbol{u}, p, \phi$ *and* ϱ *such that the functional* $\mathcal{J}$ *given in Eq.* (2-1) *is minimized subject to the requirement that Eq.* (2-2) *is satisfied.*

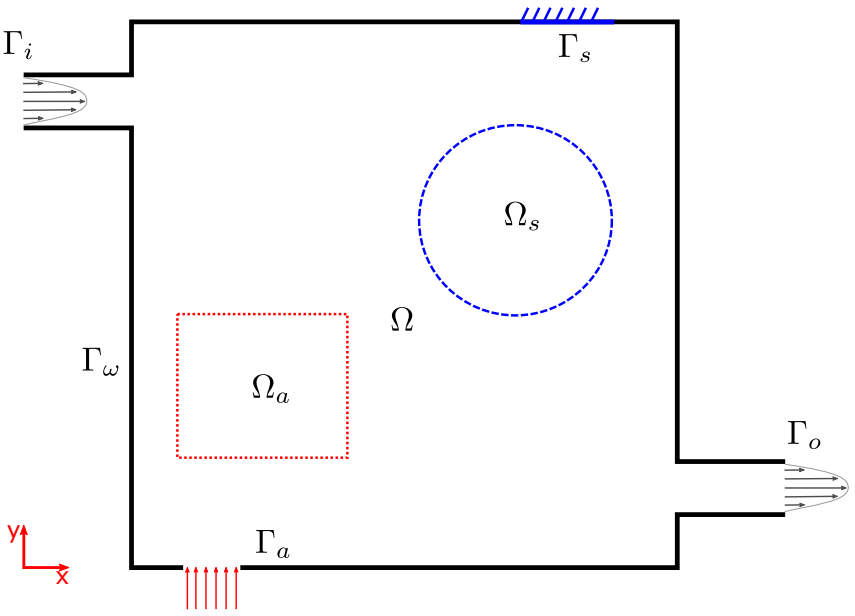

Figure 2.1: A sketch of the "sensing" subdomains $\Omega_s \subset \Omega$ and $\Gamma_s \subset \Gamma$, where we specify the objective function, and the "actuating" subdomains $\Omega_a \subset \Omega$ and $\Gamma_a \subset \Gamma$, where we specify the design and control parameters. Γ_i and Γ_o represents the inlet and the outlet flow respectively.

2.1 The State Equations

Let us denote the flow domain by Ω sketched in Fig. 2.1, and let Γ denote its boundary. The inflow and outflow parts of the boundaries are denoted by Γ_i and Γ_o, respectively, and Γ_a is considered as a subset of the boundary Γ to which a design could be applied, e.g., a finite connected part of the wall boundary Γ_ω. Similarly, Ω_a is a region of the domain Ω that is a target subdomain for optimization.

The flow field is described by the fields of velocity $\boldsymbol{u}$ and pressure p, which result from solving the Navier-Stokes and continuity equations. The scalar field ϕ results from the solution of the scalar transport equation. Therefore, in this work, we represent the set of governing equations in terms of the steady state as follows:

$$\boldsymbol{R}(\boldsymbol{u}, p, \phi, \varrho) = \begin{pmatrix} -\nabla \cdot (2\nu \mathrm{D}(\boldsymbol{u})) + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \dfrac{\nabla p}{\rho} - \boldsymbol{f}_{\Omega_a} \\ \nabla \cdot \boldsymbol{u} \\ -\nabla \cdot (D \nabla \phi) + \boldsymbol{u} \cdot \nabla \phi - f_{\Omega_a} \end{pmatrix} = \boldsymbol{0} \tag{2-3}$$

where $\boldsymbol{f}_{\Omega_a}$ and f_{Ω_a} stand for sources defined in the subdomain Ω_a and can depend on the state variables $\boldsymbol{u}$ and ϕ, respectively, $\nu = \mu/\rho$ is the kinematic viscosity, μ is the dynamic viscosity (sometimes called absolute viscosity), ρ is the density, D is the diffusion coefficient, and the rate of the strain tensor is denoted by $\mathrm{D}(\boldsymbol{u}) = 1/2(\nabla \boldsymbol{u} + \nabla^{\mathsf{T}} \boldsymbol{u})$. Also, we note that the letters **bold** denote vectors and $(\cdot)$ is the dot product operator.

The boundary conditions to solve the set of PDEs (2-3) (finding the fields $\boldsymbol{u}, p$ and ϕ) must be defined. To simplify our formulation, we assume **Dirichlet** and **Neumann** boundary conditions in this work. For the case of Fig. 2.1, we use the following set of boundary conditions:

$$\boldsymbol{u}|_{\Gamma_i} = \boldsymbol{u}_i; \qquad \boldsymbol{n} \cdot \nabla p|_{\Gamma_i} = 0 \quad \text{and} \quad \boldsymbol{n} \cdot \nabla \phi|_{\Gamma_i} = 0 \tag{2-4}$$

$$\boldsymbol{u}|_{\Gamma_\omega} = \boldsymbol{0}; \qquad \boldsymbol{n} \cdot \nabla p|_{\Gamma_\omega} = 0 \quad \text{and} \quad \boldsymbol{n} \cdot \nabla \phi|_{\Gamma_\omega} = 0 \tag{2-5}$$

$$\begin{pmatrix} \boldsymbol{u} \\ \phi \end{pmatrix}\Bigg|_{\Gamma_a} = \begin{pmatrix} \boldsymbol{g}_{\Gamma_a} \\ g_{\Gamma_a} \end{pmatrix} \quad \text{and} \quad \boldsymbol{n} \cdot \nabla p|_{\Gamma_a} = 0 \tag{2-6}$$

where $\boldsymbol{g}_{\Gamma_a}$ and g_{Γ_a} are defined on the subdomain Γ_a. For the outflow, the **Neumann** boundary condition is commonly used. This boundary condition is called the "do-nothing" boundary condition because it naturally occurs when integrating by parts using FEM.

$$\nu(\nabla \boldsymbol{u}) \cdot \boldsymbol{n} - \frac{1}{\rho} p \, \boldsymbol{n}|_{\Gamma_o} = 0 \tag{2-7}$$

$$\boldsymbol{n} \cdot \nabla \phi|_{\Gamma_o} = 0 \tag{2-8}$$

where $\boldsymbol{n}$ is the outward unit normal.

The governing equations (2-3) can be applied to a variety of heat and mass transfer problems.

2.1.1 General Considerations

Let us first simplify the flow model without much loss of understanding by making the following basic assumptions:

- Simulation of steady state at low Reynolds number, i.e. flow in laminar regime, as it is common in micro devices;
- The fluid is Newtonian, so the rate of shear stress of the fluid is linearly related to the angular deformation;
- The fluid is incompressible, so the density of the fluid is constant;
- Constant physical properties;
- The diffusion coefficient D appearing in Eq. (2-3) is $D = k/\rho C_p$ for the heat transport model, where k is the thermal conductivity and C_p is the specific heat capacity;
- To account for the presence of immersed solid regions in the fluid flow domain, the momentum equation is penalized by a source term (Brinkman penalization approach), i.e., a friction force proportional to the fluid velocity:

  $$\boldsymbol{f}_{\Omega_a} = -\alpha \boldsymbol{u} \tag{2-9}$$

  where α is the inverse of the local permeability, which allows to distinguish between areas of low and high permeability. In the fluid regions, α is equal to 0 and means that no artificial frictional force is added. In the solid regions, α is equal to the relatively large value of order $O(10^5)$ to set the velocity to near zero.

2.1.2 Distance function, level set and implicit representation

To represent more complex solid regions in the domain, we use the Heaviside function ($H_e(d_{\Omega_a})$), which is defined as:

$$H_e(d_{\Omega_a}) = \begin{cases} 0, & d_{\Omega_a} < -\epsilon, \\ \frac{1}{2}\left(1 + \frac{d_{\Omega_a}}{\epsilon} + \frac{1}{\pi} sin\left(\frac{\pi d_{\Omega_a}}{\epsilon}\right)\right), & |d_{\Omega_a}| \leq \epsilon, \\ 1, & d_{\Omega_a} > \epsilon, \end{cases} \tag{2-10}$$

where ϵ is the interface thickness and d_{Ω_a} is the signed distance function. The Heaviside function has values of zero and one outside the interface thickness ($|d_{\Omega_a}| \leq \epsilon$) and connects seamlessly to zero and one inside the interface thickness. The interface thickness should be approximately chosen to be $\epsilon \approx 1.5\Delta x$, where Δx is the cell size. In this work, we use $\epsilon = 0.0001m$ in all simulations.

Then, the solid part is expressed as:

$$\boldsymbol{f}_{\Omega_a} = -\alpha \boldsymbol{u} H_e(d_{\Omega_a}) \tag{2-11}$$

The signed distance function d_{Ω_a} is a scalar field, which associated with Ω_a, is the mapping $d_{\Omega_a} : \mathbb{R}^2 \to \mathbb{R}$ defined by:

$$d_{\Omega_a}(\mathbf{x}) = s_{\Omega_a}(\mathbf{x}) \min_{\mathbf{y} \in \partial\Omega_a} \|\mathbf{x} - \mathbf{y}\| \tag{2-12}$$

where $\partial\Omega_a$ denotes the boundary of Ω_a. The absolute field value $d_{\Omega_a}(\mathbf{x})$ at any point $\mathbf{x}$ in space is the minimal distance from $\mathbf{x}$ to $\mathbf{y}$ at the boundary of the domain. Formally, we have used the Euclidean norm. The sign function is given by:

$$s_{\Omega_a}(\mathbf{x}) = \begin{cases} -1, & \mathbf{x} \in \Omega_a \\ +1, & \mathbf{x} \in \mathbb{R}^2 \setminus \Omega_a \end{cases} \tag{2-13}$$

For points on $\partial\Omega_a$ itself, we clearly have $d_{\Omega_a} = 0$, so the signed distance function implicitly represents a surface as the zero value contour. Moreover, $d_{\Omega_a}(\mathbf{x})$ is often signed to determine whether $\mathbf{x}$ is inside or outside Ω_a. The following characterizations follow directly from this definition:

$$\bar{\Omega}_a = \left\{\mathbf{x} \in \mathbb{R}^2 : d_{\Omega_a}(\mathbf{x}) \leq 0\right\}, \tag{2-14}$$

$$\partial\Omega_a = \left\{\mathbf{x} \in \mathbb{R}^2 : d_{\Omega_a}(\mathbf{x}) = 0\right\} \tag{2-15}$$

So one task is to construct $d_{\Omega_a}(\mathbf{x})$ for a particular region we wish to represent. For simple geometries such as a circle and a rectangle, the signed distance function is easily identified using analytic expressions. For a circle of radius r whose center is the point $\mathbf{x}_o$, the distance function is given by:

$$d_{\Omega_a} = \|\mathbf{x} - \mathbf{x}_o\| - r \tag{2-16}$$

For a rectangle of size $2r_a \times 2r_b$ centered at the origin, the distance function is given by:

$$\begin{aligned} d_{\Omega_a}(x, y) = &min(max(|x| - r_a, |y| - r_b), 0) \\ &+ \|max(|x| - r_a, 0), max(|y| - r_b, 0)\|_2 \end{aligned} \tag{2-17}$$

Moreover, we can combine the regions $d_{\Omega_{a1}}$ and $d_{\Omega_{a2}}$ with the well-known set operations: Union, Intersection, Complement, and Difference. For signed distance functions, these can be implemented with min and max functions as follows:

$$d_{\Omega_{a1} \cup \Omega_{a2}}(\mathbf{x}) = min(d_{\Omega_{a1}}(\mathbf{x}), d_{\Omega_{a2}}(\mathbf{x})) \tag{2-18}$$

$$d_{\Omega_{a1} \cap \Omega_{a2}}(\mathbf{x}) = max(d_{\Omega_{a1}}(\mathbf{x}), d_{\Omega_{a2}}(\mathbf{x})) \tag{2-19}$$

$$d_{\mathbb{R}^2 \setminus \Omega_{a1}}(\mathbf{x}) = -d_{\Omega_{a1}}(\mathbf{x}) \tag{2-20}$$

$$d_{\Omega_{a1} - \Omega_{a2}}(\mathbf{x}) = max(d_{\Omega_{a1}}(\mathbf{x}), -d_{\Omega_{a2}}(\mathbf{x})) \tag{2-21}$$

Shapes can be rotated and moved by applying the inverse transformation to the input area:

$$rotate(\Omega_{a1}, R) = d_{\Omega_{a1}}(R^{\mathsf{T}}\mathbf{x}) \tag{2-22}$$

$$translate(\Omega_{a1}, T) = d_{\Omega_{a1}}(\mathbf{x} - T) \tag{2-23}$$

Rotation and translation are examples of isometric transformations, i.e. they do not change the distance between two points.

2.2 Optimization Gradient-Based Methods

Many practical optimization algorithms require at least an approximation to the gradient of the function with respect to the control or design variable ϱ, i.e., the sensitivity $\mathcal{D}\mathcal{J}/\mathcal{D}\varrho$. In the following sections, we will describe the methods used in this work to determine sensitivity.

2.2.1 The Finite Difference Method

The finite difference method is the simplest way for this task. Here we calculate the functional at a value $\tilde{\varrho}$ near ϱ and then use a difference quotient, via the formula:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\varrho} \approx \frac{\mathcal{J}(\mathbf{z}(\tilde{\varrho}), \tilde{\varrho}) - \mathcal{J}(\mathbf{z}(\varrho), \varrho)}{\tilde{\varrho} - \varrho} \tag{2-24}$$

where $\mathbf{z} = (\boldsymbol{u}, p, \phi)^{\mathsf{T}}$ denotes the set of fields from the solution of the state equations $\boldsymbol{R}(\mathbf{z}, \varrho) = 0$.

Another alternative in the search for the calculation of sensitivity is the calculation of the total derivative of $\mathcal{J}(\mathbf{z}(\varrho), \varrho)$ considering the chain rule.

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\varrho} = \frac{\partial \mathcal{J}}{\partial \mathbf{z}} \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\varrho} + \frac{\partial \mathcal{J}}{\partial \varrho} \tag{2-25}$$

The terms $\partial\mathcal{J}/\partial\mathbf{z}$ and $\partial\mathcal{J}/\partial\varrho$ are usually "easy" to determine. We still need to specify how to determine the sensitivities $\mathrm{d}\mathbf{z}/\,\mathrm{d}\varrho$, where we could again use the difference quotient approximation or alternatively differentiate the equation of state $\boldsymbol{R}(\mathbf{z}, \varrho) = 0$ by the chain rule to obtain:

$$\frac{\partial \boldsymbol{R}}{\partial \mathbf{z}} \frac{\mathrm{d}\mathbf{z}}{\mathrm{d}\varrho} = -\frac{\partial \boldsymbol{R}}{\partial \varrho} \tag{2-26}$$

Note that $\partial\boldsymbol{R}/\partial\mathbf{z}$ and $\partial\boldsymbol{R}/\partial\varrho$ depend only on ϱ and $\mathbf{z}(\varrho)$. Then one can solve for the sensitivity $(\partial\mathbf{z}/\partial\varrho)$.

2.2.2 The Discrete Adjoint Method

The gradient of the functional $\mathcal{J}$ can be determined by solving the adjoint equations. Recall the expressions (2-25) and (2-26) to eliminate the sensitivity expression $\mathrm{d}\mathbf{z}/\,\mathrm{d}\varrho$, resulting in two equations:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\varrho} = -\boldsymbol{\Lambda}^{\mathsf{T}} \frac{\partial \boldsymbol{R}}{\partial \varrho} + \frac{\partial \mathcal{J}}{\partial \varrho} \tag{2-27}$$

and

$$\left(\frac{\partial \boldsymbol{R}}{\partial \mathbf{z}}\right)^{\mathsf{T}} \boldsymbol{\Lambda} = \left(\frac{\partial \mathcal{J}}{\partial \mathbf{z}}\right)^{\mathsf{T}} \tag{2-28}$$

The last equation Eq. (2-28) is known as a discrete adjoint equation, which starts with the discretization of the primary equations, then is linearized and finally transposed. The adjoint equation must behave as follows:

$$\boldsymbol{\Lambda} \approx \frac{\partial \mathcal{J}}{\partial \boldsymbol{R}} \tag{2-29}$$

That is, we can say that the adjoint represents the sensitivity of the functional to perturbations in $\boldsymbol{R}$. Moreover, the adjoint equation can be extended to problems with nonlinear residuals (such as the Navier-Stokes equation). This is done by computing the derivative $\partial\boldsymbol{R}/\partial\mathbf{z}$ after a certain point in the state space (just as we could compute the derivative of any nonlinear function).

We note that in general there are several methods by which $\partial \boldsymbol{R}/\partial \mathbf{z}$ and $\partial \mathcal{J}/\partial \mathbf{z}$ can be computed. These methods include finite difference, automatic code differentiation, and analytic differentiation (in fact, $\partial \boldsymbol{R}/\partial \mathbf{z}$ is the 'Jacobian matrix' in the finite element discretization process). In this work, we rely primarily on analytic differentiation with FEM, which usually leads to a more accurate and efficient implementation despite the effort involved.

Equation (2-27) is an expression for the gradient of the functional with respect to the solution of the adjoint equation (2-28). The optimality condition states that $\boldsymbol{\Lambda}$ reaches its optimal value when the functional gradient is zero. Here, Eqs. (2-3), (2-27) and (2-28) form a coupled optimality system and can be solved using the "one-shot approach" or Sequential Quadratic Programming (SQP).

To understand the adjoint meaning as described by (Kast, 2016), consider the discretization of the scalar transport equation from Eq. (2-3), we obtain a system of equations of the form:

$$\underline{\boldsymbol{A}}\boldsymbol{\Phi} = \mathbf{F} \tag{2-30}$$

where $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is a matrix representing the operator $-\nabla \cdot (D\nabla) + \boldsymbol{u} \cdot \nabla$, $\boldsymbol{\Phi} \in \mathbb{R}^N$ is a vector of unknowns representing ϕ, $\mathbf{F} \in \mathbb{R}^N$ is a vector of source and Dirichlet boundary condition information, and N is the number of discretization nodes. Now, if we are interested in computing the entire solution $\boldsymbol{\Phi}$, i.e., finding all components of the $\boldsymbol{\Phi}$ vector, then we actually need to know the entire $\boldsymbol{A}^{-1}$ matrix, since we would then compute $\boldsymbol{\Phi}$ from:

$$\boldsymbol{\Phi} = \underline{\boldsymbol{A}}^{-1}\mathbf{F} \tag{2-31}$$

However, we might be interested in only one component of $\boldsymbol{\Phi}$ rather than the entire solution. In this case, we do not need to know the entire $\underline{\boldsymbol{A}}^{-1}$ matrix; it would give us less information. For example, imagine that we are interested in computing Φ_n, the *nth* entry in the $\boldsymbol{\Phi}$ vector. This situation is represented in the following explicit matrix:

$$\underbrace{\begin{bmatrix}\Phi_1\\ \Phi_2\\ \vdots\\ \Phi_n\\ \vdots\\ \Phi_N\end{bmatrix}}_{\Phi} = \underbrace{\begin{bmatrix}\blacksquare & \blacksquare & \cdots & \blacksquare\\ \blacksquare & \blacksquare & \cdots & \blacksquare\\ \vdots & \vdots & \ddots & \vdots\\ a_{n,1} & a_{n,2} & \ddots & a_{N,N}\\ \vdots & \vdots & \ddots & \vdots\\ \blacksquare & \blacksquare & \cdots & \blacksquare\end{bmatrix}}_{\underline{\mathrm{A}}^{-1}} \underbrace{\begin{bmatrix}F_1\\ F_2\\ \vdots\\ F_n\\ \vdots\\ F_N\end{bmatrix}}_{\boldsymbol{F}} \tag{2-32}$$

By the properties of matrix multiplication, it is clear that to compute Φ_n

we need to know only one row of $\underline{\boldsymbol{A}}^{-1}$, namely the *nth* row. This *nth* line, in conjunction with the vector $\mathbf{F}$, contains all the information we need to calculate our desired "output" $\mathcal{J} = \Phi_n$ (objective function).

This sensitivity vector in the *nth* line is commonly referred to in the literature as the "dual vector", "adjoint output vector", or simply as the "adjoint" $\mathbf{\Lambda}^{\mathsf{T}}$

Using the above notation, the output $\mathcal{J}$ can be written as the inner product between the adjoint and $\mathbf{F}$ vectors, i.e., as:

$$\mathcal{J} = \mathbf{\Lambda}^{\mathsf{T}}\mathbf{F} \tag{2-33}$$

This is called the "dual form" of the output and is merely a mathematical reformulation of our earlier assertion that the only information needed to compute the output is the *nth* line and the $\mathbf{F}$ vector.

For our purposes, there are two other properties of this line that are even more important. For example, imagine that the first component of $\mathbf{\Lambda}^{\mathsf{T}}$ is zero. Then, if we were to change the first component of $\mathbf{F}$, i.e., F_1, the value of $\mathcal{J}$ would not change at all (since F_1 would simply be multiplied by 0 in the calculation of $\mathcal{J}$). In other words, we could say that our output $\mathcal{J}$ is insensitive to changes in F_1. Similarly, this logic could be applied to all entries of $\mathbf{\Lambda}^{\mathsf{T}}$: The smaller a given entry is, the less sensitive $\mathcal{J}$ is to changes in the corresponding F_n component, and conversely, the larger a given entry is, the more sensitive it is. In summary, $\mathbf{\Lambda}^{\mathsf{T}}$ provides not only the information needed to compute $\mathcal{J}$, but also the sensitivity of $\mathcal{J}$ to perturbations of the $\mathbf{F}$ components in **size** or magnitude.

Second, imagine that the vector $\mathbf{F}$ has only one nonzero component along the entire vector, namely F_s; this means, for example, a source at position x_s in the domain; moreover, the components of $\mathbf{\Lambda}^{\mathsf{T}}$ are all of the same size. Thus, if we were to change the position of the source term F_s, the value of $\mathcal{J}$ would not change at all (since F_s would simply be multiplied by the same value when $\mathcal{J}$ is calculated). In other words, we could say that our output $\mathcal{J}$ is insensitive to changes in the position of F_s. Similarly, this logic could be applied if the components of $\mathbf{\Lambda}^{\mathsf{T}}$ were not all similar: The more constant the entries are, the less sensitive $\mathcal{J}$ is to changes in position of the corresponding F_s component, and conversely, the larger the variation of the entries, the more sensitive it is. Then $\mathbf{\Lambda}^{\mathsf{T}}$ also provides the sensitivity of $\mathcal{J}$ to perturbations in the **site** or position of the source term F_s, namely, any variation in the position of F_s produces a variation in $\mathcal{J}$ proportional to the gradient of $\mathbf{\Lambda}^{\mathsf{T}}$ at position x_s

To formally define the adjoint, it is helpful to first calculate the derivative of the output $\mathcal{J}$ with respect to Φ. Since in this case $\mathcal{J} = \Phi_n$, we can write its

derivative with respect to Φ as follows:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{\Phi}} \equiv \left[\frac{\partial \mathcal{J}}{\partial \Phi_1} \quad \frac{\partial \mathcal{J}}{\partial \Phi_2} \quad \cdots \quad \frac{\partial \mathcal{J}}{\partial \Phi_n} \quad \cdots \quad \frac{\partial \mathcal{J}}{\partial \Phi_N}\right] \tag{2-34}$$

This results in a row vector where all entries are zero and the *nth* entry is one. We thus see that the adjoint $\mathbf{\Lambda}^\mathsf{T}$ can be defined as follows:

$$\mathbf{\Lambda}^\mathsf{T} = \frac{\partial \mathcal{J}}{\partial \mathbf{\Phi}} \underline{\boldsymbol{A}}^{-1} \tag{2-35}$$

So far, we do not seem to have gained much with the above derivations. However, an important fact is that while this example output $\mathcal{J} = \Phi_n$ may seem trivial, both Eq. (2-35) (the so-called "adjoint equation") and Eq. (2-33) (the "dual form") hold regardless of the desired output.

Strictly speaking, Eq. (2-33) (the dual form) holds only when $\mathcal{J}$ is a linear combination of the components of $\mathbf{\Phi}$, although Eq. (2-35) (the adjoint equation) also holds in the nonlinear case, since the adjoint is always defined as a linear sensitivity vector. For example, instead of a linear "one-component" output such as $\mathcal{J} = \Phi_n$, we might be interested in computing an output that minimizes the square of the difference between Φ_n and a reference value Φ_r, i.e., an error functional.

$$\mathcal{J} = \frac{1}{2}(\Phi_n - \Phi_r)^2 \tag{2-36}$$

In this case, the output derivative results,

$$\frac{\partial \mathcal{J}}{\partial \mathbf{\Phi}} \equiv \begin{bmatrix} 0 & 0 & \cdots & (\Phi_n - \Phi_r) & \cdots & 0 \end{bmatrix} \tag{2-37}$$

Substituting Eq. (2-37) into Eq. (2-35), the adjoint gives the corresponding *nth* line of $\underline{\boldsymbol{A}}^{-1}$ times the difference $(\Phi_n - \Phi_r)$, and this states: the smaller the difference, the closer to the minimum of $\mathcal{J}$ we are.

Generalizing even further, we would typically write the set of discrete "residuals" from Eq. (2-30) like this:

$$\mathbf{R}(\mathbf{\Phi}) \equiv \underline{\boldsymbol{A}}\mathbf{\Phi} - \mathbf{F} = 0 \tag{2-38}$$

where the residual vector $\mathbf{R} \in \mathbb{R}^N$. Taking the derivative of $\mathbf{R}$ with respect to $\mathbf{\Phi}$, we obtain:

$$\frac{\partial \mathbf{R}}{\partial \mathbf{\Phi}} = \underline{\boldsymbol{A}} \tag{2-39}$$

So we see that the $\underline{\boldsymbol{A}}$ matrix corresponds to the "residual Jacobian" matrix, $\partial\mathbf{R}/\partial\mathbf{\Phi}$. Substituting this Jacobian matrix into the adjoint equation (2-35), we obtain:

$$\mathbf{\Lambda}^\mathsf{T} = \frac{\partial \mathcal{J}}{\partial \mathbf{\Phi}} \frac{\partial \mathbf{R}^{-1}}{\partial \mathbf{\Phi}} \tag{2-40}$$

To get a more general form of this equation, we transpose both sides and get:

$$\boldsymbol{\Lambda} = \frac{\partial \mathbf{R}^{-\mathsf{T}}}{\partial \boldsymbol{\Phi}} \frac{\partial \mathcal{J}^{\mathsf{T}}}{\partial \boldsymbol{\Phi}} \tag{2-41}$$

Finally, multiplying both sides by the Jacobian transpose gives:

$$\frac{\partial \mathbf{R}^{\mathsf{T}}}{\partial \boldsymbol{\Phi}} \boldsymbol{\Lambda} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{\Phi}}^{\mathsf{T}} \tag{2-42}$$

This is the adjoint equation and is essentially the same as the equation (2-28). This general form suggests to us that we can perturb not only the source and boundary terms appearing in $\mathbf{F}$, but also any variable related to the residue $\mathbf{R}$, such as physical properties (for topology optimization) and geometry variables (for shape optimization).

2.2.3 The Continuous Adjoint Method

Calculating the sensitivities of the objective functions is the most important part of a gradient-based optimization method. The adjoint method is the most efficient method for calculating the sensitivities to deal with problems where the number of design variables is larger than the number of objective functions. As mentioned in the introduction, the adjoint method can be either discrete or continuous. The present work follows the continuous adjoint approach.

The adjoint equations are derived by introducing a Lagrangian function $\mathcal{L}$ and reformulating the cost function as follows:

$$\begin{aligned} \mathcal{L}(\boldsymbol{u}, p, \phi, \varrho, \boldsymbol{v}, q, \varphi, \boldsymbol{s}, s) = & \mathcal{J}(\boldsymbol{u}, p, \phi, \varrho) - \int_{\Omega} (\boldsymbol{v}, q, \varphi)^{\mathsf{T}} \boldsymbol{R}(\boldsymbol{u}, p, \phi, \varrho) \, \mathrm{d}\Omega \\ & - \int_{\Gamma_a} \boldsymbol{s}(\boldsymbol{u} - \boldsymbol{g}_{\Gamma_a}) \, \mathrm{d}\Gamma - \int_{\Gamma_a} s(\phi - g_{\Gamma_a}) \, \mathrm{d}\Gamma \end{aligned} \tag{2-43}$$

where $\boldsymbol{v}, q$ and φ are Lagrange multipliers used to enforce the residuals $\boldsymbol{R}$ (the constraint equations). We also include the Lagrange multipliers $\boldsymbol{s}$ and s to enforce the boundary conditions on Γ_a corresponding to the state variables $\boldsymbol{u}$ and ϕ, respectively. Again, ϱ stands for the control or design parameters. These include anything that can be used to satisfy the objective, such as the injection velocity of a fluid injected into a stream or the parameters used to describe the geometry.

The introduction of Lagrange multipliers transforms the constrained optimization problem into an unconstrained problem:

find $\boldsymbol{u}, p, \phi, \varrho, \boldsymbol{v}, q, \varphi, \boldsymbol{s}$ *and* s *satisfying Eq.*(2-3)-(2-8) *such that the Lagrangian functional* $\mathcal{L}$ *given by Eq.*(2-43) *is rendered stationary.*

In this new formulation, each argument of the Lagrangian functional is considered an independent variable (subject only to the constraints (2-4)-(2-8)), so that each variable can be varied independently.

The total variation of $\mathcal{L}$, ignoring geometry deformation, gives:

$$\begin{aligned}\delta\mathcal{L} =&\delta_\varrho\mathcal{J} + \delta_{\boldsymbol{u}}\mathcal{J} + \delta_p\mathcal{J} + \delta_\phi\mathcal{J} - \int_\Omega (\boldsymbol{v}, q, \varphi)^\mathsf{T}(\delta_\varrho\boldsymbol{R} + \delta_{\boldsymbol{u}}\boldsymbol{R} + \delta_p\boldsymbol{R} \\ &+ \delta_\phi\boldsymbol{R})\,\mathrm{d}\Omega - \int_{\Gamma_a} \boldsymbol{s}\,\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_{\Gamma_a} \boldsymbol{s}\,(\delta_\varrho\boldsymbol{g}_{\Gamma_a} + \delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a})\,\mathrm{d}\Gamma \\ &- \int_{\Gamma_a} s\,\delta\phi\,\mathrm{d}\Gamma + \int_{\Gamma_a} s\,(\delta_\varrho g_{\Gamma_a} + \delta_\phi g_{\Gamma_a})\,\mathrm{d}\Gamma\end{aligned} \tag{2-44}$$

Since changes in ϱ due to the equations of state entail variations in the flow and scalar fields, the total variation of $\mathcal{L}$ also includes contributions from the changes in $\boldsymbol{u}$, p and ϕ. Rearranging Eq. (2-44) and collecting all terms with δ_ϱ, we obtain:

$$\begin{aligned}\delta\mathcal{L} =&\Big[\delta_\varrho\mathcal{J} - \int_\Omega (\boldsymbol{v}, q, \varphi)^\mathsf{T}\,\delta_\varrho\boldsymbol{R}\,\mathrm{d}\Omega + \int_{\Gamma_a} \boldsymbol{s}\,\delta_\varrho\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma + \int_{\Gamma_a} s\,\delta_\varrho g_{\Gamma_a}\,\mathrm{d}\Gamma\Big] \\ &+ \Big[\delta_{\boldsymbol{u}}\mathcal{J} + \delta_p\mathcal{J} + \delta_\phi\mathcal{J} - \int_\Omega (\boldsymbol{v}, q, \varphi)^\mathsf{T}(\delta_{\boldsymbol{u}}\boldsymbol{R} + \delta_p\boldsymbol{R} + \delta_\phi\boldsymbol{R})\,\mathrm{d}\Omega \\ &- \int_{\Gamma_a} \boldsymbol{s}\,\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_{\Gamma_a} \boldsymbol{s}\,\delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma - \int_{\Gamma_a} s\,\delta\phi\,\mathrm{d}\Gamma + \int_{\Gamma_a} s\,\delta_\phi g_{\Gamma_a}\,\mathrm{d}\Gamma\Big]\end{aligned} \tag{2-45}$$

Let us choose the adjoints $\boldsymbol{v}$, q and φ so that the second term in parentheses in Eq. (2-45) vanishes. Thus, we obtain the stationary adjoint equations in the general form.

$$\begin{aligned}&\delta_{\boldsymbol{u}}\mathcal{J} + \delta_p\mathcal{J} + \delta_\phi\mathcal{J} - \int_\Omega (\boldsymbol{v}, q, \varphi)^\mathsf{T}(\delta_{\boldsymbol{u}}\boldsymbol{R} + \delta_p\boldsymbol{R} + \delta_\phi\boldsymbol{R})\,\mathrm{d}\Omega \\ &- \int_{\Gamma_a} \boldsymbol{s}\,\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_{\Gamma_a} \boldsymbol{s}\,\delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma - \int_{\Gamma_a} s\,\delta\phi\,\mathrm{d}\Gamma + \int_{\Gamma_a} s\,\delta_\phi g_{\Gamma_a}\,\mathrm{d}\Gamma = 0\end{aligned} \tag{2-46}$$

The variations $\delta_{\boldsymbol{u}}$, δ_p and δ_ϕ are arbitrary functions such that $\boldsymbol{u}+\delta\boldsymbol{u}$, $p+\delta p$ and $\phi + \delta\phi$ satisfy the governing equations (2-3).

The variations of $\boldsymbol{R}$ (Eq. (2-3)) with respect to $\boldsymbol{u}$, p and ϕ gives:

$$\delta_{\boldsymbol{u}}\boldsymbol{R} = \begin{pmatrix} -\nabla\cdot(2\nu\mathrm{D}(\delta\boldsymbol{u})) + (\delta\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + (\boldsymbol{u}\cdot\nabla)\delta\boldsymbol{u} - \frac{\partial \boldsymbol{f}_{\Omega_a}}{\partial \boldsymbol{u}}\cdot\delta\boldsymbol{u} \\ \nabla\cdot\delta\boldsymbol{u} \\ \delta\boldsymbol{u}\cdot\nabla\phi \end{pmatrix} = 0 \tag{2-47}$$

$$\delta_p\boldsymbol{R} = \begin{pmatrix} \frac{1}{\rho}\nabla\delta p \\ 0 \\ 0 \end{pmatrix} = 0 \tag{2-48}$$

$$\delta_\phi\boldsymbol{R} = \begin{pmatrix} 0 \\ 0 \\ -\nabla\cdot(D\nabla\delta\phi) + \boldsymbol{u}\cdot\nabla\delta\phi - \frac{\partial f_{\Omega_a}}{\partial\phi}\delta\phi \end{pmatrix} = 0 \tag{2-49}$$

and, the decomposition of the functional $\mathcal{J}$ into interior and boundary parts, renders:

$$\mathcal{J} = \int_\Omega \mathcal{J}_\Omega \, \mathrm{d}\Omega + \int_\Gamma \mathcal{J}_\Gamma \, \mathrm{d}\Gamma \tag{2-50}$$

next, we insert Eqs. (2-47), (2-48), (2-49) and (2-50) into Eq. (2-46) and using integration by parts, in order to factorize the $\delta\boldsymbol{u}$, δp and $\delta\phi$ terms, we get the following equality:

$$\begin{aligned}
&\int_\Omega \Big[\frac{\partial \mathcal{J}_\Omega}{\partial \boldsymbol{u}} - \Big(-\nabla\cdot(2\nu \mathrm{D}(\boldsymbol{v})) - (\boldsymbol{u}\cdot\nabla)\boldsymbol{v} - \nabla\boldsymbol{v}\cdot\boldsymbol{u} + \frac{\nabla q}{\rho} - \nabla\varphi\phi \\
&- \frac{\partial \boldsymbol{f}_{\Omega_a}}{\partial \boldsymbol{u}}\boldsymbol{v}\Big)\Big]\cdot\delta\boldsymbol{u}\ \mathrm{d}\Omega + \int_\Gamma 2\nu\boldsymbol{n}\cdot D(\delta\boldsymbol{u})\cdot\boldsymbol{v}\ \mathrm{d}\Gamma \\
&+ \int_\Gamma \Big[\frac{\partial \mathcal{J}_\Gamma}{\partial \boldsymbol{u}} - \Big(2\nu\boldsymbol{n}\cdot\mathrm{D}(\boldsymbol{v}) + \boldsymbol{v}(\boldsymbol{u}\cdot\boldsymbol{n}) + \boldsymbol{n}(\boldsymbol{v}\cdot\boldsymbol{u}) - \frac{q\boldsymbol{n}}{\rho} \\
&+ \varphi\,\phi\,\boldsymbol{n}\Big)\Big]\delta\boldsymbol{u}\,\mathrm{d}\Gamma - \int_{\Gamma_a} \boldsymbol{s}\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_{\Gamma_a} \boldsymbol{s}\delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma \\
&+ \int_\Omega \Big(\frac{\partial \mathcal{J}_\Omega}{\partial p} - (-\nabla\cdot\boldsymbol{v})\Big)\delta p\,\mathrm{d}\Omega + \int_\Gamma \Big(\frac{\partial \mathcal{J}_\Gamma}{\partial p} - \boldsymbol{v}\cdot\boldsymbol{n}\Big)\delta p\,\mathrm{d}\Gamma \\
&+ \int_\Omega \Big[\frac{\partial \mathcal{J}_\Omega}{\partial \phi} - \Big(-\nabla\cdot(D\nabla\varphi) - \nabla\cdot(\boldsymbol{u}\varphi) - \frac{\partial f_{\Omega_a}}{\partial\phi}\varphi\Big)\Big]\delta\phi\,\mathrm{d}\Omega \\
&+ \int_\Gamma \varphi\,\boldsymbol{n}\cdot D\nabla\,\delta\phi\,\mathrm{d}\Gamma + \int_\Gamma \Big[\frac{\partial \mathcal{J}_\Gamma}{\partial\phi} - (\boldsymbol{n}\cdot D\nabla\varphi + \boldsymbol{u}\cdot\boldsymbol{n}\varphi)\Big]\delta\phi\ \mathrm{d}\Gamma \\
&- \int_{\Gamma_a} s\,\delta\phi\ \mathrm{d}\Gamma + \int_{\Gamma_a} s\delta_\phi g_{\Gamma_a}\,\mathrm{d}\Gamma = 0,
\end{aligned} \tag{2-51}$$

The equation (2-51) holds for admissible perturbations $\delta\boldsymbol{u}$, δp and $\delta\phi$ such that the perturbed quantities $\boldsymbol{u}+\delta\boldsymbol{u}$, $p+\delta p$ and $\phi+\delta\phi$ satisfy the primal equation system of equations (2-3). This can be achieved only if the integrals vanish one by one. Therefore,the interior integrals must be zero, which leads to the system of adjoint equations.

$$\begin{aligned}
&-\nabla\cdot(2\nu\mathrm{D}(\boldsymbol{v})) - \nabla\boldsymbol{v}\cdot\boldsymbol{u} - (\boldsymbol{u}\cdot\nabla)\boldsymbol{v} \\
&+\frac{\nabla q}{\rho} + \varphi\nabla\phi - \frac{\partial \boldsymbol{f}_{\Omega_a}}{\partial\boldsymbol{u}}\boldsymbol{v} = \frac{\partial \mathcal{J}_\Omega}{\partial\boldsymbol{u}}\Big|_{\Omega_s}
\end{aligned} \tag{2-52}$$

$$(-\nabla\cdot\boldsymbol{v}) = \frac{\partial \mathcal{J}_\Omega}{\partial p}\Big|_{\Omega_s} \tag{2-53}$$

$$-\nabla\cdot(D\nabla\varphi) - \nabla\cdot(\boldsymbol{u}\varphi) - \frac{\partial f_{\Omega_a}}{\partial\phi}\varphi = \frac{\partial \mathcal{J}_\Omega}{\partial\phi}\Big|_{\Omega_s} \tag{2-54}$$

2.2.3.1 Adjoint Boundary Conditions

From Eq. (2-51), remains the following boundary integrals:

$$\int_\Gamma \left[\frac{\partial \mathcal{J}_\Gamma}{\partial \boldsymbol{u}} - \Big(2\nu \boldsymbol{n}\cdot \mathrm{D}(\boldsymbol{v}) + \boldsymbol{v}(\boldsymbol{u}\cdot\boldsymbol{n}) + \boldsymbol{n}(\boldsymbol{v}\cdot\boldsymbol{u}) - \frac{q\boldsymbol{n}}{\rho} + \varphi\phi\boldsymbol{n}\Big)\right]\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_\Gamma 2\nu\boldsymbol{n}\cdot\mathrm{D}(\delta\boldsymbol{u})\cdot\boldsymbol{v}\,\mathrm{d}\Gamma - \int_{\Gamma_a} \boldsymbol{s}\,\delta\boldsymbol{u}\,\mathrm{d}\Gamma + \int_{\Gamma_a} \boldsymbol{s}\delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma = 0 \tag{2-55}$$

$$\int_\Gamma \Big(\frac{\partial \mathcal{J}_\Gamma}{\partial p} - \boldsymbol{v}\cdot\boldsymbol{n}\Big)\delta p\,\mathrm{d}\Gamma = 0 \tag{2-56}$$

$$\int_\Gamma \varphi\boldsymbol{n}\cdot D\nabla\delta\phi\,\mathrm{d}\Gamma + \int_\Gamma \left[\frac{\partial \mathcal{J}_\Gamma}{\partial \phi} - (\boldsymbol{n}\cdot D\nabla\varphi + \boldsymbol{u}\cdot\boldsymbol{n}\varphi)\right]\delta\phi\,\mathrm{d}\Gamma - \int_{\Gamma_a} s\delta\phi\,\mathrm{d}\Gamma + \int_{\Gamma_a} s\delta_\phi g_{\Gamma_a}\,\mathrm{d}\Gamma = 0 \tag{2-57}$$

Othmer (2008), has explained the viscosity term (second integral involving the rate of strain tensor D) of Eq. (2-55) in some detail to show that the following relation holds:

$$\int_\Gamma 2\nu\boldsymbol{n}\cdot\mathrm{D}(\delta\boldsymbol{u})\cdot\boldsymbol{v}\,\mathrm{d}\Gamma = \int_\Gamma \nu(\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u}\cdot\boldsymbol{v}\,\mathrm{d}\Gamma \tag{2-58}$$

In this way, the adjoint boundary conditions for the inlet, the wall, the outlet and parts of the domain boundary Γ_a are now evaluated.

Walls and inlet. First, we take the variation of the state variables corresponding to the boundaries Γ_ω (Eq. (2-4)) and Γ_i (Eq. (2-5)), resulting:

$$\delta\boldsymbol{u}|_{\Gamma_i} = 0; \qquad \boldsymbol{n}\cdot\nabla\delta p|_{\Gamma_i} = 0 \quad \text{and} \quad \boldsymbol{n}\cdot\nabla\delta\phi|_{\Gamma_i} = 0 \tag{2-59}$$

$$\delta\boldsymbol{u}|_{\Gamma_\omega} = 0; \qquad \boldsymbol{n}\cdot\nabla\delta p|_{\Gamma_\omega} = 0 \quad \text{and} \quad \boldsymbol{n}\cdot\nabla\delta\phi|_{\Gamma_\omega} = 0 \tag{2-60}$$

Next, we take the variation of the continuity equation,

$$\nabla\cdot\delta\boldsymbol{u} = (\boldsymbol{n}\cdot\nabla)\delta u_n + \nabla_t\cdot\delta\boldsymbol{u}_t = 0 \tag{2-61}$$

where ∇_t is the tangential component of ∇, also u_n and $\boldsymbol{u}_t$ are the normal and tangential components of the velocity $\boldsymbol{u}$. Then, since $\delta\boldsymbol{u}_t = 0$ along the Dirichlet boundary. It follows that $(\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u}_n = 0$, and $\nabla\cdot\delta\boldsymbol{u} = (\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u}_t$. Therefore, the boundary integrals (2-55), (2-56) and (2-57) reduce to:

$$\int_\Gamma \nu(\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u}_n\cdot\boldsymbol{v}_t\,\mathrm{d}\Gamma = 0 \tag{2-62}$$

$$\int_\Gamma \Big(\frac{\partial \mathcal{J}_\Gamma}{\partial p} - \boldsymbol{v}\cdot\boldsymbol{n}\Big)\delta p\,\mathrm{d}\Gamma = 0 \tag{2-63}$$

$$\int_\Gamma \left[\frac{\partial \mathcal{J}_\Gamma}{\partial \phi} - (\boldsymbol{n}\cdot D\nabla\varphi + \boldsymbol{u}\cdot\boldsymbol{n}\varphi)\right]\delta\phi\,\mathrm{d}\Gamma = 0 \tag{2-64}$$

Hence, since $(\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u}_n$, δp and $\delta\phi$ are arbitrary values, we can obtain:

$$\boldsymbol{v}_t|_{\Gamma_i-\Gamma_\omega} = 0 \tag{2-65}$$

$$\boldsymbol{v}_{\boldsymbol{n}}|_{\Gamma_i-\Gamma_\omega} = \frac{\partial \mathcal{J}_\Gamma}{\partial p}\Big|_{\Gamma_s} \tag{2-66}$$

$$(\boldsymbol{n}\cdot D\nabla\varphi + \boldsymbol{u}\cdot\boldsymbol{n}\varphi)|_{\Gamma_i-\Gamma_\omega} = \frac{\partial \mathcal{J}_\Gamma}{\partial \phi}\Big|_{\Gamma_s} \tag{2-67}$$

Note that the velocity $\boldsymbol{u}$ is zero at the wall boundary. To ensure the well-formedness of the adjoint system of equations, the zero gradient of q at the wall and inlet boundaries is used in a similar way as its primal counterpart p in the primal Navier-Stokes equations.

$$\boldsymbol{n}\cdot\nabla q|_{\Gamma_i-\Gamma_\omega} = 0 \tag{2-68}$$

Inlet controls. Again, we take the variation of the state variables corresponding the the boundary Γ_a (Eq. (2-4)).

$$\begin{pmatrix}\delta\boldsymbol{u}\\ \delta\phi\end{pmatrix}\Big|_{\Gamma_a} = \begin{pmatrix}\delta_{\boldsymbol{u}}\boldsymbol{g}_{\Gamma_a}\\ \delta_\phi g_{\Gamma_a}\end{pmatrix} \quad \text{and} \quad \boldsymbol{n}\cdot\nabla\delta p|_{\Gamma_a} = 0 \tag{2-69}$$

If the functions $\boldsymbol{g}_{\Gamma_a}$ and g_{Γ_a} are simply prescribed values (i.e., they are not functions of $\boldsymbol{u}$, ϕ and its derivatives), the adjoint boundary conditions are obtained similarly to the previous procedure:

$$\boldsymbol{v}|_{\Gamma_a} = 0 \tag{2-70}$$

$$\varphi|_{\Gamma_a} = 0 \tag{2-71}$$

$$\boldsymbol{n}\cdot\nabla q|_{\Gamma_a} = 0 \tag{2-72}$$

However, if $\boldsymbol{g}_{\Gamma_a}$ and g_{Γ_a} are functions of $\boldsymbol{u}$, ϕ and its derivatives, we should take a little more trouble to obtain the adjoint boundary conditions.

Outlets. Taking the variation of the state variables corresponding to the boundary Γ_o (Eqs. (2-7) and (2-8)), yields:

$$\nu(\boldsymbol{n}\cdot\nabla)\delta\boldsymbol{u} = 0 \tag{2-73}$$

$$\boldsymbol{n}\cdot\nabla\delta\phi = 0 \tag{2-74}$$

The pressure p is usually given at the outlet, so $\delta p = 0$. After simplifying the integrals (2-55),(2-56) and (2-57), and since $\delta\boldsymbol{u}$ and $\delta\phi$ are arbitrary values, we get:

$$\Big[\nu(\boldsymbol{n}\cdot\nabla)\boldsymbol{v} + \boldsymbol{v}(\boldsymbol{u}\cdot\boldsymbol{n}) + \boldsymbol{n}(\boldsymbol{v}\cdot\boldsymbol{u}) + \varphi\phi\boldsymbol{n} - \frac{q\boldsymbol{n}}{\rho}\Big]\Big|_{\Gamma_o} = \frac{\partial \mathcal{J}_\Gamma}{\partial \boldsymbol{u}}\Big|_{\Gamma_s} \tag{2-75}$$

$$\Big(\boldsymbol{n}\cdot D\nabla\varphi + \boldsymbol{u}\cdot\boldsymbol{n}\varphi\Big)|_{\Gamma_o} = \frac{\partial \mathcal{J}_\Gamma}{\partial \phi}\Big|_{\Gamma_s} \tag{2-76}$$

For functions $\boldsymbol{g}_{\Omega_a}$ and g_{Ω_a} that are independent of the state variables $\boldsymbol{u}$ and ϕ, respectively, we obtain the adjoints $\boldsymbol{s}$ and s as follows:

$$-\Big[2\nu\boldsymbol{n}\cdot\mathrm{D}(\boldsymbol{v})+\boldsymbol{v}(\boldsymbol{u}\cdot\boldsymbol{n})+\boldsymbol{n}(\boldsymbol{v}\cdot\boldsymbol{u})-\frac{q\boldsymbol{n}}{\rho}+\varphi\phi\boldsymbol{n}\Big]|_{\Gamma_a}=\boldsymbol{s} \tag{2-77}$$

$$-\Big[\boldsymbol{n}\cdot D\nabla\varphi+\boldsymbol{u}\cdot\boldsymbol{n}\varphi\Big]|_{\Gamma_a}=s \tag{2-78}$$

We have obtained the general form of the adjoint system of equations for the stationary incompressible Navier-Stokes equations and the scalar transport equations. The structure of the adjoint Navier-Stokes equations and the scalar transport equations is very similar to that of the primal equations. The main difference between the primal and adjoint equations is that the adjoint equations are always linear, new terms appear as sources because of the objective functions, and the minus sign before the convective term indicates that the adjoint information is moving upstream and not downstream.

2.2.3.2 The Optimality Condition

The first set of terms in parentheses of Eq. (2-45) remains. It is usually called the optimality condition, where ϱ is a representation for the design variables.

$$\delta_\varrho\mathcal{L}=\delta_\varrho\mathcal{J}-\int_\Omega(\boldsymbol{v},q,\varphi)^\mathsf{T}\,\delta_\varrho\boldsymbol{R}\,\mathrm{d}\Omega+\int_{\Gamma_a}\boldsymbol{s}\,\delta_\varrho\boldsymbol{g}_{\Gamma_a}\,\mathrm{d}\Gamma+\int_{\Gamma_a}s\,\delta_\varrho g_{\Gamma_a}\,\mathrm{d}\Gamma \tag{2-79}$$

Equation (2-79) is an expression for the gradient of the functional in terms of the solution of the system of adjoint equations (2-52)-(2-53). The optimality condition states that the adjoints $(\boldsymbol{v},q,\varphi)$ reach their optimal values when the gradient of the Lagrangian function is zero. Thus, Eqs. (2-3), (2-52)-(2-53) and (2-79) form a coupled optimality system and can be solved using the "one-shot approach" or Sequential Quadratic Programming (SQP).

However, for complex PDEs, such as the Navier-Stokes equation, the resulting coupled system of equations would be complex and difficult to solve using the the "one-shot approach" method. Instead, we can use optimization algorithms where sensitivity allows us to update the design variable, which in turn helps us calculate the primary and adjoint variables in each cycle.

Figure 2.2, shows the flowchart of the optimization process, where the blocks represent computational components and the arrows indicate the data flow. In the flowchart we find three main modules: the primal solver, the adjoint solver and the optimization solver.

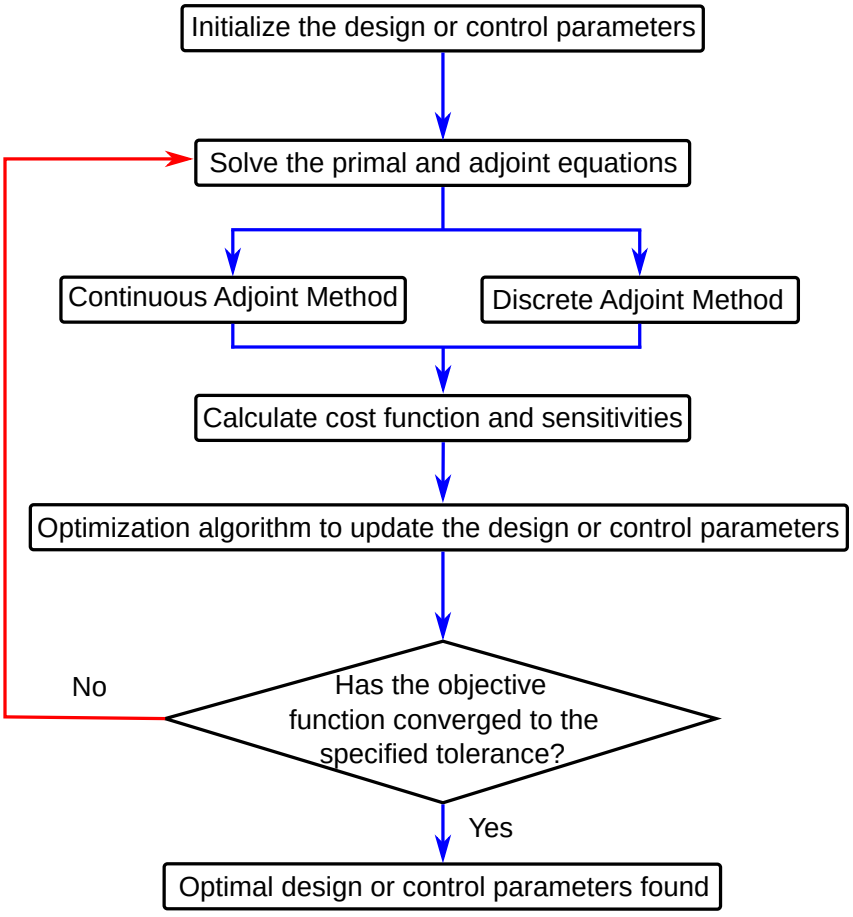

Figure 2.2: Flowchart of the optimization process.

3
Sensitivities Derivation

3.1
The Objective Function

Typical cost functions for external aerodynamics are drag and lift force. For ducts, the objective may be to minimize losses through the duct, maximize scalar mixing, or achieve uniform flow at the outlet. This section presents the cost functions implemented in the current work; each of them is described below.

3.1.1
Error functional

Set the velocity, pressure, and scalar fields to have a specific value at a specific location of the doamin is an important design criterion, e.g. to avoid risks from the spread of pollutants and for thermal comfort in closed environments.

Suppose that Ω_s is a finite, connected part of the inner domain Ω that is disjoint from Ω_a, and that Γ_s is a finite, connected part of the boundary Γ that is disjoint from Γ_a. Then the domain Ω_s and the boundary Γ_s can be thought of as places where we sensor the primal state variables, i.e., along which we wish to fit the primal state variables to given functions $\mathbf{z}_{\Omega_r}$ and $\mathbf{z}_{\Gamma_r}$, respectively. Thus, consider the functional of the form:

$$\mathcal{J}(\mathbf{z}) = \frac{\sigma_1}{2}\int_{\Omega_s}(\mathbf{z}-\mathbf{z}_{\Omega_r})^2\,\mathrm{d}\Omega + \frac{\sigma_2}{2}\int_{\Gamma_s}(\mathbf{z}-\mathbf{z}_{\Gamma_r})^2\,\mathrm{d}\Gamma \tag{3-1}$$

where $\mathbf{z}$ is a vector of the primal state variables.

$$\mathbf{z} = \begin{pmatrix} \boldsymbol{u} \\ \phi \end{pmatrix} \tag{3-2}$$

$\mathbf{z}_{\Omega_r}$ and $\mathbf{z}_{\Gamma_r}$ are given functions defined on Ω_s and Γ_s, respectively, and are defined as:

$$\mathbf{z}_{\Omega_r} = \begin{pmatrix} \boldsymbol{u}_{\Omega_r} \\ \phi_{\Omega_r} \end{pmatrix} \quad \text{and} \quad \mathbf{z}_{\Gamma_r} = \begin{pmatrix} \boldsymbol{u}_{\Gamma_r} \\ \phi_{\Gamma_r} \end{pmatrix} \tag{3-3}$$

The parameters σ_1, σ_2 can be used to adjust the relative importance of the terms occurring in the functional and to ensure consistency of units. Then we calculate the required derivatives needed by the adjoint system as follows:

$$\frac{\partial \mathcal{J}_\Omega}{\partial \mathbf{z}} = \sigma_1(\mathbf{z} - \mathbf{z}_{\Omega_r}), \tag{3-4}$$

$$\frac{\partial \mathcal{J}_\Gamma}{\partial \mathbf{z}} = \sigma_2(\mathbf{z} - \mathbf{z}_{\Gamma_r}) \tag{3-5}$$

In general, the cost function of the error defined in the subdomain Ω_s is given in discrete points within the domain. Both finite volume and finite element solvers provide information only at the nodal points and at the center of the elements. Discrete points in the model that do not coincide with the grid must be interpolated to arbitrary points in the domain. In this work, we use an interpolation approach called Moving Least Square in the FVM and Iterative inverse mapping in the FEM (see Appendix A.3.2 for details).

3.1.2 Total pressure loss functional

The power dissipated by the fluid flow in the domain Ω can be calculated from the total pressure losses through the overall domain boundaries Γ as the following:

$$\mathcal{J} = -\int_\Gamma \left(p + \frac{1}{2}|\boldsymbol{u}|^2\right)\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma \tag{3-6}$$

where $\boldsymbol{n}$ is the unit vector normal to the boundary Γ. Then, we compute the required derivatives needed by the adjoint system, as:

$$\begin{aligned}\frac{\partial \mathcal{J}_\Gamma}{\partial \boldsymbol{u}} &= -\left[\left(p + \frac{1}{2}|\boldsymbol{u}|^2\right)\boldsymbol{n} - (\boldsymbol{u}\cdot\boldsymbol{n})\boldsymbol{u}\right],\\ \frac{\partial \mathcal{J}_\Gamma}{\partial p} &= -\boldsymbol{u}\cdot\boldsymbol{n}\end{aligned} \tag{3-7}$$

3.1.3 Average scalar functional

Cooling capacity is the main concern when designing a cooling system and can be measured by the average temperature:

$$\mathcal{J} = \frac{1}{|\Omega|}\int_\Omega \phi\,\mathrm{d}\Omega \tag{3-8}$$

where $|\Omega|$ denotes the total volume of the computational domain. Then we calculate the required derivative needed by the adjoint system as follows:

$$\frac{\partial \mathcal{J}_\Omega}{\partial \phi} = \frac{1}{|\Omega|} \tag{3-9}$$

3.1.4 Scalar flux functional

The net thermal power is evaluated by the scalar flow through the domain boundaries as follows:

$$\mathcal{J} = \int_{\Gamma} C\,\phi\,\boldsymbol{u}\cdot\boldsymbol{n}\,\mathrm{d}\Gamma \tag{3-10}$$

where $\boldsymbol{n}$ is the unit vector normal to the boundary Γ and C is a scalar transport property. Then we compute the required derivatives needed by the adjoint system as follows:

$$\frac{\partial \mathcal{J}_{\Gamma}}{\partial \phi} = \boldsymbol{u}\cdot\boldsymbol{n} \tag{3-11}$$

3.1.5 Multi-objective functional

When we consider multiple objective functions in the same problem, the weighted sum or aggregate objective function method is used to linearly combine the two objective functions. As a prerequisite for this approach, the two objective functions are normalized to avoid large differences in their numerical values corresponding to two different scales. A simple but efficient method for normalization is to divide the objective functions by their respective extreme values. The extreme value for each objective function is found independently by solving the optimization problem for two cases separately. Finally, the aggregate objective function $\mathcal{J}$ for the current problem, which is formed by the linear combination of the two normalized objective functions $\bar{\mathcal{J}}_1$ and $\bar{\mathcal{J}}_2$, can be described as follows:

$$\mathcal{J}(\mathbf{z},\varrho) = \sigma_1\bar{\mathcal{J}}_1 + \sigma_2\bar{\mathcal{J}}_2 \tag{3-12}$$

where σ_1 and σ_2 are scalar-valued weighting factors that emphasize the degree of influence or priority of each objective function ($\sigma_1 + \sigma_2 = 1$)

Another alternative is to focus on the main cost function and consider the other secondary objective functions as constraint functions.

3.2 Sensitivity of Sources

Here we are concerned with optimizing the design of the sources $\boldsymbol{f}_{\Omega_a}$ and f_{Ω_a} from Eq. (2-3). In the momentum equation, the source $\boldsymbol{f}_{\Omega_a}$ is the type of frictional force by adopting the Brinkman penalization approach as given in Eq. (2-9). In the scalar transport equation, however, the source f_{Ω_a} is

independent of the state variable ϕ, defined as $f_{\Omega_a} = Q$, where Q is a function related to the magnitude of the source.

Recall the optimality condition Eq. (2-79), from which we calculate the sensitivity of the functional $\mathcal{J}$ with respect to changes in the design parameter ϱ. After simplification, we have:

$$\delta_\varrho \mathcal{L} = \delta_\varrho \mathcal{J} - \int_\Omega \boldsymbol{v} \cdot \, \delta_\varrho \boldsymbol{f}_{\Omega_a} \, \mathrm{d}\Omega - \int_\Omega \varphi \, \delta_\varrho f_{\Omega_a} \, \mathrm{d}\Omega \tag{3-13}$$

Equation (3-13) expresses the components of the gradient of the functional with respect to the design variable ϱ in terms of the adjoints $\boldsymbol{v}$ and φ, i.e., the change in the functional $\mathcal{J}$ resulting from a change in the design variable ϱ in the direction of $\delta\varrho$. In general, the functional $\mathcal{J}$ is independent of the design variable ϱ, so the first term on the right-hand side of Eq. (3-13) is zero.

3.2.1 Source Magnitude Sensitivity

The sensitivities with respect to the parameters α and Q from Eq. (3-13) can be easily calculated by simply taking the partial derivative of each term, and result in each case in:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\alpha} = \int_{\Omega_a} \boldsymbol{v} \cdot \boldsymbol{u} \, \mathrm{d}\Omega \tag{3-14}$$

and

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}Q} = \int_{\Omega_a} \varphi \, \mathrm{d}\Omega \tag{3-15}$$

3.2.2 Source Position Sensitivity

Our goal here is to optimize the position of the sources $\boldsymbol{f}_{\Omega_a}$ and f_{Ω_a} from Eq. (2-3). Recall the optimality condition Eq. (3-13), from which we can compute the sensitivity of the functional $\mathcal{J}$ with respect to changes in the position of $\boldsymbol{f}_{\Omega_a}$ and f_{Ω_a}, so the design parameter is $\varrho = \mathbf{x}_o$. Let us define the sources with the delta functional $\boldsymbol{f}_{\Omega_a} = \alpha\delta(\mathbf{x} - \mathbf{x}_o)\boldsymbol{u}$ and $f_{\Omega_a} = Q\delta(\mathbf{x} - \mathbf{x}_o)$ as point sources. The gradient of the functional with respect to the coordinates $\mathbf{x}_o$, i.e., the derivative $\mathcal{D}\mathcal{J}/\mathcal{D}\mathbf{x}_o$ is calculated by varying $\delta\mathbf{x}_o$ and results:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}_o} = \int_\Omega \alpha \boldsymbol{v} \cdot \boldsymbol{u} \frac{\partial\delta(\mathbf{x} - \mathbf{x}_o)}{\partial \mathbf{x}_o} \, \mathrm{d}\Omega + \int_\Omega Q\varphi \frac{\partial\delta(\mathbf{x} - \mathbf{x}_o)}{\partial \mathbf{x}_o} \, \mathrm{d}\Omega \tag{3-16}$$

In both terms the partial derivative of the discontinuous function delta Dirac appears. If we resort to the following delta Dirac property (Kanwal (1998), Gross (2005)):

$$\int_{-\infty}^{\infty} f(x)\,\frac{\partial\delta(x-a)}{\partial a} = f'(a) \quad \text{for} \quad a \in \mathbb{R} \tag{3-17}$$

in our case we get the following:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}_o} = \alpha\nabla(\boldsymbol{v}(\mathbf{x}_o)\cdot\boldsymbol{u}(\mathbf{x}_o)) + Q\nabla\varphi(\mathbf{x}_o) \tag{3-18}$$

For the sensitivity of an area source, we keep the integral on the subdomain Ω_a, which gives:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}_o} = \int_{\Omega_a} \alpha\nabla(\boldsymbol{v}\cdot\boldsymbol{u})\,\mathrm{d}\Omega + \int_{\Omega_a} Q\nabla\varphi\,\mathrm{d}\Omega \tag{3-19}$$

The insight of this resulting sensitivity was explained in subsection 2.2.2 through a qualitative analysis of the global matrix resulting from the discretization of a partial differential equation.

3.3 Sensitivity of the Inlet Boundary

Here we want to optimize the design of the inlet boundary conditions $\boldsymbol{u} = \boldsymbol{g}_{\Omega_a}$ and $\phi = g_{\Omega_a}$, on Γ_a. For this purpose and for simplicity, we consider the functions $\boldsymbol{g}_{\Omega_a}$ and g_{Ω_a} as independent of the state variables $\boldsymbol{u}$ and ϕ, respectively. The adjoint variables $\boldsymbol{s}$ and s are computed from Eqs. (2-77) and (2-78).

Recall the optimality condition, Eq. (2-79), from which we calculate the sensitivity of the functional $\mathcal{J}$ with respect to changes in the design parameter ϱ. After simplification, we have:

$$\delta_\varrho\mathcal{L} = \delta_\varrho\mathcal{J} + \int_\Gamma \boldsymbol{s}\cdot\,\delta_\varrho\boldsymbol{g}_{\Omega_a}\,\mathrm{d}\Gamma + \int_\Gamma s\,\delta_\varrho g_{\Omega_a}\,\mathrm{d}\Gamma \tag{3-20}$$

Equation (3-20) expresses the components of the gradient of the functional with respect to the design variable ϱ in terms of the adjoint $\boldsymbol{s}$ and s. This equation gives the change in the functional $\mathcal{J}$ resulting from a change in the design variable ϱ in the direction of $\delta\varrho$.

Since the partial derivatives of both $\boldsymbol{g}_{\Omega_a}$ and g_{Ω_a} are relative to themselves, the sensitivity regarding these parameters results respectively,

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\boldsymbol{g}_{\Omega_a}} = \int_{\Gamma_a}\boldsymbol{s}\,\mathrm{d}\Gamma \quad \text{and} \quad \frac{\mathcal{D}\mathcal{J}}{\mathcal{D}g_{\Omega_a}} = \int_{\Gamma_a} s\,\mathrm{d}\Gamma \tag{3-21}$$

3.4 Sensitivity for Topology Optimization

The material distribution method is a common approach in topology optimization. The strategy is to assign an individual pseudo-density value $\gamma \in [0,1]$ to each element volume of the discretized domain. These pseudo-density

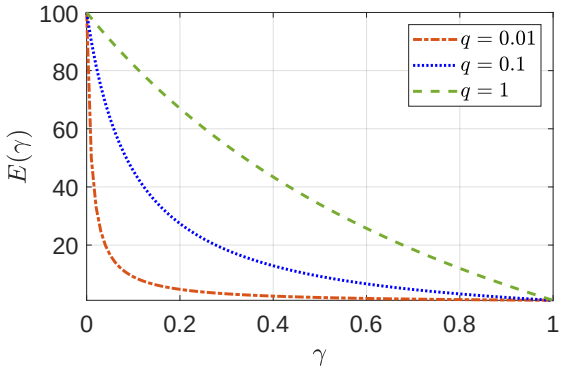

Figure 3.1: RAMP interpolation function for different values of the curvature parameter q, with $E_{max} = 100$ and $E_{min} = 1$.

values can be related to a material physical parameter E by a generic increasing monotone and continuously differentiable function of γ. For example, in the so-called "power law approach" or SIMP (Solid Isotropic Material with Penalization) (Bendsøe; Sigmund, 2004), the material properties are modeled as the pseudodensity raised to a power times the material properties. The Rational Approximation of Material Properties (RAMP)-type (Stolpe; Svanberg, 2001) is another interpolation function, which given by :

$$E(\gamma) = E_{max} + (E_{min} - E_{max})\gamma\frac{1+q}{q+\gamma}, \ \gamma \in [0,1] \tag{3-22}$$

where q is the penalty parameter that determines the form of the function $E(\gamma)$. The influence of the penalty parameter q is shown in Fig. 4.2, where higher values lead to an interpolation function with linear behavior and the distribution of γ in the domain tends to find extreme values (0 or 1). Due to nonconvexity, as noted by Borrvall; Petersson (2003), using a high value of q may lead to local optimal solution problems. Therefore, the optimization is first solved with a small value of $q = 0.01$ to obtain a better initial estimate for the problem, and then the value of the penalty parameter is increased to obtain a discrete solution.

In conjugate scalar fluid flow systems, both the friction coefficient (α) from the Brinkman penalization approach and the effective diffusion coefficient (D) from the scalar transport equation change with (γ), and in both cases, we will use the interpolation function from Eq. (3-22).

In the fluid domain, the term $\alpha\boldsymbol{u}$ goes to zero to recover the classical Navier-Stokes equation. In contrast, in the solid domain, the friction coefficient α has a very large value, so that the local velocity approaches zero. In the present work, the coefficient of friction for the fluid is $\alpha_f \approx 0$ and for the solid $\alpha_s \approx 10^5$. The equivalence with equation Eq. (3-22) would be $\alpha_s = E_{max}$

and $\alpha_f = E_{min}$. For the scalar transport equation, the diffusion coefficient is interpolated from the diffusion coefficient of the solid (D_s) to that of the fluid (D_f). The equivalence with equation Eq. (3-22) would be $D_s = E_{max}$ and $D_f = E_{min}$.

Recall the optimality condition, equation (2-79), from which we calculate the sensitivity of the functional $\mathcal{J}$ with respect to changes in the design parameter γ. Thus, we have:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\gamma} = \delta_\gamma \mathcal{J} - \int_\Omega \boldsymbol{v} \cdot \boldsymbol{u} \frac{\partial \alpha}{\partial \gamma} \delta\gamma \, \mathrm{d}\Omega - \int_\Omega \nabla\varphi \cdot \nabla\phi (D_s - D_f) \frac{\partial D}{\partial \gamma} \delta\gamma \, \mathrm{d}\Omega \tag{3-23}$$

where the partial derivatives of α and D are:

$$\frac{\partial \alpha}{\partial \gamma} = \frac{(1+q)q}{(q+\gamma)^2} \quad \text{and} \quad \frac{\partial D}{\partial \gamma} = (D_s - D_f)\frac{(1+q)q}{(q+\gamma)^2} \tag{3-24}$$

Since $\delta\gamma$ is an arbitrary function, we choose it as the Dirac delta function. Finally, we obtain the desired sensitivity for a given point $\mathbf{x}' \in \Omega$,

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\gamma}(\mathbf{x}') = \boldsymbol{v}(\mathbf{x}') \cdot \boldsymbol{u}(\mathbf{x}') \frac{\partial \alpha}{\partial \gamma} V_e + \nabla\varphi(\mathbf{x}') \cdot \nabla\phi(\mathbf{x}')(D_s - D_f) \frac{\partial D}{\partial \gamma} V_e \tag{3-25}$$

where V_e is the volume of the element corresponding to the point $\mathbf{x}'$. The equation (3-25) expresses the components of the gradient of the functional with respect to the design variable γ in terms of the adjoints $\boldsymbol{v}$ and φ. The final pseudo-density values (γ) are found by an iterative loop involving repeated evaluation of the objective function, constraints, and gradients.

3.4.1 Boundary Topology Optimization

When optimizing the boundary topology, we also use the material distribution method. The strategy is to assign an individual pseudo-density value $\gamma \in [0, 1]$ to each elemental surface of the discretized boundary. These pseudo-density values can be connected to specific values on the boundary by a generic increasing monotone and continuously differentiable function in the term of γ.

We seek that the value of the state variable $\boldsymbol{u}$ on the boundary Γ_a varies between the given values $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ as follows:

$$\boldsymbol{u} = \boldsymbol{u}_1 + \beta(\boldsymbol{u}_2 - \boldsymbol{u}_1) \quad \text{on} \quad \Gamma_a \tag{3-26}$$

This is the Dirichlet boundary condition for $\boldsymbol{u}$, which can be implemented in both FEM and FVM. The condition is that the inlet surface with a relatively

small velocity value is closed and set as the wall boundary. And the inlet surfaces with relatively large velocity values remain unchanged.

The scalar ϕ can vary between Dirichlet and homogeneous Neumann boundary conditions, and here we derive the equations to optimize the boundary topology specifically only in the context of FVM. The value of the state variable ϕ on the boundary Γ_a, varies between the values ϕ_i and ϕ_1 as:

$$\phi = \phi_1 + \beta(\phi_i - \phi_1) \quad \text{on} \quad \Gamma_a \tag{3-27}$$

where β is the interpolation function:

$$\beta = \gamma \frac{1+q}{1+\gamma} \tag{3-28}$$

and ϕ_i is the inner value of the state variable ϕ on the boundary cell. In discrete form, we can represent the gradient in a given boundary cell as follows:

$$\frac{\phi - \phi_i}{\Delta n_{BC}} = \boldsymbol{n} \cdot \nabla\phi \rightarrow \phi_i = \phi - \boldsymbol{n} \cdot \nabla\phi \Delta n_{BC} \tag{3-29}$$

where Δn_{BC} is the distance between ϕ and ϕ_i.

For the scalar, a homogeneous Neumann boundary condition is required for the wall and a Dirichlet boundary condition for the inlet surfaces with relatively large velocity values.

The adjoint variable $\boldsymbol{s}$ is computed from Eqs. (2-77). However, the computation of s requires some elaboration since the boundary condition depends on the scalar variable ϕ. Thus, the last term of the equation (2-43) changes by:

$$\begin{aligned} &\cdots - \int_{\Gamma_a} s\left[\phi - \phi_1 - \beta(\phi - \boldsymbol{n} \cdot \nabla\phi \Delta n_{BC} - \phi_1)\right] \mathrm{d}\Gamma \\ &\cdots - \int_{\Gamma_a} s\left[(\phi - \phi_1)(1-\beta) + \beta \boldsymbol{n} \cdot \nabla\phi \Delta n_{BC}\right] \mathrm{d}\Gamma \end{aligned} \tag{3-30}$$

then by the variation $\delta\phi$ from Eq. (2-57) we get:

$$\begin{aligned} &\int_{\Gamma_a} \varphi \boldsymbol{n} \cdot D\nabla\delta\phi \,\mathrm{d}\Gamma - \int_{\Gamma_a} (\boldsymbol{n} \cdot D\nabla\varphi + \boldsymbol{u} \cdot \boldsymbol{n}\varphi)\delta\phi \,\mathrm{d}\Gamma \\ &- \int_{\Gamma} s\left[\delta\phi(1-\beta) + \beta \boldsymbol{n} \cdot \nabla\delta\phi \Delta n_{BC}\right] \mathrm{d}\Gamma = 0 \end{aligned} \tag{3-31}$$

finally, if we combine the terms associated with $\delta\phi$ and $\boldsymbol{n} \cdot \nabla\delta\phi$ and consider the inflow velocity perpendicular to the boundary, i.e. $\boldsymbol{u} \cdot \boldsymbol{n} = 0$, we obtain the adjoint variable s and the boundary condition for Γ_a.

$$-\frac{1}{(1-\beta)}(\boldsymbol{n} \cdot D\nabla\varphi) = s \tag{3-32}$$

and

$$\varphi = s\beta\Delta n_{BC}/D = \beta\varphi_i \tag{3-33}$$

Recall the optimality condition (equation (2-79)) from which we calculate the sensitivity of the functional $\mathcal{J}$ with respect to changes in the design parameter γ. Thus we have:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\gamma} = \int_{\Gamma_a} \boldsymbol{s} \cdot (\boldsymbol{u}_2 - \boldsymbol{u}_1) \frac{\partial \beta}{\partial \gamma} \delta\gamma \, \mathrm{d}\Gamma + \int_{\Gamma_a} \frac{\partial \varphi}{\partial n} \frac{\partial \phi}{\partial n} \Delta n_{BC} \frac{\partial}{\partial \gamma} \left[\frac{\beta}{(1-\beta)} \right] \delta\gamma \, \mathrm{d}\Gamma \quad (3\text{-}34)$$

Since $\delta\gamma$ is an arbitrary function, we choose it as the Dirac delta function. Finally, we obtain the desired sensitivity for a given point $\mathbf{x}' \in \gamma_a$,

$$\begin{aligned} \frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\gamma}(\mathbf{x}') =& \boldsymbol{s}(\mathbf{x}') \cdot [\boldsymbol{u}_2(\mathbf{x}') - \boldsymbol{u}_1(\mathbf{x}')] \frac{\partial \beta}{\partial \gamma} A_e \\ &+ \frac{\partial \varphi}{\partial n}(\mathbf{x}') \frac{\partial \phi}{\partial n}(\mathbf{x}') \Delta n_{BC} \frac{\partial}{\partial \gamma} \left[\frac{\beta}{(1-\beta)} \right] A_e \end{aligned} \quad (3\text{-}35)$$

where A_e is the element boundary surface on Γ_a corresponding to the point $\mathbf{x}'$. The equation (3-35) expresses the components of the gradient of the functional with respect to the design variable γ in terms of the adjoints $\boldsymbol{v}$ and φ. The final pseudo-density values (γ) are found by an iterative loop involving repeated evaluation of the objective function, constraints, and gradients.

3.4.2 Volume and Area constraints

In topology optimization, it is common to introduce a constraint on the amount of available control in order to limit the volume of the solid (or fluid) to a certain fraction of the total domain, thus avoiding trivial optimal solutions. We choose to express this constraint as an equality constraint:

$$\int_{\Omega} \gamma = V_\gamma \quad (3\text{-}36)$$

where V_γ represents the volume of the solid to be kept constant during optimization. It can be assigned directly as external data or computed from a given control ρ_0 whose volume we wish to maintain.

3.4.3 Filters

Optimization methods are likely to encounter numerical difficulties such as mesh-dependence, checkerboard patterns, and local minima (Bendsøe; Sigmund, 2004). To mitigate such problems, researchers have proposed the use of regularization techniques, see (Gunzburger, 2002) and (Bendsøe; Sigmund, 2004) for an overview. One of the most common approaches is to use density and sensitivity filtering methods to obtain mesh-independent and checkerboard solutions with length scale control.

A density filter modifies the density of an element (or a mesh cell) as a function of the densities in a given neighborhood of an element. It transforms the original densities γ_e as follows:

$$\widehat{\gamma}_e = \frac{1}{\sum_{f=1}^{N} \widehat{H}_f} \sum_{f=1}^{N} \widehat{H}_f \widehat{\gamma}_f \tag{3-37}$$

In the same way, the sensitivity filter works by modifying the sensitivities of the elements as follows:

$$\frac{\widehat{\partial \mathcal{J}}}{\partial \gamma_e} = \frac{1}{\gamma_e \sum_{f=1}^{N} \widehat{H}_f} \sum_{f=1}^{N} \widehat{H}_f \gamma_f \frac{\partial \mathcal{J}}{\partial \gamma_f} \tag{3-38}$$

The convolution operator (weight factor) $\widehat{H}_f$ is written as:

$$\widehat{H}_f = r_{min} - dist(e, f),\ \{f \in N | dist(e, f) \leq r_{min}\},\ e = 1, ..., N, \tag{3-39}$$

where the operator $dist(e, f)$ is defined as the distance between the midpoint of the element e and the midpoint of the element f. The convolution operator $\widehat{H}_f$ is zero outside the filter domain. The convolution operator decays linearly with the distance from element f.

An alternative approach is to use a filter implicitly represented by the solution of the Helmholtz PDE filter (Lazarov; Sigmund, 2011; Stolpe; Svanberg, 2001). This is efficient for large problems to prevent numerical instabilities in topology optimization problems:

$$-r^2 \nabla^2 \widehat{\gamma} + \widehat{\gamma} = \gamma \tag{3-40}$$

where γ is a continuous representation of the unfiltered design field, $\widehat{\gamma}$ denotes the filtered design field, and r denotes the filter radius. The parameter r is roughly related to r_{min} from equation (3-39) by $r = r_{min}/2\sqrt{3}$, (Lazarov; Sigmund, 2011).

To solve equation (3-40), the homogeneous Neumann boundary condition is placed on the boundary of the design domain.

$$\frac{\partial \widehat{\gamma}}{\partial \boldsymbol{n}} = 0 \tag{3-41}$$

In the same way the sensitivity is filtered by the solution of the Helmholtz equation:

$$-r^2 \nabla^2 \widehat{\mathcal{S}} + \widehat{\mathcal{S}} = \mathcal{S} \tag{3-42}$$

where $\mathcal{S}$ denotes the unfiltered sensitivity $\partial \mathcal{J}/\partial \gamma$ and $\widehat{\mathcal{S}}$ is the filtered sensitivity. The boundary condition of the equation (3-42) is also of the homogeneous Neumann type

T he Helmholtz PDE filter is advantageous when the optimization problem is solved in parallel and problems with a complex geometry are present.

4
Numerical Methods

The equations and cost functions presented above are solved by a numerical method because they are very difficult to solve analytically. The numerical solution of a partial differential equation representing a physical phenomenon consists in finding the values of the dependent variable at certain points from which its distribution over the domain of interest can be constructed.

4.1
Equation Discretization

Simulations were run on a PC with the following specifications: CPU Intel Core3 M 370 2.40GHz, RAM DDR3 2.8GB 1066MHz. The finite element formulation was implemented in Matlab R2019b using polygonal grids, and the finite volume method was implemented in the open source software package CFD OpenFOAM-v2012.

In the finite element method (FEM), we represent the domain implicitly using the signed distance function, which provides great flexibility to construct a relatively large class of domains using algebraic expressions, as shown by Persson; Strang (2004) and Talischi et al. (2012). We also use a polygonal grid with explicit topological information, i.e., connectivity and numbering of geometric features. A discretization of the domain is constructed from a Centroidal Voronoi Tessellation (CVT), which includes an approximation of the boundaries. Then, the approximation is obtained by including the reflections of the seeds. Finally, the Lloyd method is used to create a uniform (optimal) distribution of seeds and thus a high-quality mesh with convex polygons. We use shape functions for polygonal elements proposed by Wachspress (1975). This numerical formulation allows the construction of conformal approximations on n vertices ($n \geq 3$), thus extending the applications of finite elements to convex polygons of arbitrary order. Details of the implementation of FE, the shape function, and the algorithms used in this work can be found in the Appendix A.

In the finite volume method (FVM), we can create the geometry domain and generate the mesh using the OpenFOAM utilities blockMesh and snappyHexMesh, see (Openfoam, 2021b) and (Openfoam, 2021a).

The FVM owes much of its flexibility and popularity to the fact that the discretization is done directly in physical space, without requiring any transformation between the physical and computational coordinate systems

(Versteeg; Malalasekera, 2007). In solving the NS /adjoint equations, this study used a semi-implicit method for a pressure-linked equation algorithm (SIMPLE) to couple the velocity and pressure (see appendix B and C for more details). For a thorough review of this approach, see the books on the subject (Golub; Loan, 2013) and (Patankar, 1967).

4.1.1 Code Verification

We are going to verify the finite element code in terms of its computational consistency and order of accuracy. Discretization methods are consistent if the error approaches zero as the representative cell size (h) decreases toward zero. The rate at which the error decreases toward zero is called the order of accuracy. In this context, we use the manufactured solutions method, where an analytical solution is chosen a priori and the governing equations are modified by adding analytical source terms.

These exact solutions are then used to accurately evaluate the discretization error in the numerical solutions. (Salari; Knupp, 2000; Roy et al., 2004; Roache, 2002) has presented a detailed discussion of manufactured solutions for code verification.

We have chosen the problem studied by Shih; Tan; Hwang (1989), which is similar to the classical lid-driven cavity flow, but for which an exact solution is known. The domain consists of a square with dimensions 1×1, as shown in Fig. 4.1.

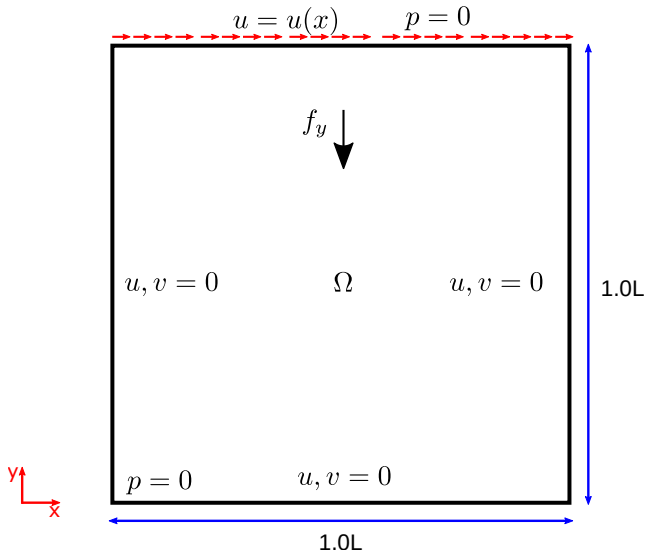

Figure 4.1: Geometry domain and boundaries of Lid-driven cavity.

The manufactured solutions with kinematic viscosity equal to 1, for velocity components and pressure are:

$$u(x,y) = 8f(x)g'(y) = 8(x^4 - 2x^3 + x^2)(4y^3 - 2y), \tag{4-1}$$

$$v(x,y) = -8f'(x)g(y) = -8(-4x^3 - 6x^2 + 2x)(y^4 - y^2), \tag{4-2}$$

$$p(x,y) = \frac{8}{\nu}[F(x)g'''(y) + f'(x)g'(y)] + 64F_2(x)\{g(y)g''(y) - [g'(y)]^2\} \tag{4-3}$$

A body force generated for the manufactured solution produced is present in the y-direction and is prescribed as:

$$\begin{aligned} \boldsymbol{f}_y(x,y) = &-\frac{8}{\nu}[24F(x) + 2f'(x)g''(y) + f'''(x)g(y)] \\ &- 64[F_2(x)G_1(y) - g(y)g'(y)F_1(x)] \end{aligned} \tag{4-4}$$

where

$$f(x) = x^4 - 2x^3 + x^2, \tag{4-5}$$

$$g(x) = y^4 - y^2, \tag{4-6}$$

$$F(x) = \int f(x)dx = 0.2x^5 - 0.5x^4 + x^3/3, \tag{4-7}$$

$$F_1(x) = f(x)f''(x) - [f'(x)]^2 = -4x^6 + 12x^5 - 14x^4 + 8x^3 - 2x^2, \tag{4-8}$$

$$F_2(x) = \int f(x)f'(x)dx = 0.5[f(x)]^2, \tag{4-9}$$

$$G_1(y) = g(y)g'''(y) - g'(y)g''(y) = -24y^5 + 8y^3 - 4y \tag{4-10}$$

The boundary conditions for the velocity along the upper surface are $u(x,1) = 16(x^4 - 2x^3 + x^2)$ and equal to zero everywhere. We calculate the discretization error (E_{grid^h}) using the L_2 norm.

$$E_{grid^h} = \left\| \boldsymbol{u}^{ex} - \boldsymbol{u}^h \right\|_2 = \left(\int_\Omega (\boldsymbol{u}^{ex} - \boldsymbol{u}^h)^2 d\Omega \right)^{\frac{1}{2}} \tag{4-11}$$

The systematic grid refinement with a constant grid refinement ratio (r) yields a sequence of observed orders of accuracy ($\hat{p}$) calculated as follows:

$$\hat{p} \approx log\left(\frac{E_{grid^h}}{E_{grid^{h/2}}} \right) / log(r) \tag{4-12}$$

For polygonal meshes, we consider the characteristic size of the element (h) as the square root of the average area of the element.

Table 4.1 shows the calculated errors based on grid size and the observed accuracy for all calculated variables. Figures 4 and 5 show the solution of the velocity components and pressure and their corresponding error distributions. The tabulated results show second order behavior for velocity and first order behavior for pressure.

Table 4.1: Numerical errors and orders-of-accuracy of the FE solutions.

Grid size (h)	$L_2 - norm\ \boldsymbol{u}$	$L_2 - norm\ p$	Order $\boldsymbol{u}$	Order p
1/5	4.01e-02	6.75e-01	-	-
1/10	8.86e-03	2.72e-01	2.07	1.24
1/20	2.43e-03	1.36e-01	1.82	0.97
1/40	6.23e-04	6.59e-02	1.94	1.03
1/80	1.51e-04	3.26e-02	2.03	1.01

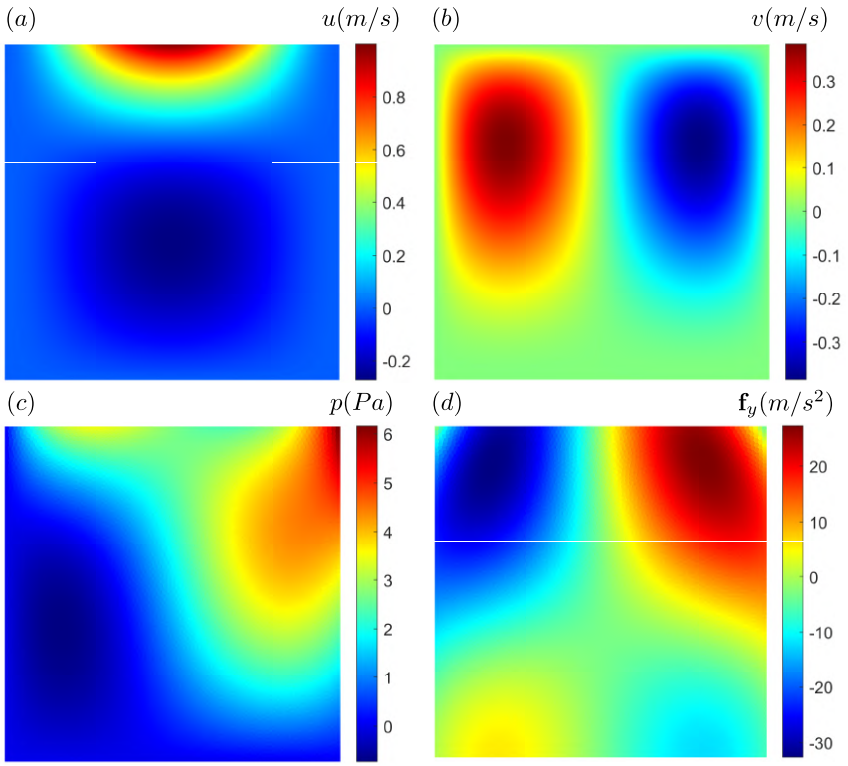

Figure 4.2: Manufactured solutions for the incompressible Navier-Stokes equations. Velocity components and pressure are shown.

4.1.2 Model Validation

The flow over a backward-facing step with a sudden expansion of 1:2 is considered. Details of the geometry and mesh are given in Fig. 4.3. It is a prototype for separating, circulating, and reconnecting flows in nature and in numerous engineering applications. The case has already been studied by several researchers and the data has been obtained not only experimentally (Lee; Mateescu, 1998) (Armaly et al., 1983), but also numerically by direct numerical simulation (DNS) (Kopera et al., 2014; Le; Moin; Kim, 1997).

In the inlet, the fully developed parabolic velocity profile for a laminar flow is considered as follows:

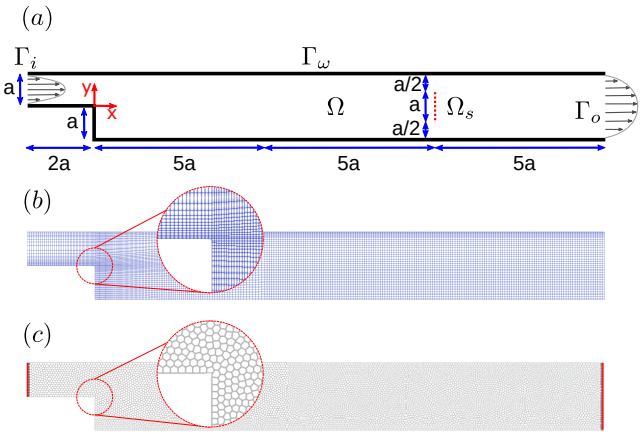

Figure 4.3: Backward-facing step domain. (a) Geometry dimensions and boundaries, with $a = 0.02m$. (b) Structured mesh for FVM. (c) Polygonal mesh for FEM.

$$U(y) = U_{max}\left[1 - \left(\frac{y}{a/2}\right)^2\right] \tag{4-13}$$

where a is the internal distance in the inlet and U_{max} is the maximum velocity. The average velocity for this case is:

$$U_{avg} = \frac{2}{3}U_{max} \tag{4-14}$$

and the Reynolds Number results:

$$Re = \frac{U_{avg}2a}{\nu} \tag{4-15}$$

First, we performed simulations in the laminar regime at Reynolds numbers of 20, 100, and 200, using both FEM and FVM, respectively. The velocity fields with FVM are shown in Fig. 4.4. The simulations were performed after a mesh convergence test. Fig. 4.5, shows the primal velocity profile $\boldsymbol{u}$ at $Re = 200$, for different numbers of cells. In this way, we decided to use 44580 cells.

The code FE has scale limitations, so a higher number of elements leads to numerical instability. Therefore, the simulation in this case was performed with a maximum of 7000 polygonal elements. A comparison between the solutions $\boldsymbol{u}_{FVM}$ and $\boldsymbol{u}_{FEM}$ obtained with FEM and FVM, respectively, was performed in the L^2 norm, calculating the relative error:

$$\varepsilon_{\boldsymbol{u}} = \frac{\|\boldsymbol{u}_{FEM} - \boldsymbol{u}_{FVM}\|_{L^2(\Omega)}}{\|\boldsymbol{u}_{FVM}\|_{L^2(\Omega)}} \tag{4-16}$$

Figure 4.6b shows that the relative error increases with increasing Reynolds number. The profiles of $\boldsymbol{u}$ along the x direction at $y = 0.01m$ in Fig. 4.6a illustrate the difference in the solutions. Although we use a coarse mesh in the FEM, the relative error is small compared to the FVM. Since the adjoint

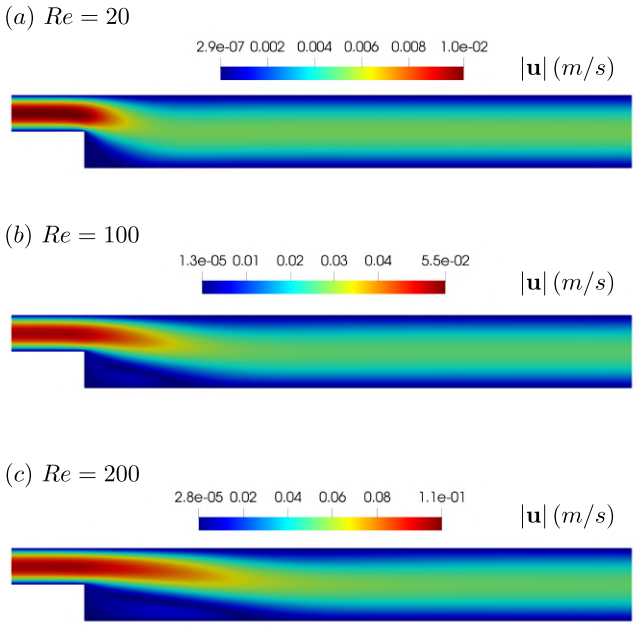

Figure 4.4: Velocity field at Reynolds number (a) 20, (b) 100 and (c) 200.

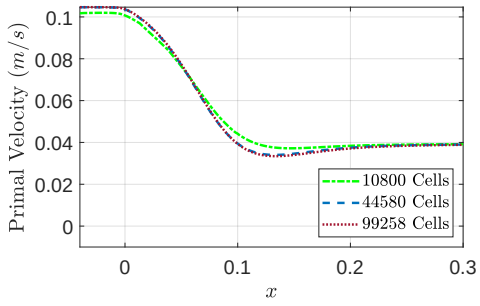

Figure 4.5: Primal velocity profile $\boldsymbol{u}$ at $Re = 200$ along x direction at $y = 0.01$, for different number of cells.

velocity calculation requires a primal velocity as a reference, in Chapter 5 we will use the primal velocity obtained from the FVM for both the continuous and discrete methods to compare the two methods.

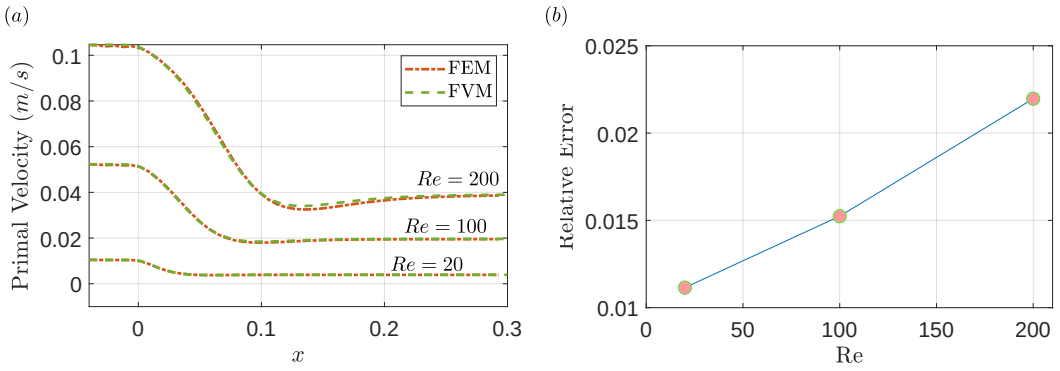

Figure 4.6: Solution comparison between FEM and FVM at different Reynolds numbers. (a) Velocity profile and (b) relative error.

The simulation results for the reattachment length X_r as a function of Re are shown together with the experimental results of (Armaly et al., 1983) in Fig. (4.7). The location of the main reattachment length was found where the shear stress changes sign. At $Re = 600$, the numerical results begin to diverge from the experimental measurements. As (Armaly et al., 1983) noted, the experimental flow becomes three-dimensional at this Reynolds number. Thus, we see that in the range of Reynolds numbers from 600-800, the present results give a much larger reattachment length than the experimental results.

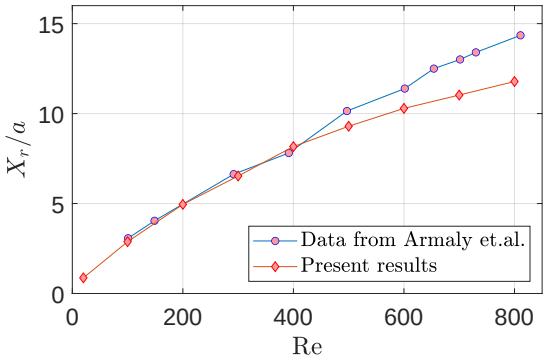

Figure 4.7: Reattachment length for a range of Reynolds numbers in the laminar regime.

5
Sensitivity Analysis

When we resort to conventional methods, e.g., method based on finite difference, the number of simulations required for gradient calculations increases dramatically with the number of design parameters, and this usually results in only a limited number of design parameters being considered. However, when using the adjoint method, the gradient calculations can be performed virtually independent of the number of design parameters, resulting in a more flexible and robust optimization tool.

5.1
Verification and Analysis of Adjoint Variables

In chapter 3 we note that sensitivities are directly related to adjoint variables. This means that if we solve the adjoint equations correctly and get the exact adjoint variables, we can confidently use them to calculate the sensitivities.

Let us also remember that with FEM we can perform the CA and DA methods, at least in the laminar regime. With FVM, on the other hand, we can only use the CA method. So, when we compare the methods using the field profiles, CA-FEM indicates that the results correspond to the simulations with the continuous adjoint in the framework of the finite element method, and so on.

The problem with the CA method are the correct implementation of the adjoint boundary conditions and the solution method used. The implementation of the boundary conditions in FEM and FVM is quite different, see appendix A and B. The system of adjoint equations is linear, it can be solved either by a direct method with the FEM, i.e. $Ax = B$, where $x = A^{-1}B$, or in an iterative way by the SIMPLE method with the FVM. Using FEM we can access the Jacobian matrix and thus make the DA formulation and obtain the exact adjoint variables. However, the FE -solver implemented in this work limits us to the simulation of laminar flows (low Reynolds numbers). With the FVM, on the other hand, we can simulate both laminar and turbulent flow conditions, but we are forced to use only CA formulations since we do not have access to the Jacobian matrix. The equations of CA are not conservative; solving them with the Semi-Implicit Method for Pressure-Linked Equations algorithm (SIMPLE) could lead to convergence problems and local residual errors when the mesh is not sufficiently refined and the global matrix solution

method is not suitable.

In this context, we will compare the continuous and discrete adjoint methods presented in chapter 3 for different objective functions and correct possible errors in the CA formulation. For this purpose, we will consider the case of flow over a backward step from chapter 4 .

For the adjoint proof, we work with $Re = 200$ in the following. Computing the adjoint variables by either discrete or continuous methods considered in sections 2.2.2 and 2.2.3 show that we need to specify the form of the cost function. First, we use the cost function of error presented in section 3.1 with reference velocity $\boldsymbol{u}_{\Omega_r} = (0.05, 0.0)m/s$ in the whole domain, which corresponds to a value close to the inlet velocity of the fluid. We chose this objective function acting in the domain with the idea of analyzing the values of the adjoint variables in the inlet and outlet contours, thus verifying the implementation of the continuous adjoint method with FVM.

$$\mathcal{J} = \frac{1}{2}\int_{\Omega}(\boldsymbol{u} - \boldsymbol{u}_{\Omega_r})^2 \, \mathrm{d}\Omega \tag{5-1}$$

The adjoint velocity and adjoint pressure fields using FVM are shown in Fig. 5.1.

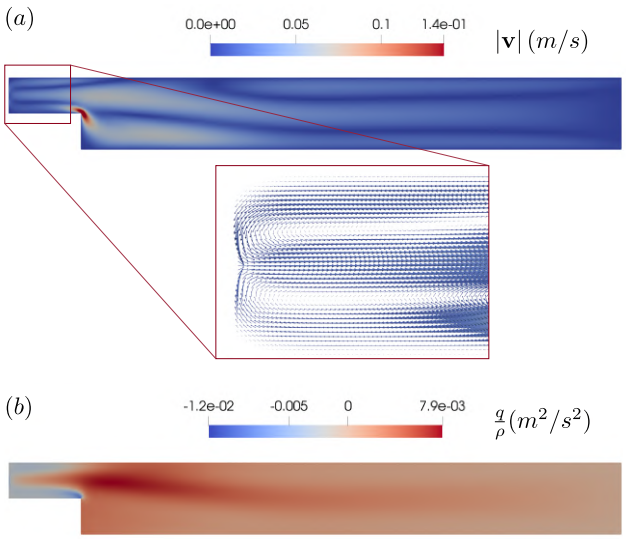

Figure 5.1: (a) Adjoint velocity and (b) adjoint pressure fields, for the cost function Eq.(5-1).

Figures 5.2a and 5.2b show us that the adjoint velocity and pressure profiles along the x direction at $y = 0.01m$ give essentially the same results for the continuous and discrete methods. However, if the mesh is not appropriately refined and the solver is not appropriately tuned, local convergence errors can occur, as shown by the green line. Note that the error occurred near the inlet boundary, which in the case of the adjoint method is a wall-like boundary,

i.e., the adjoint velocity is zero at Γ_i. This leads to a strong gradient and a recirculating behavior in this zone, as shown in Fig. 5.1a is highlighted. Therefore, this zone requires mesh refinement to adequately solve the fields.

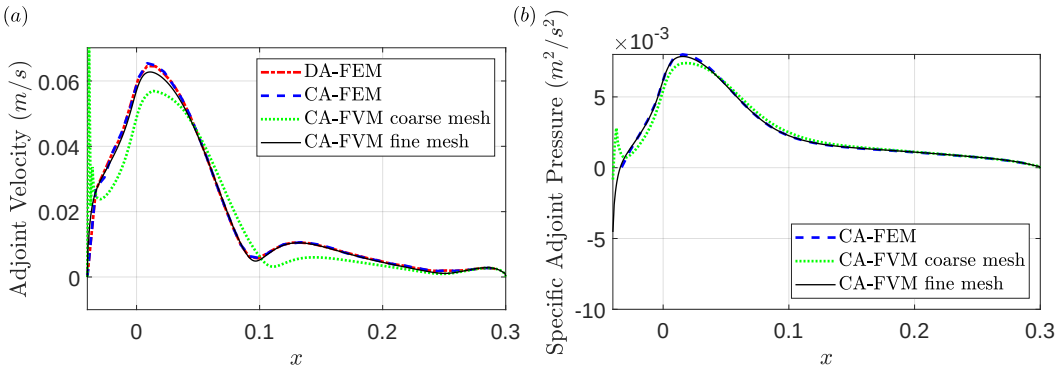

Figure 5.2: (a) Adjoint velocity and (b) adjoint pressure profiles, along x direction at $y = 0.01m$, for cost function Eq.(5-1).

Now we add the equation of transport of energy. For simplicity in Eq. (2-3) we replace the scalar ϕ for the temperature variable T, and its counterpart adjoint variable φ which appears in Eq. (2-54) for the adjoint temperature T_a. The fluid enters the domain at Γ_i with a constant temperature of $300K$; the third part of the lower wall is kept at $310K$ to create a temperature gradient, and the other walls are adiabatic boundaries. In this case, the cost function of error has the form:

$$\mathcal{J} = \frac{1}{2}\int_{\Omega}(\boldsymbol{u} - \boldsymbol{u}_{\Omega_r})^2 \,\mathrm{d}\Omega + \frac{1}{2}\int_{\Omega}(T - T_{\Omega_r})^2 \,\mathrm{d}\Omega \tag{5-2}$$

where the reference temperature $T_{\Omega_r} = 302K$ is considered in the whole the domain. Figure 5.3 shows the adjoint velocity, primal temperature, and adjoint temperature fields, respectively.

If we include the energy transport equation, we find that the magnitude of the adjoint velocity increases compared to the previous case, where the maximum value of the magnitude of the adjoint velocity was $0.14m/s$, and here the maximum value of the magnitude of the adjoint velocity is $680m/s$. This happens because an additional term $\varphi\nabla\phi$ (in this case for energy equation $T_a\nabla T$) appears in the adjoint momentum equation (Eq. (2-52)) as a source term. Again, we have a strong gradient at the entrance of the domain. Although the mesh is refined, the adjoint variables solved with FVM have local errors in this region, as shown in Figs. 5.4a and 5.4b. Therefore, the sensitivities in this region should be corrected and handled carefully.

Since we want to control some regions in the domain, we use the cost function of error in a particular region of the domain. Recall from Fig. 4.3

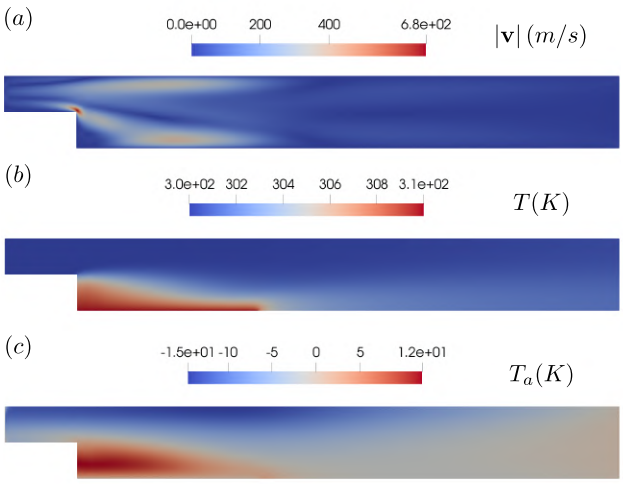

Figure 5.3: (a) Adjoint velocity, (b) primal temperature and (c) adjoint temperature fields, for the cost function Eq.(5-2).

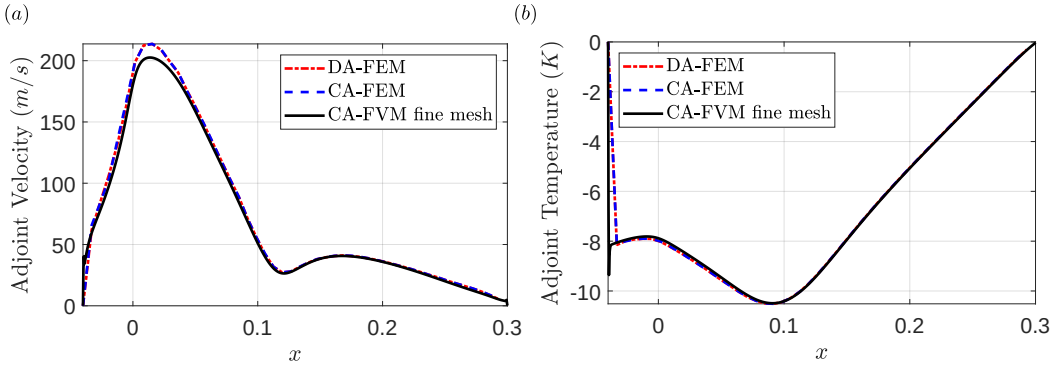

Figure 5.4: (a) Adjoint velocity and (b) adjoint temperature profile, along x direction at $y = 0.01m$, for cost function Eq.(5-2).

in chapter 2 where we defined Ω_s. In this case, Ω_s is a line consisting of 20 consecutive uniformly spaced points at $x = 0.2m$, $y \in [-0.01m, 0.01m]$. The objective function has the form of Eq. (5-3) and we call it the line error cost function. We use interpolation methods (see Appendix A.3) to represent this type of cost function.

$$\mathcal{J} = \frac{1}{2}\int_{\Omega_s} (\boldsymbol{u} - \boldsymbol{u}_{\Omega_r})^2 \, \mathrm{d}\Omega + \frac{1}{2}\int_{\Omega_s} (T - T_{\Omega_r})^2 \, \mathrm{d}\Omega \tag{5-3}$$

The adjoint velocity and temperature fields using the CA-FVM are shown in Fig. 5.5 and the profiles along the x direction at $y = 0.01m$, using different methods is presented in Fig. 5.6. Again, the error is present in the inlet due to the steep gradient in this zone. Although we have used interpolation methods to draw the derivative of the cost function, moving least square in the FVM and iterative inverse mapping in the FEM, there is a slight difference in their profiles. The line error cost function acting in Ω_s has provided higher values of the adjoint velocity and temperature than in the previous case. Moreover, the

adjoint velocity and temperature fields are more complex. The sensitivity on the right side of Ω_s, i.e., where the cost function was defined, decreases rapidly for the adjoint velocity and is practically zero for the adjoint temperature, as expected, because the flow direction is from left to right. Any perturbation in this region has little or no effect on the cost function.

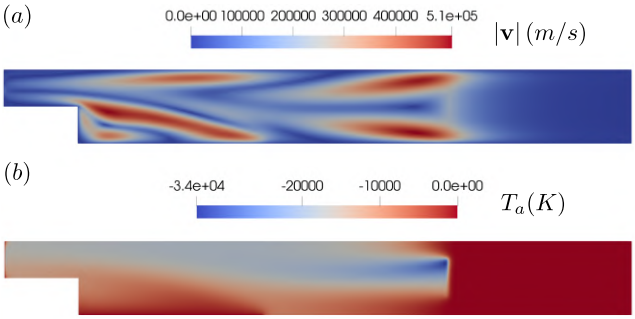

Figure 5.5: (a) Adjoint velocity and (b) adjoint temperature fields, for cost function Eq.(5-3).

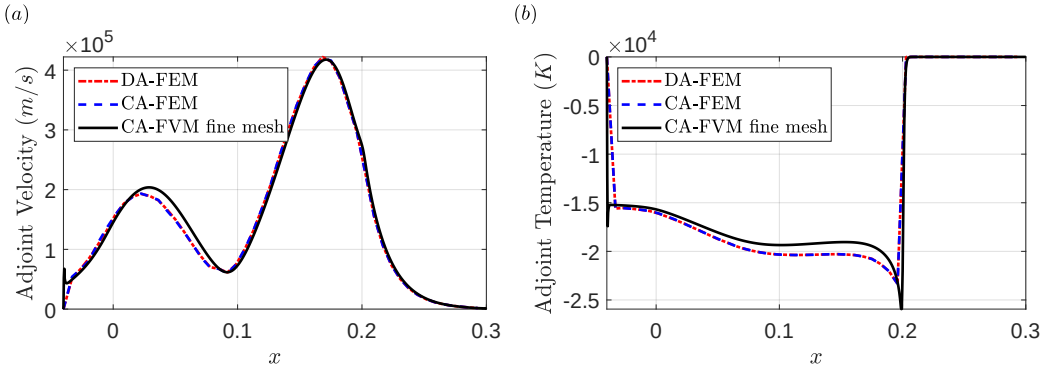

Figure 5.6: (a) Adjoint velocity and (b) adjoint temperature profiles, along x direction at $y = 0.01m$, for cost function Eq.(5-3).

All these adjoint fields show us regions with strong positive and negative output sensitivities. Any residual perturbation made in these regions would therefore have a large effect on the final objective function, while perturbations in regions where the adjoint is close to zero would have a small effect on the objective function. Moreover, a general representative feature of adjoint problems is that the information in the adjoint problem flows in the opposite direction as the information in the original ("primal") problem, as shown in Fig. 5.7a shown.

We note that the adjoint values are independent of the choice of design parameters (ϱ). Moreover, we might have expected more or less random "spikes" in sensitivity when plotting their contours. However, since perturbations in the residuals propagate by physical mechanisms (e.g., acoustic waves or convection

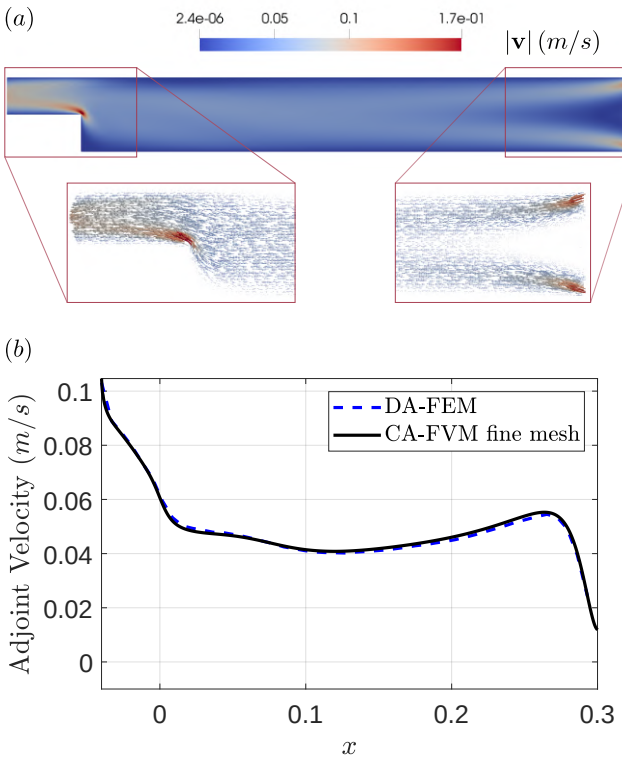

Figure 5.7: (a) Adjoint velocity field and (b) adjoint velocity profile, along x direction at $y = 0.01m$, for total pressure loss cost function.

motion) and most physical propagation mechanisms are inherently "smooth," this smoothness is reflected in the adjoint.

If you plot the adjoint velocity and adjoint temperature profiles along the x direction at $y = 0.01$, as shown in Figs. 5.2, 5.4, 5.6, and 5.7b.

We note that the solutions from FEM and FVM are essentially identical. So we can confidently move on to the next step, which is to determine the sensitivities.

5.1.1 No Convergence of Adjoint Equations by FVM in the Cavity Case

We consider flow in a square cavity with an inlet and an outlet. We consider the transport of energy. The fluid enters the cavity at Γ_i with a parabolic velocity profile and a constant temperature of $305K$; the top and bottom walls are kept at $300K$ and $310K$, respectively; the other walls are adiabatic boundaries. Details of the geometry and mesh are shown in Fig. 5.8.

The contours of the velocity and temperature fields with FVM at Reynolds numbers 20 and 100 are shown in Fig. 5.9. Since we solve the governing equations iteratively, we must guarantee convergent solutions. In these cases, for example, about 3000 iterations were required until the given tolerance of 10^{-9} in the momentum, continuity, and energy residuals was reached, as shown

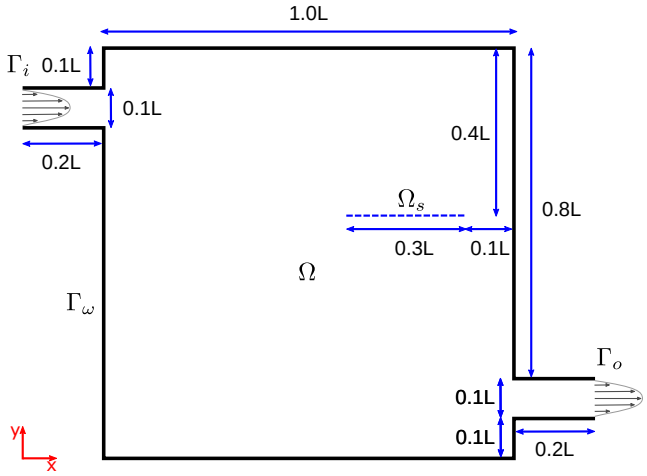

Figure 5.8: Sketch of geometry, dimensions and boundaries of a cavity domain with L=1m.

in Fig.5.10.

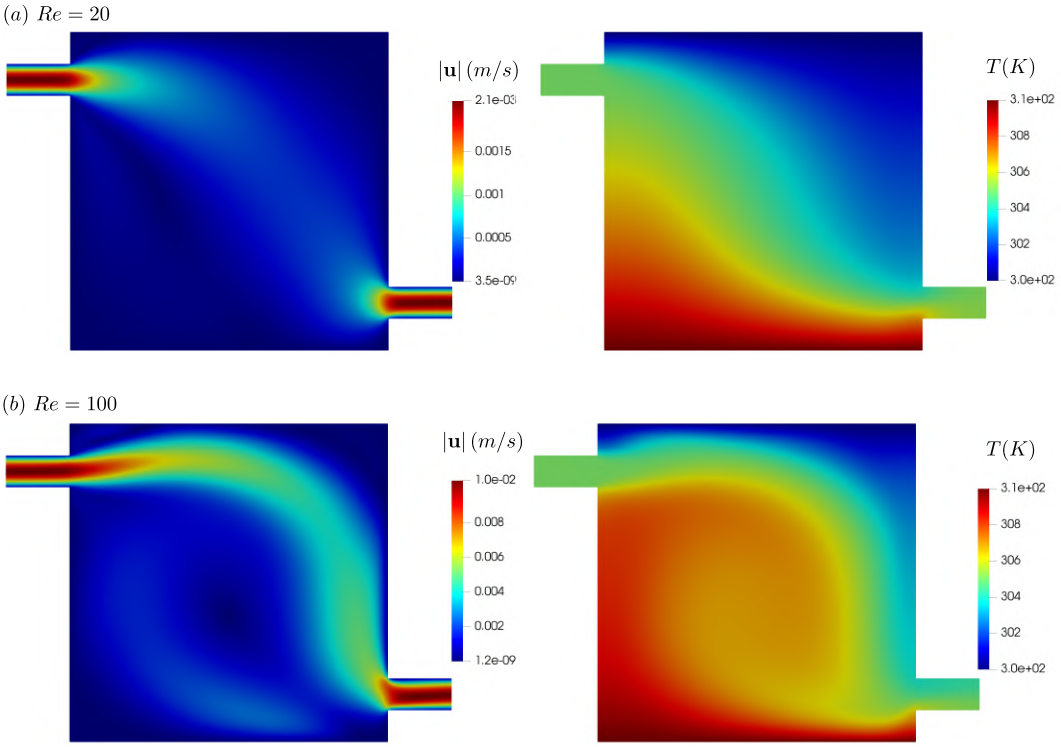

Figure 5.9: Contour of velocity and temperature fields at Reynolds number (a) 20 and (b) 100.

To compute the adjoint variables, we use the line error cost function Eq. (5-3) in a given region of the domain, i.e., on Ω_s, with $\boldsymbol{u}_{\Omega_r} = (0, -0.005, 0)m/s$ and $T_{\Omega_r} = 302K$, as outlined in Fig.5.8a. This is the case when we want to control some regions in the domain; therefore, we need to place sensors in these regions, i.e., define the cost function on Ω_s.

If we use the SIMPLE algorithm to solve the CA equations, we see in

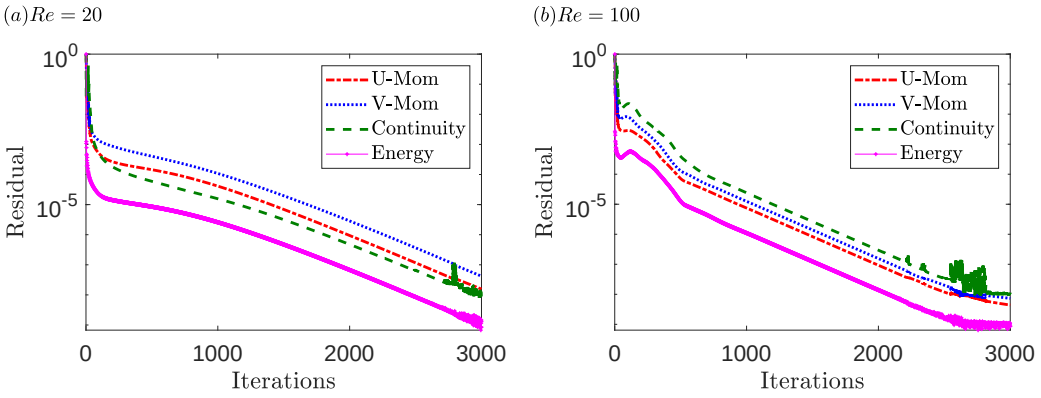

Figure 5.10: Residuals at Reynolds number (a) 20 and (b) 100, for cavity case.

Fig. 5.11 that the residuals for the adjoint velocity and adjoint pressure do not converge. Figures 5.12a, 5.12b and 5.12c show the adjoint velocity fields at iteration number 200, 500 and 1000, respectively. It is obvious that the final result (iteration 1000) is not correct when compared to the result obtained with DA-FEM and shown in Fig. 5.12d.

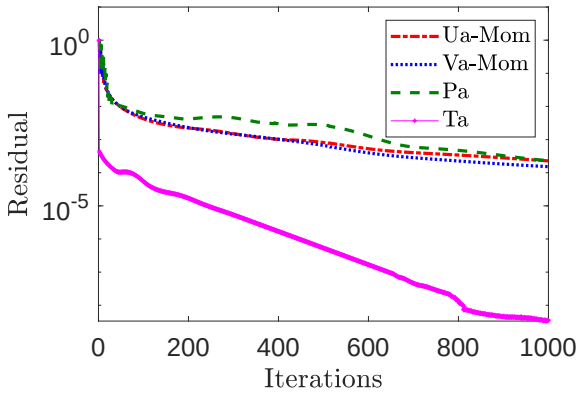

Figure 5.11: Adjoint residuals at $Re = 100$, for cavity case.

We recall that in the adjoint equations (2-52), (2-53) and (2-54) new terms appear, acting as sources. Moreover, the associated boundary conditions also change, e.g., some terms at the outlet boundary appear as partial derivatives of the objective function with respect to the primal state variables. Similarly, the inlet boundary condition, which was defined as a Dirichlet boundary in the primal equation, becomes homogeneous. In this way, the fact that the adjoint equations are not derived from the flux balance does not make the FVM a suitable method, since it aims precisely to satisfy the flux balance in any finite volume. However, the adjoint system of equations is linear and it is not necessary to use the SIMPLE algorithm to solve the couple velocity-pressure . We can do this with the FEM. First we need to consider that the adjoint

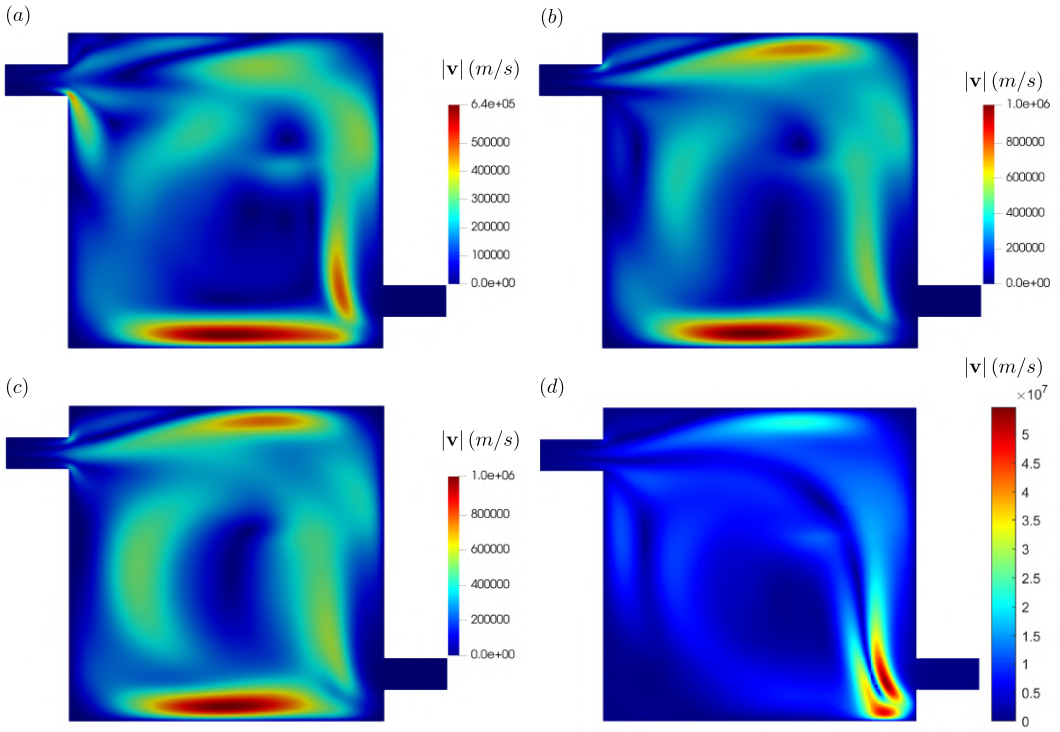

Figure 5.12: Adjoint velocity field using FVM at number of iteration (a) 200, (b) 500 and (c) 1000. Correct adjoint velocity field using FEM.

equations depend on the state variables (velocity and temperature) computed with FVM in the OpenFOAM environment using a structured mesh. Therefore, it is necessary to map the primal values of velocity and temperature at each node of the polygon mesh to solve the adjoints by the FEM. For this purpose, we use the methods of linear interpolation and nearest neighbor interpolation.

In this work we didn't couple the FEM and FVM. So, the suggestion for future work is to use FVM and FEM together to solve the primal and adjoint equations, respectively. Then use the results to calculate the sensitivities. For this purpose, we could either combine Matlab and OpenFOAM or integrate a FEM code into the OpenFOAM environment.

5.2 Validation of sensitivities

The sensitivities of the design variables (ϱ), e.g., sources, inlet boundaries, material distribution, are closely related to the adjoint values and were presented in chapter 3. To investigate the validity of these sensitivities, a comparison was made with sensitivities calculated by the finite difference (FD) method.

5.2.1 Sensitivity of Sources

First, we consider the sensitivity of the sources, both in terms of size and position. We begin by expressing the problem in the 1D domain $\Omega(x) = (0, L) \in \mathbb{R}$ of length $L = 1m$ and we define ϕ as a dimensionless scalar. The primal equation is a linear second order ordinary differential equation,

$$u\frac{\partial \phi}{\partial x} - D\frac{\partial^2 \phi}{\partial x^2} - f_{\Omega_a} = 0 \quad \text{in} \quad \Omega, \tag{5-4}$$

with Dirichlet and Neumann boundary conditions respectively,

$$\phi|_{\Gamma_i} = g_D, \quad \text{and} \quad n_x D\frac{\partial \phi}{\partial x}\bigg|_{\Gamma_o} = 0, \tag{5-5}$$

The objective function and derivatives in this case have the form:

$$\mathcal{J} = \frac{1}{2}\int_{\Omega_s} (\phi - \phi_{\Omega_r})^2 \, \mathrm{d}x, \quad \frac{\partial \mathcal{J}_\Gamma}{\partial \phi} = 0 \quad \text{and} \quad \frac{\partial \mathcal{J}_{\Omega_s}}{\partial \phi} = (\phi - \phi_{\Omega_r}) \tag{5-6}$$

where ϕ_{Ω_r} is a scalar field computed with reference parameters that define f_{Ω_a}, and g_D is a particular value for the Dirichlet boundary condition. In the following cases, we conveniently consider $u = 0.01m/s$ and $D = 0.005m^2/s$.

For the 1D cases, we consider "inverse design optimization", i.e., starting from an initial state variable ϕ_o given by f_{Ω_o}, we use the Quasi-Newton BFGS optimization algorithm to get as close as possible to a reference state variable ϕ_{Ω_r} obtained from a fixed f_{Ω_r}. The final result is the optimized state variable ϕ_{opt}, which corresponds to $f_{\Omega_{opt}}$.

Figure 5.13 shows the case of a non-uniformly distributed source. The sensitivity of the cost function ($\mathcal{J}$) with respect to the size of each source is proportional to the value of the adjoint at each position, as shown in Fig. 5.13d. In Fig. 5.13a, we note that f_{Ω_r} and $f_{\Omega_{opt}}$ do not completely match. This is because we may have found a local minimum since we are optimizing using the gradient method. Although there is a significant difference between f_{Ω_o} and f_{Ω_r}, the difference between the scalar fields ϕ_o and ϕ_r is relatively small.

Figure 5.14 shows the case of two-point sources and their position optimization. A look at Figs. 5.14d and 5.14f confirms that the sensitivity of $\mathcal{J}$ from Eq. (5-6) with respect to the position of each source is proportional to the value of the gradient of the adjoint variable at each position of the source.

In the contour representation of Fig. 5.14e we can see that there are two minima of the objective function relative to the position of the sources. The minimum corresponding to f_{Ω_r} is denoted by P_{opt}. Since we are using the gradient method, whether we fall into one of the minima depends on our

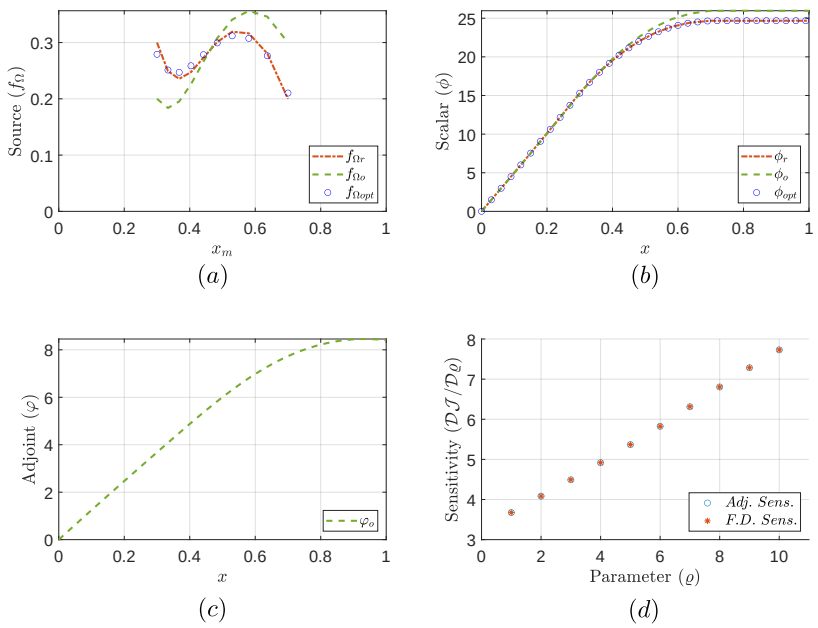

Figure 5.13: Distributed source magnitude optimization for 1D case (a) source, (b) scalar and (c) adjoint profiles. (d) Sensitivity of each source.

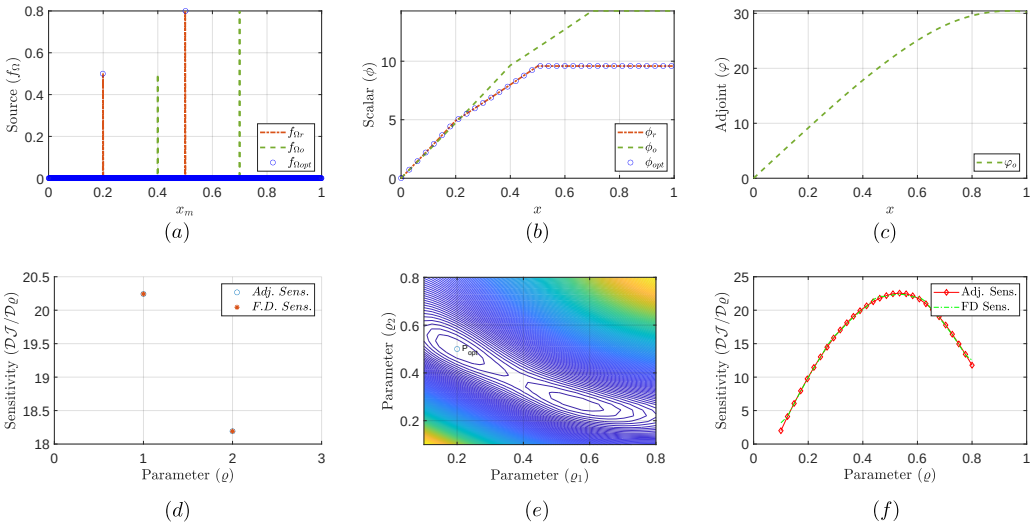

Figure 5.14: Two point source position optimization for 1D case (a) source, (b) scalar and (c) adjoint profiles. (d) Sensitivity of each source at initial state. (e) Contour of objective function relative to the coordinates of the two sources. (f) Sensitivity for a range of coordinate values of source 1.

choice of initial parameters. In view of this, it is advisable to acquire a good knowledge of the phenomena in order to determine the objective function and the value of the initial parameters.

Figure 5.15 shows the case of two area sources formed by assuming a Heaveside function. Here, the sensitivity of the objective function with respect to the central position of each source is proportional to the integral of the adjoint spanning the entire size of each source. Then the case is similar to the

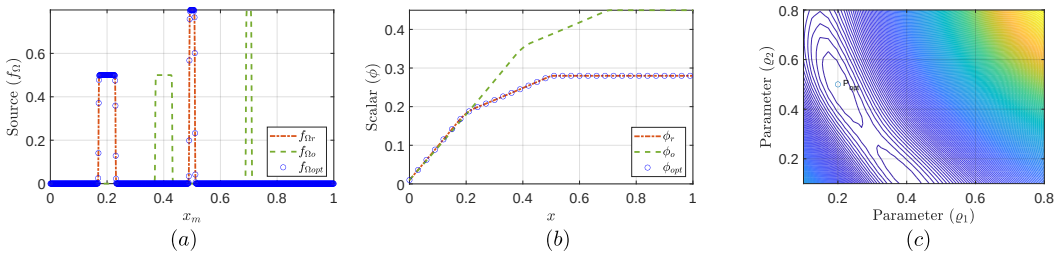

Figure 5.15: Two area source position optimization for 1D case (a) source and (b) scalar profiles. (c) Contour of objective function relative to the coordinates of the two sources.

point sources in the previous case.

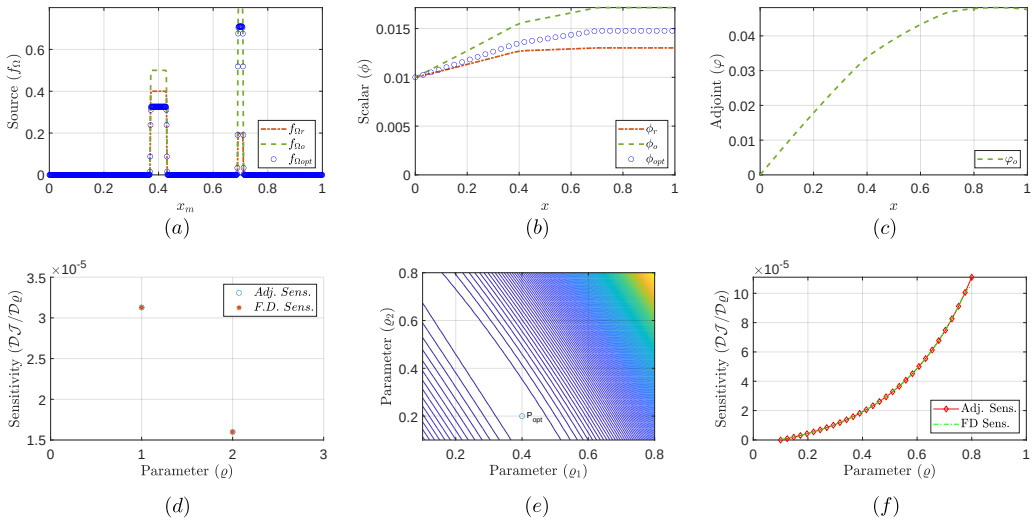

Figure 5.16: Two area source magnitude optimization for 1D case (a) source, (b) scalar and (c) adjoint profiles. (d) Sensitivity of each source at initial state. (e) Contour of objective function relative to the magnitude of the two sources (f) Sensitivity for a range of magnitude values of source 1.

Figures 5.16 and 5.17 show the cases for optimizing the size and position of the area sources when f_{Ω_a} depends on ϕ, i.e., $f_{\Omega_a} = Q\phi$. The sensitivities are given by the following expressions,

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}Q} = \int_{\Omega_a} \varphi\phi \, \mathrm{d}\Omega, \tag{5-7}$$

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}_o} = \int_{\Omega_a} Q\nabla(\varphi\phi) \, \mathrm{d}\Omega \tag{5-8}$$

In both cases, the sensitivities calculated using the adjoint and FD methods are essentially identical. However, when using the Quasi-Newton BFGS optimization algorithm, we are not able to reach the minimum because f_{Ω_r} and $f_{\Omega_{opt}}$ are as wide as shown in Figs. 5.16a and 5.17a, respectively. Figures 5.16e and 5.17e show the contour plot of the objective function with respect to the size and position of the sources, respectively. We note that the

global minimum P_{opt} is in a very flat zone in both cases, so the algorithm fails or has difficulty reaching the global minimum using the gradient method.

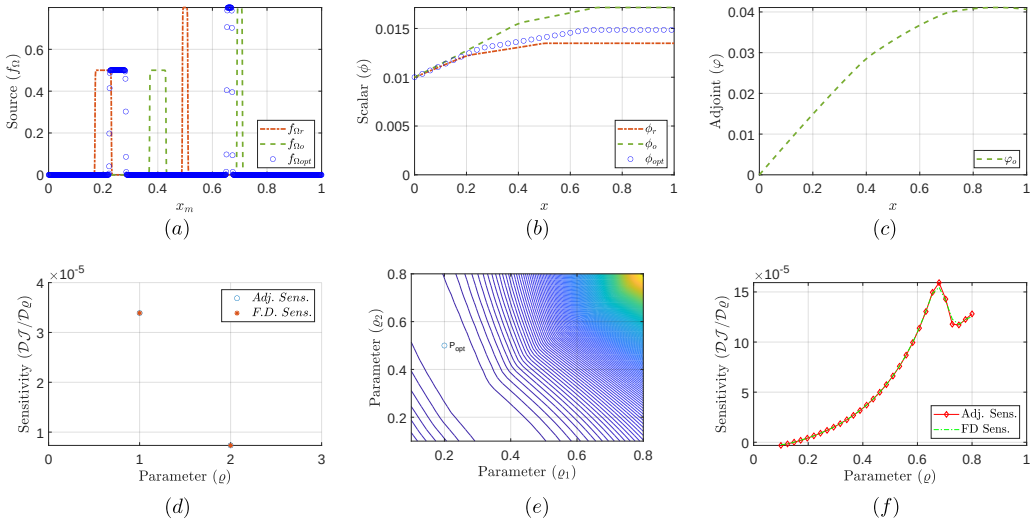

Figure 5.17: Two area scalar dependent source position optimization for 1D case (a) source, (b) scalar and (c) adjoint profiles. (d) Sensitivity of each source at initial state. (e) Contour of objective function relative to the coordinates of the two sources. (f) Sensitivity for a range of coordinate values of source 1.

For the 2D case, the flow is considered over a backward-facing step with a sudden expansion of 1:2, as shown in Fig. 5.18. Two rectangular solid blocks are represented using the Brinkman penalization approach and the level set method presented in Eq. (2-11). The initial coordinates and shape parameters of the sources are given in Table 5.1.

Table 5.1: Initial coordinates and shape parameters of sources.

Coordinates (m)		Shape parameters (m)	
ϱ_1 –> x_{a1}	0.08	r_{x1}	0.01
ϱ_2 –> x_{a2}	0.12	r_{x2}	0.002
ϱ_3 –> y_{a1}	0.005	r_{y1}	0.002
ϱ_4 –> y_{a2}	-0.005	r_{y2}	0.005

Sensitivity was calculated for the line error and total pressure loss cost functions, as shown in Figs. 5.18e and 5.18f. In analogy to a sensor within the computational domain, we use the line error cost function as a line with 20 points distributed in $y = 0.2$ and $-0.01 \leq x \leq 0.01$, with $\boldsymbol{u}_{\Omega_r} = (0.05, 0.0)m/s$. The sensitivity is given by the following expression,

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}_o} = \int_{\Omega_a} \alpha \nabla(\boldsymbol{v} \cdot \boldsymbol{u}) \, \mathrm{d}\Omega \tag{5-9}$$

We can see that there is a good agreement between the sensitivities.

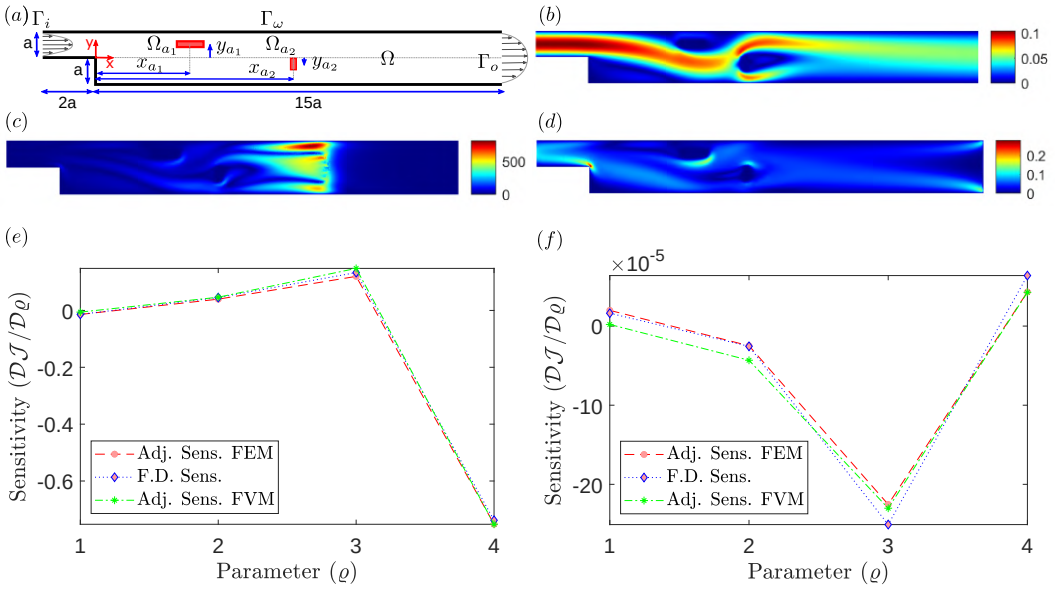

Figure 5.18: (a) Sketch of geometry and dimensions for two area source controls. Results using FEM: (b) primal velocity field; (c) adjoint velocity field for line error functional; (d) adjoint velocity field for total pressure loss functional; (e) sensitivities for line error functional and (f) sensitivities for total pressure loss functional.

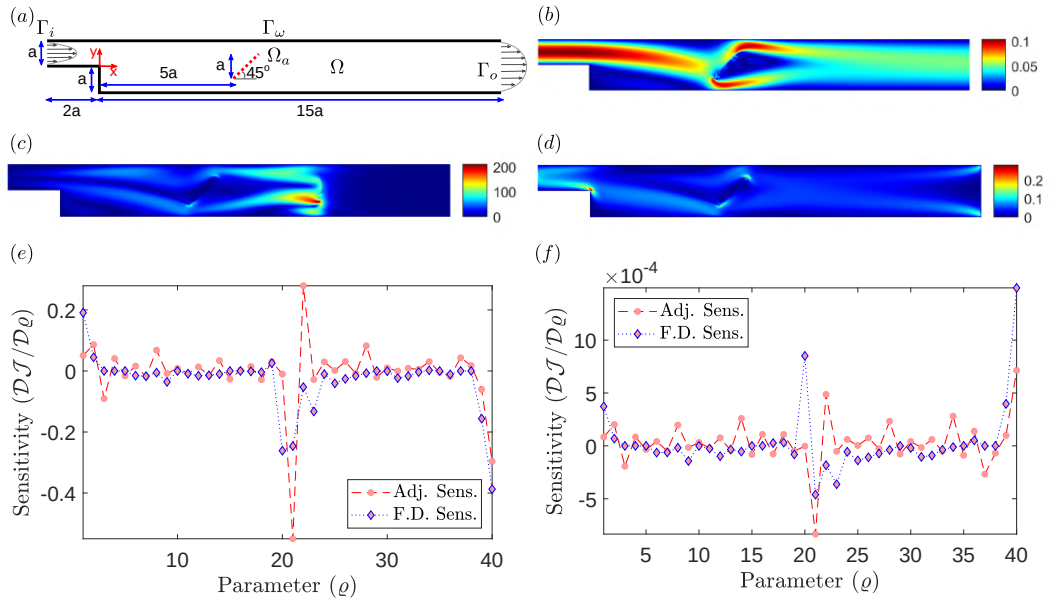

Figure 5.19: (a) Sketch of geometry and dimensions for line source controls. Results using FEM: (b) primal velocity field; (c) adjoint velocity field for line error functional; (d) adjoint velocity field for total pressure loss functional; (e) sensitivities for line error functional and (f) sensitivities for total pressure loss functional.

We can represent a solid boundary within the domain with twenty consecutive points to form a continuous line by adopting Darcy's law and using an interpolation method, as in Fig. 5.19. As in the previous case, the sensitivities are determined for the line error and total pressure loss cost

functions. The first twenty parameters are related to the x coordinate and the next twenty parameters are related to the y coordinate. We note that despite the agreement between the sensitivities using the adjoint and FD methods, both sensitivities exhibit variations. This results from using the inverse mapping interpolation method in the range FEM. When we use the Moving Least Square (MLS) interpolation method in the FVM domain, the solid line plot is better defined and the sensitivities are smoother, as shown in Fig. 5.20. This is convenient for the study of line deformations.

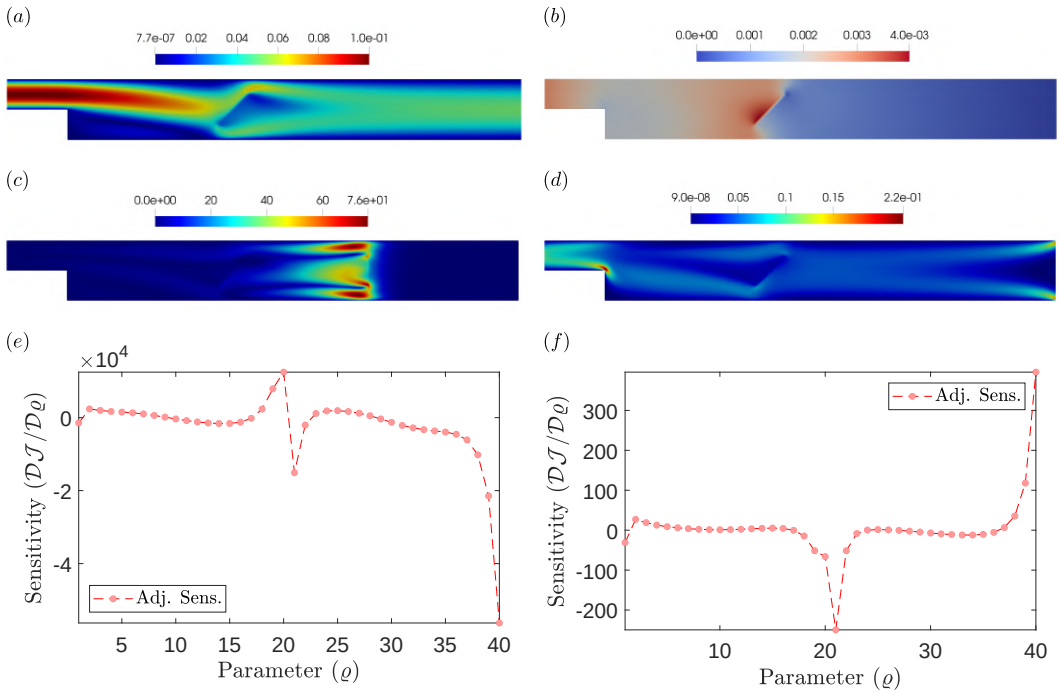

Figure 5.20: Results using FVM: (a) primal velocity and (b) pressure fields; (c) adjoint velocity field for line error functional; (d) adjoint velocity field for total pressure loss functional; (e) sensitivities for line error functional and (f) sensitivities for total pressure loss functional.

5.2.2 Sensitivity of the Inlet Boundary

In the same computational domain of flow over a backward-facing step, we place two uniform fluid inlets as shown in Fig. 5.21. The control variables in this case are the velocity components at each fluid inlet. Details of the two fluid inlets can be found in Table 5.2.

The sensitivity is given by the following expression:

$$\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\boldsymbol{g}_{\Omega_a}} = \int_{\Gamma_a} \boldsymbol{s}\, \mathrm{d}\Gamma \tag{5-10}$$

where in 2D from Eq. 2-77 the two components of $\boldsymbol{s}$ are:

Table 5.2: Coordinates, shape parameters and initial size of inlets.

Coordinates (m)		Shape p. (m)		Vel. x dir. (m/s).	Vel. y dir. (m/s)
x_{a1}	0.08	r_{x1}	0.005	ϱ_1 –>$u_1 = 0.01$	ϱ_3 –> $v_1 = 0.05$
x_{a2}	0.13	r_{x2}	0.004	ϱ_2 –>u_2 = -0.01	ϱ_4 –> $v_2 = 0.05$

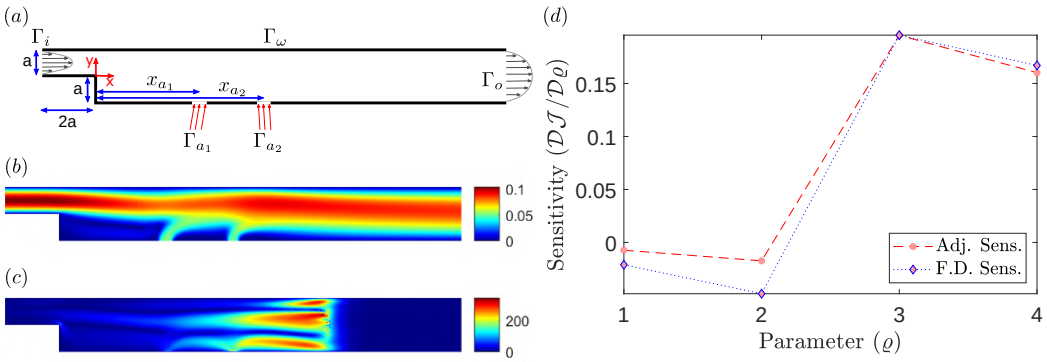

Figure 5.21: (a) Sketch of geometry and dimensions for two inlet boundary controls. Results using FEM: (b) primal velocity and (c) adjoint velocity fields for line error functional. (d) Sensitivity relative to each parameter.

$$s_x = -\frac{q}{\rho}n_x - \hat{u}(un_x + vn_y) - 2\nu\frac{\partial\hat{u}}{\partial x}n_x - \nu\left(\frac{\partial\hat{u}}{\partial y} + \frac{\partial\hat{v}}{\partial x}\right)n_y \tag{5-11}$$

$$s_y = -\frac{q}{\rho}n_y - \hat{v}(un_x + vn_y) - \nu\left(\frac{\partial\hat{u}}{\partial y} + \frac{\partial\hat{v}}{\partial x}\right)n_x - 2\nu\frac{\partial\hat{v}}{\partial y}n_y \tag{5-12}$$

where n_x and n_y are the normal components of each velocity inlet. In this particular case, we have $(n_x, n_y) = (0, -1)$ for both inlets. Moreover, the adjoint velocity components in the two inlets are zero. After simplification, we then have:

$$s_x = \nu\left(\frac{\partial\hat{u}}{\partial y} + \frac{\partial\hat{v}}{\partial x}\right) \tag{5-13}$$

$$s_y = \frac{q}{\rho} + 2\nu\frac{\partial\hat{v}}{\partial y} \tag{5-14}$$

Note that the sensitivities depend on the partial derivative of the adjoint velocity in the region of these boundaries. As we can see in Fig. 5.22, the x-component of the adjoint velocity has a strong gradient on these contours, which could lead to a decrease in the accuracy of the sensitivity calculation, as can be seen in Fig. 5.21d, where a not very good agreement is obtained between the sensitivities of the adjoint and FD methods for the velocity component in the x direction.

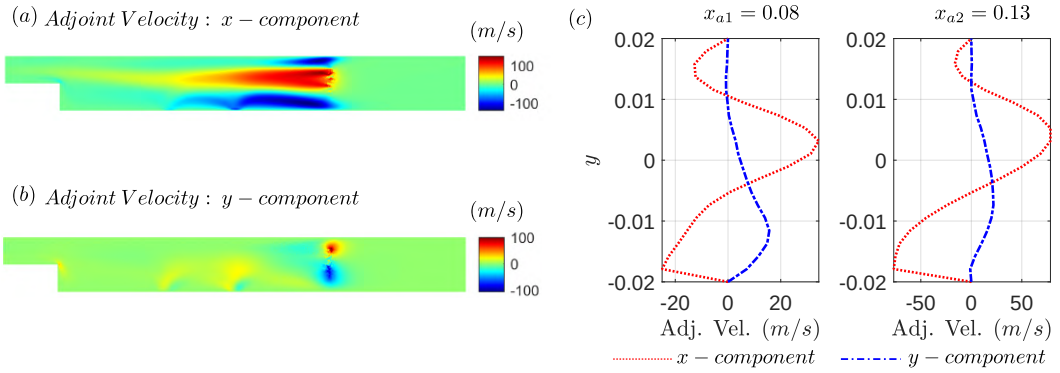

Figure 5.22: Adjoint velocity fields for (a) x-component and (b) y-component. (c) Profiles for the x and y components of the adjoint velocity at x_{a1} and x_{a2} along the y axis.

5.2.3 Sensitivity for Topology Optimization

To determine the sensitivity with respect to the pseudodensity γ for the optimization of material distribution, we assume $\gamma = 0.4$ uniformly throughout the domain as the initial structure. The sensitivity mapping for the total pressure loss and mean temperature cost functions are shown in Figs. 5.23c and 5.24e, respectively. Ten points were selected for comparison with the FD method. Figures 5.23d and 5.24f show that the sensitivities calculated with the adjoint method show good agreement with those calculated with the FD method.

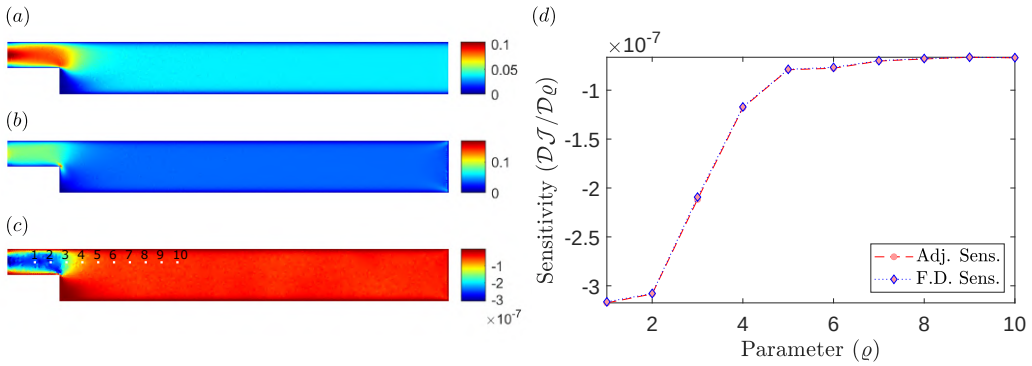

Figure 5.23: Topology optimization of fluid flow system. Results using FEM: (a) primal velocity, (b) adjoint velocity and (c) sensitivity fields with uniform initial value $\gamma = 0.4$ for total pressure loss functional. (d) Sensitivity profile at ten points.

Topological optimization can also work for a subregion of the computational domain, as shown in Fig. 5.25 shown. In this case, the objective function is the total pressure loss. The rectangular region can be determined by the exponential function presented in chapter 3. Ten points from the exponential function shown in Fig. 5.25d shown in the sensitivity map (white

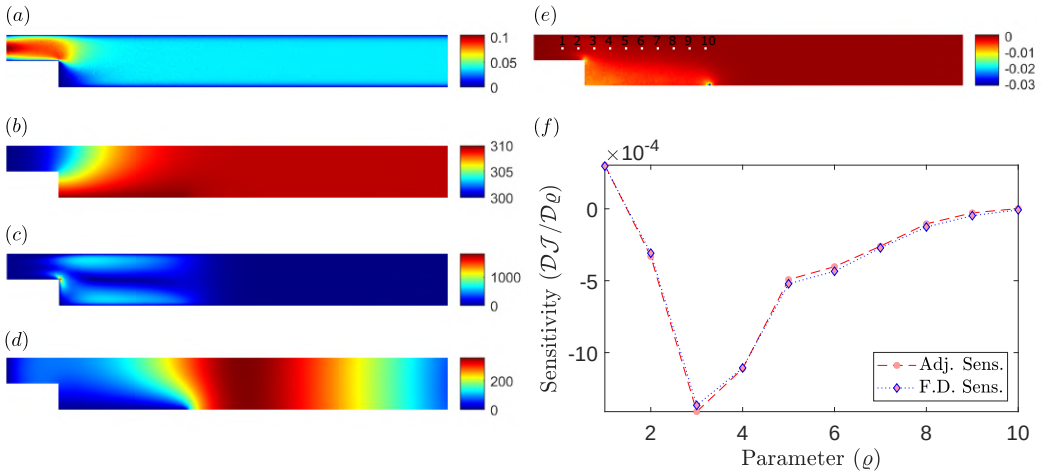

Figure 5.24: Topology optimization of thermal-fluid flow system. Results using FEM: (a) Primal velocity, (b) primal temperature, (c) adjoint velocity, (d) adjoint temperature and (e) sensitivity fields with uniform initial value $\gamma = 0.4$ for mean temperature functional. (f) Sensitivity profile at ten points.

color points) are compared with the results of the FD method and show good agreement.

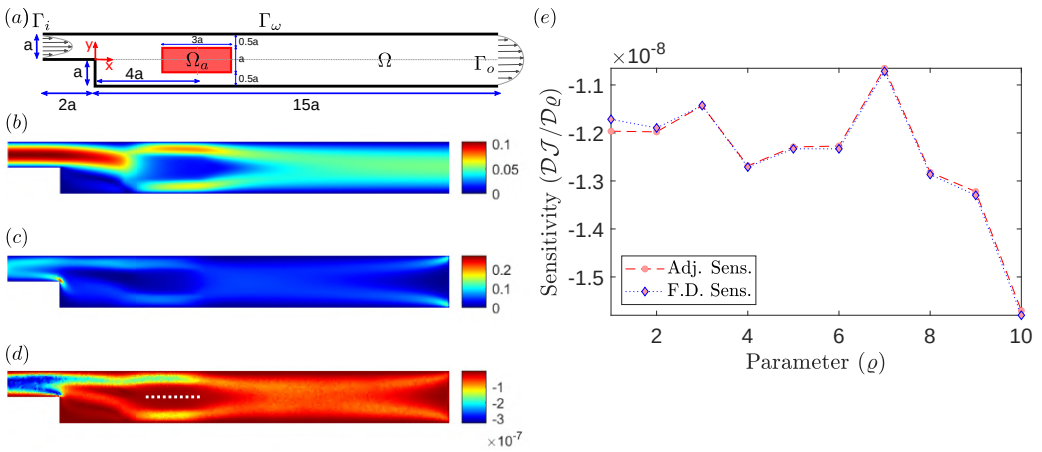

Figure 5.25: (a) Sketch of geometry and dimensions for partial topology optimization of fluid flow system. Results using FEM: (b) primal velocity, (c) adjoint velocity and (d) sensitivity fields with uniform initial value $\gamma = 0.4$ for total pressure loss functional. (e) Sensitivity profile at ten points.

6
Case Studies

In this chapter, we report the results of the source position and topology optimization performed with the solvers and algorithms presented in the previous chapters.

6.1
Source Position Optimization

Let's denote the flow domain by Ω sketched in Fig. 6.1, and let Γ denote its boundary. The inflowing and outflowing parts of the boundaries are denoted by Γ_i and Γ_o, respectively, and Γ_w is the wall boundary.

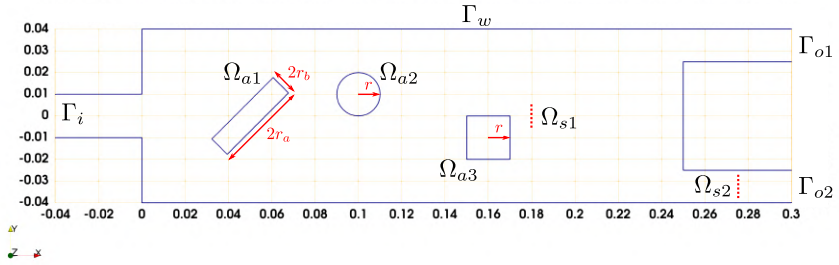

Figure 6.1: Study domain Ω. (a) Geometry dimensions and boundaries.

where $U = (0.1, 0)(m/s)$ is a uniform velocity prescribed at the inlet boundary, and the working fluid is air with $\nu = 1.39 \times 10^{-5}$, which will be used in all simulations from now on.

In the domain Ω, we place three sub domains Ω_a, whose geometric features are in Table 6.1.

Table 6.1: Geometric properties of the three subdomains Ω_a.

	Ω_{a1}	Ω_{a2}	Ω_{a3}
Central coordinate $x_o(m)$	0.05	0.1	0.16
Central coordinate $y_o(m)$	0.0	0.01	-0.01
Rotation	45°	0°	0°
r_a	0.02	-	-
r_b	0.005	-	-
r	-	0.01	0.01

These three subdomains Ω_a could be either explicit internal boundaries or implicitly represented solid regions. In both cases, the Ω domain was discretized using the OpenFOAM utilities Blockmesh and Snappyhexmesh, as shown in Fig. 6.2, where the details of the refinement and the layers of the mesh are shown.

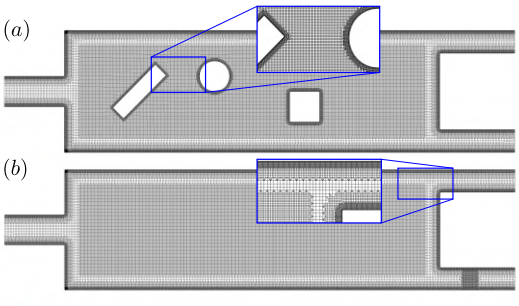

Figure 6.2: Mesh created with SnappyHexMesh-OpenFOAM. (a) Mesh domain with explicit internal boundaries. (b) Mesh domain without internal boundaries.

For the case of explicit internal boundaries, the flow field and streamlines are shown in Fig. 6.3. We will use this result as a 'base case' to compare with cases where regions are implicitly represented as solids, using the Brinkman penalization approach.

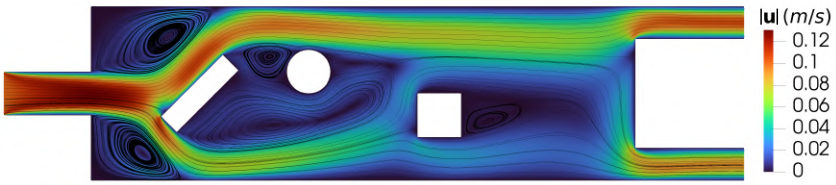

Figure 6.3: Velocity field with streamlines in a region with explicit internal boundaries.

We use the Brinkman penalization approach ($\boldsymbol{f}_{\Omega_a} = -\alpha \boldsymbol{u}$) to account for the presence of immersed solid regions in the fluid flow domain. In this section, we use $\alpha = 5000$ in all simulations.

In Fig. 6.4 we show the result of the implicit representation of the solid region within the domain and the velocity field using the Brinkman penalization approach and the level set method. We present three cases as follows:

- Case 1: Three individual sources whose characteristics are shown in Table 6.1.
- Case 2: Three groups of successive sources mimicking Case 1. Each group (rectangle, circle, and square) consists of 80 individual sources with the features shown in Case 3.
- Case 3: Three groups of successive sources to show how Case 2 was composed. Each group consists of 10 individual sources. The small circles have a radius of $0.002m$ and the side length of the small squares is $0.004m$. The small squares that make up the rectangle are rotated 45°.

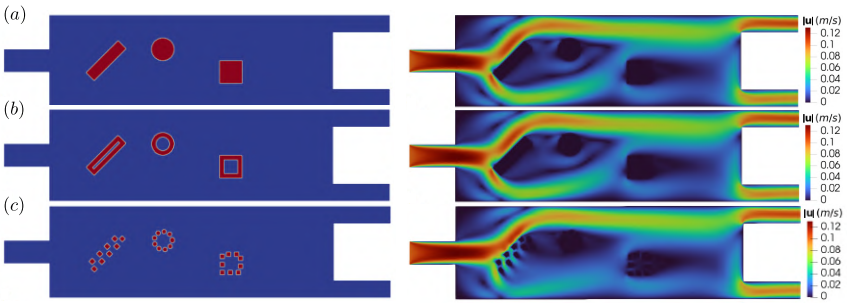

Figure 6.4: Implicit representation of the solid region within the domain and velocity field for (a) Case 1: Three single sources, (b) Case 2: Successive small sources to represent Case 1, and (c) Case 3: Successive small sources to show how Case 2 arose.

In Fig. 6.4, the x - component of the velocity profiles for the base case, case 1 and case 2, are compared at positions $x = 0.08m$, $x = 0.14m$ and $x = 0.2m$. The results show very good agreement for the three cases.

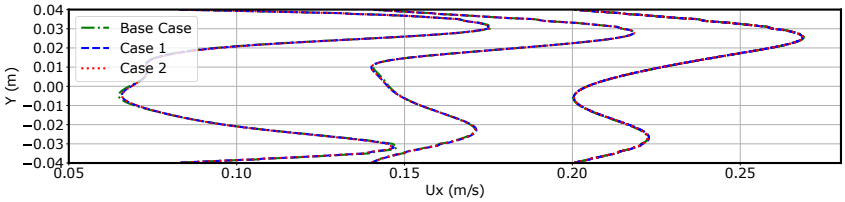

Figure 6.5: Comparison of the x - component of the velocity profiles for the base case, case 1 and case 2. The positions of the profiles are at $x = 0.08m$, $x = 0.14m$ and $x = 0.2m$.

We employed the functional of error,

$$\mathcal{J} = \frac{1}{2}\int_{\Omega_s} (\boldsymbol{u} - \boldsymbol{u}_{ref})^2 \, \mathrm{d}\Omega \tag{6-1}$$

where $\boldsymbol{u}_{ref}$ is a reference velocity. Moreover, we also use the power dissipation of the fluid device, as presented in Eq.(3-6).

We define the cost function of the error in the subdomain Ω_s as shown in Fig. 6.1, which is a line formed by neighboring discrete points in the domain, details are shown in Table 6.2. Note that the partial derivative of the cost function of the error appears in Eq.(2-52) as a source term, i.e., successive point sources. We can think of this type of objective function as analogous to a sensor within the computational domain.

In general, finite volume solvers provide information only at the nodal points and at the center of the elements. Discrete points in the model that do not coincide with the grid must be interpolated to arbitrary points in the domain. In this work, we use an interpolation approach called Moving Least Square (see Appendix A.3.2 for details).

Table 6.2: Properties of the error cost function in the two subdomains Ω_s.

	Ω_{s1}	Ω_{s2}
Coordinate $x(m)$	0.18	0.275
Coordinate $y(m)$	[-0.039,-0.026]	[-0.006,0.007]
Number of points	10	10
Reference velocity $\boldsymbol{u}_{ref}(m/s)$	(0.08,0)	(0.15,0)

Source position optimization is performed using the Steepest Descent method as follows:

$$\mathbf{x}_a = min(max(\mathbf{x}_a + \lambda \cdot \nabla_{\mathbf{x}_o}\mathcal{J}, \mathbf{x}_{min}), \mathbf{x}_{max}); \tag{6-2}$$

where λ is the step size (we use $\lambda = 10^{-3}$ in all our study cases). The bounds constraining the source positions are the bounds of the range Ω, i.e. $\mathbf{x}_{min} = [0.0, -0.04]m$ and $\mathbf{x}_{max} = [0.25, 0.04]m$.

Consider the optimization resulting from minimizing the error functional on Ω_{s1} with respect to the position of the sources for cases 1 and 2, as shown in Fig.6.4. Figures 6.6a and 6.6b show the position of the sources and the corresponding velocity field after 100 iterations in each case. In both cases, the cost function is minimized, as shown in Fig. 6.6c. The sources are arranged so that the fluid flow is close to that given in Ω_{s1}, i.e., $\boldsymbol{u}_{ref} = (0.08, 0)m/s$. The variation of the objective function behaves differently in the two cases. In case 1, the cost function starts to vary near the minimum value after iteration 40. This could be because we are using a simple optimization algorithm with constant step size λ. In contrast, in Case 2, where we have 240 small individual sources, the objective function varies from the beginning, which is mainly caused by the separation of the sources, which creates a complex flow.

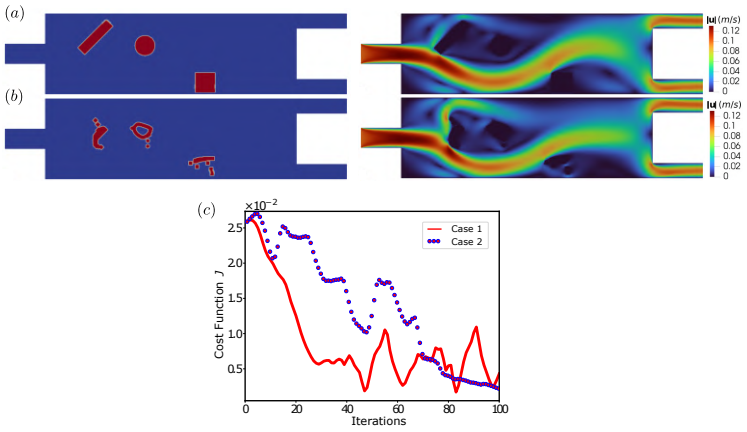

Figure 6.6: Position of sources after 100 iterations and velocity field for cases (a) 1 and (b) 2, using the error functional in Ω_{s1}. (c) Variation of the objective function with the number of iterations for both cases.

If we perform the optimization resulting from the minimization of the total pressure loss functional with respect to the position of the sources for case 1, as in Fig. 6.4, we obtain the position of the sources and the corresponding velocity field after 100 iterations, as shown in Fig. 6.7a. In this case, the cost function is minimized, as shown in Fig. 6.7b. The sources are arranged to minimize obstruction to the fluid flow path, resulting in a reduction of the total pressure loss.

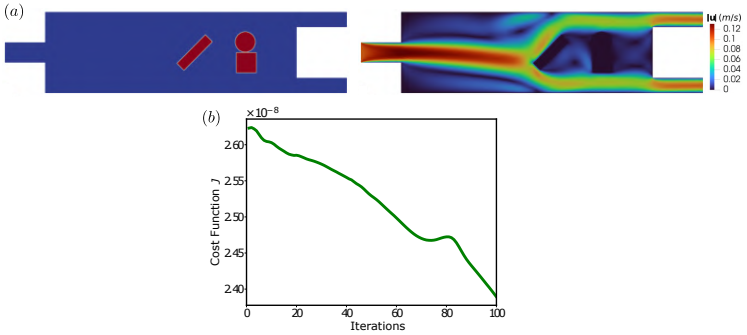

Figure 6.7: (a) Position of sources after 100 iterations and velocity field for case 1, using the total pressure loss functional. (b) Variation of the objective function with the number of iterations.

These cases illustrate the importance of knowing the physical phenomenon of the problem and thus being able to formulate objective functions that can be minimized.

We have found practical applications for optimizing the source position, namely designing the shape of impermeable inner walls in the fluid flow region. As shown in Case 2, we can model interesting solid shapes with a number of small, adjacent sources. To illustrate the design of different configurations, we use small circles with a radius of $0.002m$ in all of the following cases. In each case, we represent the variation of the objective function, the position of the sources, and the velocity field. We also represent the magnitude of the sensitivity field in the whole domain, expressed as:

$$\left|\frac{\mathcal{D}\mathcal{J}}{\mathcal{D}\mathbf{x}}\right| = |\alpha\nabla(\boldsymbol{v}(\mathbf{x}) \cdot \boldsymbol{u}(\mathbf{x}))| \tag{6-3}$$

The sensitivity field shows us where the sensitivity is stronger in relation to the design parameters, in our case the position of the sources. In this way, we can determine if a change in a design parameter causes a change in the objective function.

6.1.1 Line Shape

Figure 6.8a shows the case where a sequence of 200 small circular sources forms a line shape. The artificial solid line acts like a wall obstructing the fluid flow and causing separation of the flow, as shown by the velocity field in Fig. 6.8b. In this case, we aim for the fluid velocity in the region Ω_{s2} to reach a constant value of $\boldsymbol{u}_{ref} = (0.15, 0)m/s$, as given in Table 6.2. The region Ω_{s2} is where the wall slows down the flow velocity near it, so we cannot obtain a uniform velocity. Therefore, we expect that the average velocity $\boldsymbol{u}_{ref}$ converges once the artificial solid line takes a shape.

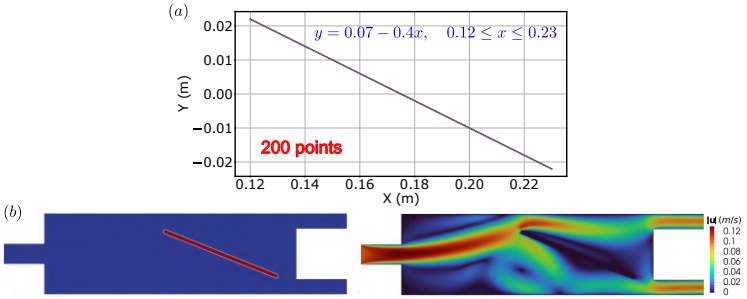

Figure 6.8: Consecutive circular small sources forming a solid line in the domain. (a) Geometric details of the line source. (b) Initial position of the sources and velocity field.

The sensitivity field shown in Fig. 6.9a warns us that the position of the upper left side of the line is changed. Figure 6.9c shows the shape of the line and the corresponding velocity field after 100 iterations. The cost function decreases smoothly, as shown in Fig. 6.9b. The line mainly fits at the extremes so that the fluid flow in Ω_{s2} increases to approach $\boldsymbol{u}_{ref} = (0.15, 0)m/s$. On the upper left side, the line is modeled to impede the flow of the fluid, and on the lower right side, it forms a hook to reduce the area through which the fluid flows to produce its acceleration.

6.1.2 "Y" channel Shape

Figure 6.10a shows the geometry details for the case where a series of small circular sources form a "Y" channel shape. The channel consists of six lines, with lines 1 and 2 formed by 200 small circles each. The remaining lines are made up of 100 small circles each. The artificial solid lines serve as a wall to guide the fluid flow, as shown in Fig. 6.10b.

First, we will consider the optimization that results from minimizing the error functional for Ω_{s2} with respect to the position of the sources. The sensitivity field shown in Fig. 6.11a warns us that the position of the sources

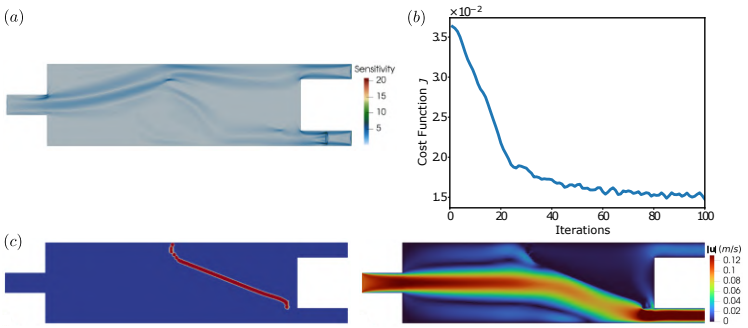

Figure 6.9: Optimization of the solid line case using the error objective function in Ω_{s2}. (a) Initial magnitude of the sensitivity field in the whole domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 100 iterations and velocity field.

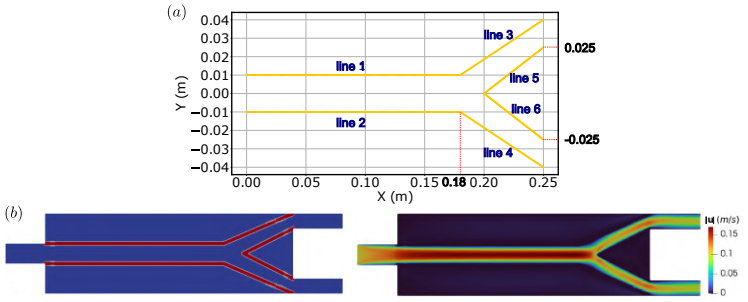

Figure 6.10: Successive circular small sources form solid lines implicitly representing a "Y" channel in the domain. (a) Geometric details of the lines. (b) Initial position of the sources and velocity field.

is strongly affected by the division of the channel and has no effect on the objective function near the inlet. Figure 6.11c shows the shape of the lines and the corresponding velocity field after 100 iterations. The sources mainly close the passage of the fluid through the outlet Γ_{o2}, so the fluid flow in Ω_{s2} is increased to approach $\boldsymbol{u}_{ref} = (0.15, 0)m/s$. The cost function decreases rapidly until iteration 10, when small oscillations occur, as shown in Fig. 6.11b.

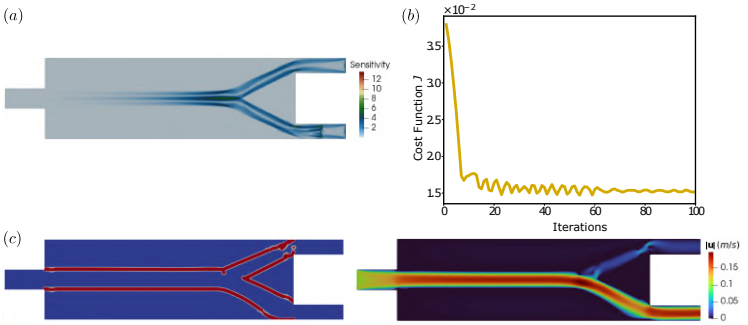

Figure 6.11: Optimization of the solid line case forming an implicit "Y" channel using the error functional in Ω_{s2}. (a) Initial magnitude of the sensitivity field in the whole domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 100 iterations and velocity field.

Now we treat the optimization that results from minimizing the total pressure loss with respect to the position of the sources. Moreover, the sources are forced to move only in y direction. The sensitivity field shown in Fig.6.12a warns us that the position of the sources is affected uniformly along the path. Figure 6.12c shows the shape of the lines and the corresponding velocity field, after 100 iterations. Here, the channel widens and takes on a smooth profile at the end of the outlet for the passage of the fluid. It is also interesting to see that the symmetry is not preserved. The cost function decreases smoothly, as shown in Fig. 6.12b.

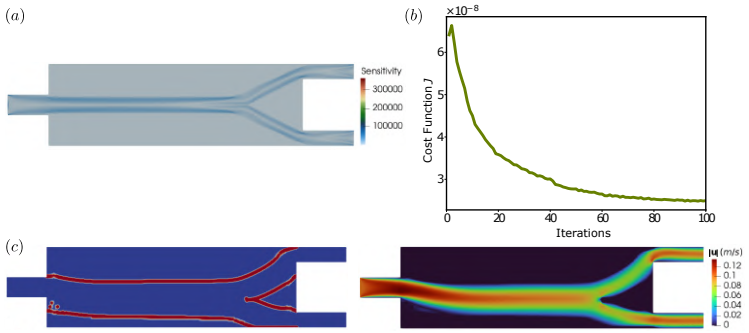

Figure 6.12: Optimization of the solid line case forming an implicit "Y" channel using the total pressure loss functional. (a) Initial magnitude of the sensitivity field in the whole domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 100 iterations and velocity field.

6.1.3 "Y" channel Shape with Airfoil

Figure 6.13a shows the geometry details for the case where a series of small circular sources form a "Y" channel shape with an airfoil in it. The channel consists of six lines, with lines 1 and 2 formed by 200 small circular sources each. The remaining lines consist of 100 small circles each. Line 1 is a curve that follows a polynomial equation. The airfoil (NACA 2414) consists of 200 small circle sources. The artificial solid lines act as walls that guide the flow of the fluid, as shown in Fig. 6.13b.

Here we optimize the position of the sources so that the total pressure loss is minimized. Also, the sources must move only in the y direction. The sensitivity field shown in Fig. 6.14a warns us that the position of the sources would be affected uniformly along the path, especially at the narrowest points. Figure 6.14c shows the shape of the channel and the corresponding velocity field after 7 iterations and Fig. 6.14d after 100 iterations. Here the channel becomes wider and takes on a smooth shape at the end of the outlets. It is also

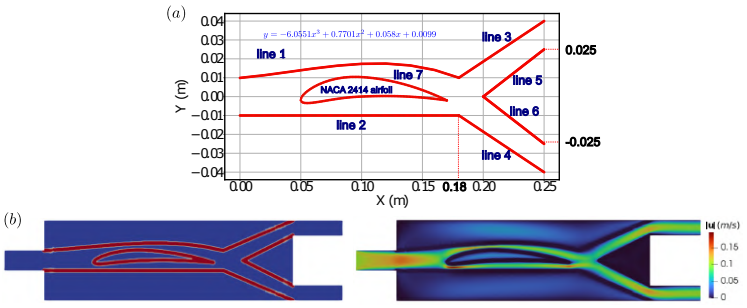

Figure 6.13: Successive circular small sources forming solid lines, implicitly representing a "Y" channel with an airfoil in it. (a) Geometric details of the lines. (b) Initial position of the sources and velocity field.

interesting to see how the airfoil gradually becomes a line. The cost function decreases smoothly, as shown in Fig. 6.14b.

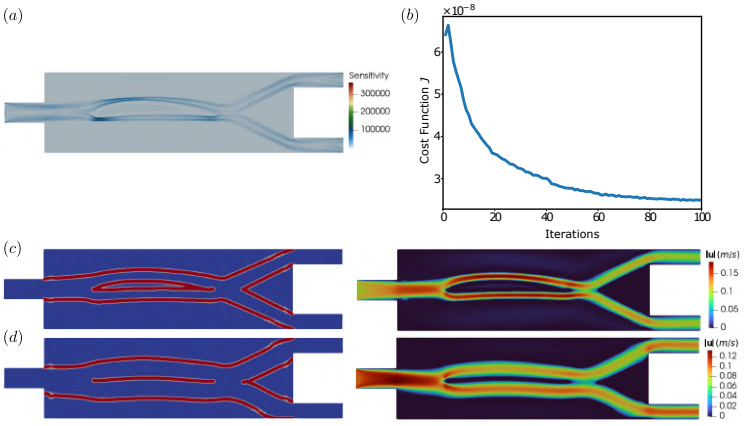

Figure 6.14: Optimization of the case of the solid line forming an implicit "Y" channel with an airfoil therein, using the total pressure loss functional. (a) Initial magnitude of the sensitivity field in the entire domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 7 iterations and velocity field. (d) Position of sources after 100 iterations and velocity field.

6.1.4 Sources Randomly Distributed

Figure 6.15a shows the geometry details for the case where 1000 small circular sources are randomly distributed in the domain. The random distribution of sources acts like a porous medium in which the fluid flows, as shown in Fig. 6.15b.

Here we optimize the position of the sources so that the total pressure loss is minimized. Also, the sources must move only in the y direction. The sensitivity field shown in Fig. 6.16a warns us that the sensitivity of the sources position are high at points where the fluid is accelerating. Figure 6.16c shows

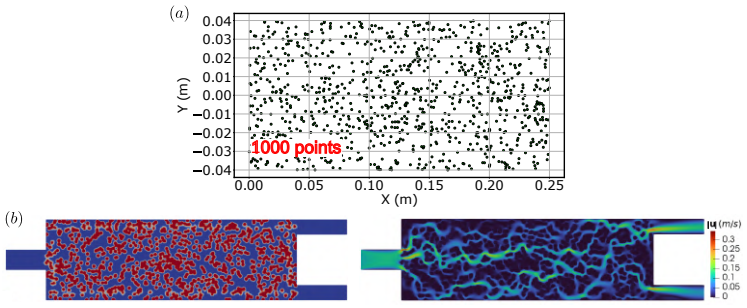

Figure 6.15: Circular small sources randomly distributed throughout the whole domain, representing a porous medium. (a) Geometric details of the sources. (b) Initial position of the sources and velocity field.

the distribution of the sources and the corresponding velocity field after 100 iterations. Interestingly, the sources are distributed in such a way that the fluid flow adopts a profile similar to that shown in Fig. 6.12. The cost function decreases smoothly as shown in Fig. 6.16b.

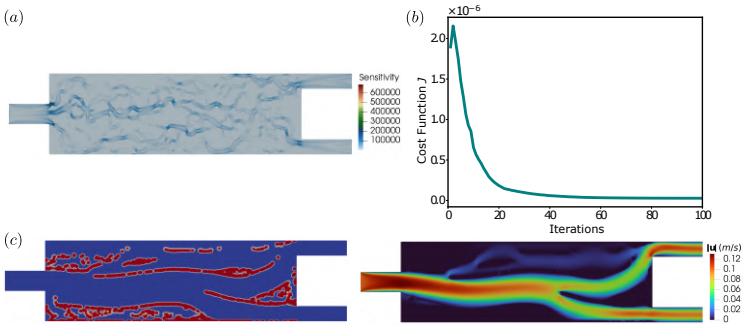

Figure 6.16: Optimization of the position of circular small sources randomly distributed over the whole domain, using the total pressure loss functional. (a) Initial magnitude of the sensitivity field throughout the domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 100 iterations and velocity field.

6.1.5 Sources Randomly Distributed with "Y" Path

In the previous case, when the number of sources increases, the passage of the fluid is restricted and the sensitivity is very low, and there is no shift of the sources in the domain. For this reason, a path was intentionally created here. Figure 6.17a shows the geometry details for the case where 2000 small circle sources are randomly distributed in the domain, where a "Y" path was created by removing sources near the blue line. The initial distribution of sources and velocity field is shown in Fig. 6.17b.

First, we optimize the position of the sources so that the total pressure loss is minimized. Also, the sources must move only in the y direction. The

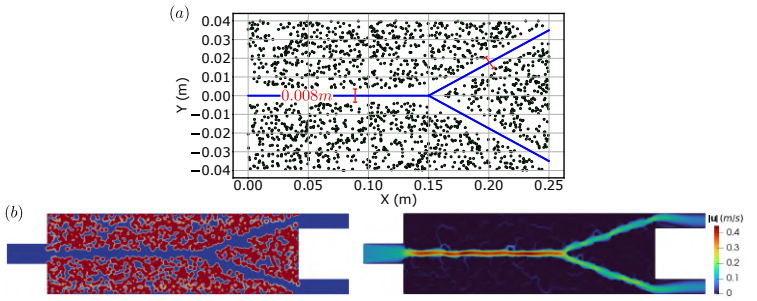

Figure 6.17: Small circular sources randomly distributed throughout the whole domain with a "Y" path in it, representing a porous medium. (a) Geometric details of the sources. (b) Initial position of the sources and velocity field.

sensitivity field shown in Fig. 6.18a shows us a strong sensitivity at the beginning of the flow path where the fluid is accelerated. Figure 6.18c shows the distribution of sources and the corresponding velocity field after 100 iterations. The resulting path along which the fluid flows is similar to the result obtained by topology optimization with the same objective function. The cost function decreases slightly, as shown in Fig. 6.18b.

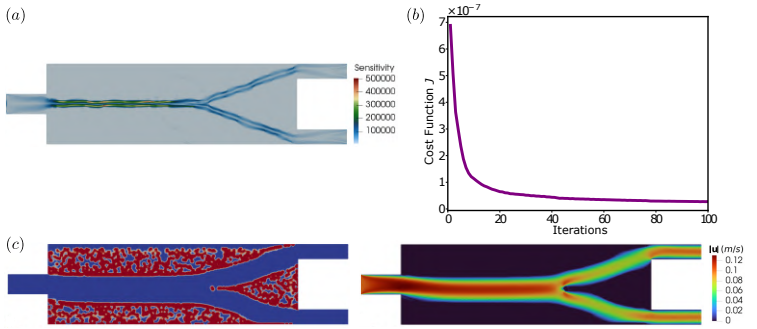

Figure 6.18: Optimization of the position of small circular sources randomly distributed throughout the domain with a "Y" path therein, using the total pressure loss functional. (a) Initial magniturde of the sensitivity field in the whole domain. (b) Variation of the objective function with the number of iterations. (c) Position of sources after 100 iterations and velocity field.

Now we optimize the position of the sources that minimizes the error functional on Ω_{s2}. The sensitivity field shown in Fig. 6.19a warns us that the position of the sources is strongly affected from the division of the path and has no effect on the objective function near the inlet. Figure 6.19c shows the shape of the lines and the corresponding velocity field after 100 iterations. The sources mainly close the passage of the fluid through the outlet Γ_{o2}, so the fluid flow in Ω_{s2} is increased to approach $\boldsymbol{u}_{ref} = (0.15, 0)m/s$. Note that the sources do not move in the middle part. The cost function decreases rapidly until iteration 30, when small oscillations begin to occur, as shown in Fig. 6.19b.

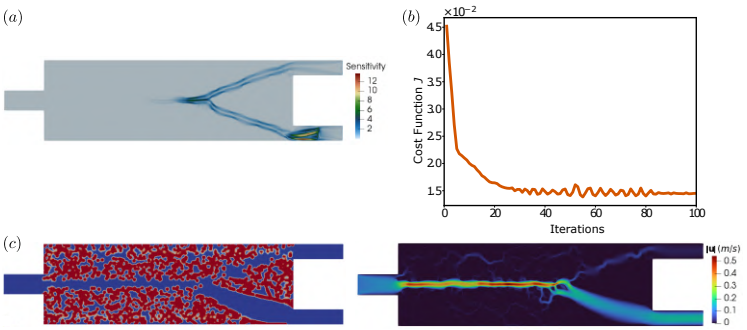

Figure 6.19: Optimization of the position of circular small sources randomly distributed throughout the domain with a "Y" path in it, using the error functional in Ω_{s2}. (a) Initial magnitude of the sensitivity field in the whole domain. (b) Variation of the objective function with the number of iterations. (c) Position of the sources after 100 iterations and velocity field.

6.2
Topology Optimization of Heat Exchanger

Heat exchangers are widely used in fluid cooling to remove high heat fluxes. In this regard, proper design of the cooling channels can increase the efficiency of the systems and ensure the safety of the products. Currently, the CFD -based adjoint method is a suitable approach for optimization in thermo-fluid dynamics.

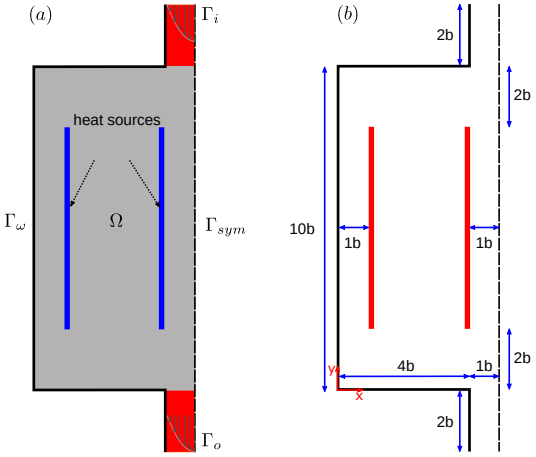

Figure 6.20: Case of the heat line sources. Sketch of geometric domain including definitions of (a) boundaries and (b) dimensions.

In this case study we will consider two heat transfer systems. The first system with internal heat sources and the second with heat transfer with convection on the walls. The sketch of the geometrical domain including the definitions of the boundaries and the details of the dimensions for both cases are shown in Fig. 6.20 and Fig. 6.21.

The adopted parameters are shown in Tab 6.3. Aluminum and water are the working materials and their physical properties are in Tab. 6.4.

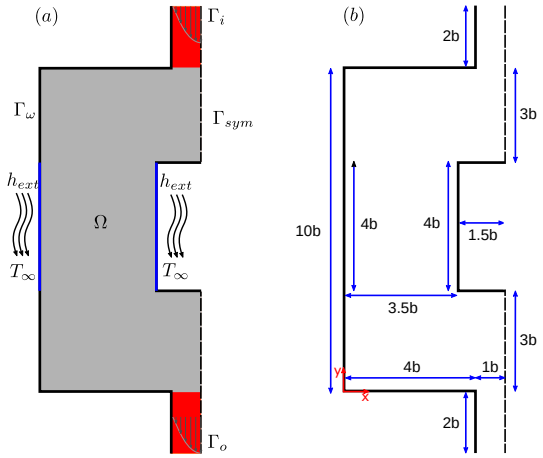

Figure 6.21: Case of convective heat flow. Sketch of geometric domain including definitions of (a) boundaries and (b) dimensional data.

Table 6.3: Parameters of design for topology optimization.

Parameter	Value
Inlet velocity $\boldsymbol{u}_i$ (ms^{-1})	(0 , -0.05)
Inlet Temperature T_i (K)	300
Reference Temperature T_{Ω_r} (K)	300
External Temperature T_∞ (K)	350
Convective heat transfer coefficient (h_{ext})	1000
Characteristic length scale 2b (m)	0.002
Reynolds number (Re)	100
Length of line sources (m)	0.006

Table 6.4: Material properties of aluminium and water.

Material properties	Aluminium	Water
Thermal conductivity, k $(Wm^{-1}K^{-1})$	237	0.61
Density, ρ (kgm^{-3})	2700	1000
Specific heat, Cp $(Jkg^{-1}K^{-1})$	880	4180
Inverse of permeability, α $(kgm^{-3}s^{-1})$	2×10^{-5}	-
Kinematic viscosity, ν (m^2s^{-1})	-	1×10^{-6}

Topology optimization of flow systems has been studied by several researchers. A typical case is the one shown in Fig. 6.22, where the objective is to minimize the total pressure loss. The result is a conduit that easily bypasses the obstacle as it traverses the channel.

Figure 6.23 shows the variation of the objective function with the number of cycles, as well as the evolution of the topology in different cycles

Another case is the design of a heat sink studied by Yu et al. (2020). Figure 6.24 shows the topology optimization of a heat sink with a uniform heat source $Q = 10^8 W/m^2$ generated throughout the domain. The objective is to minimize the error cost function to obtain a uniform temperature

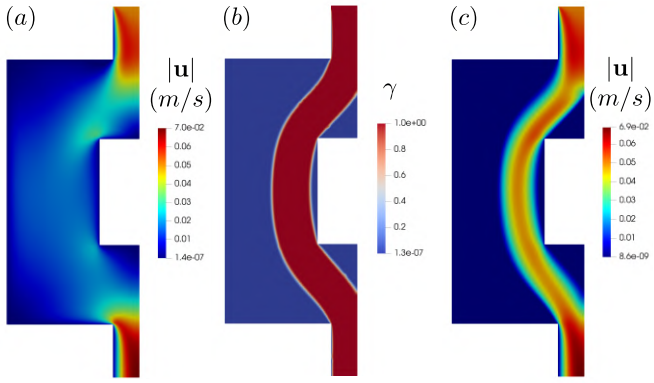

Figure 6.22: Topology optimization of a fluid flow system to minimize total pressure loss. (a) Initial velocity field with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (b) pseudo-density and (c) primal velocity fields.

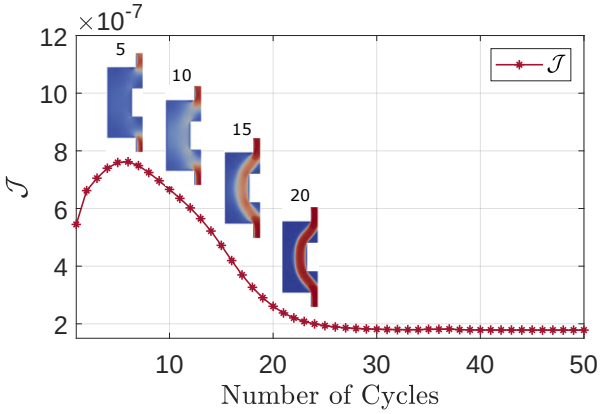

Figure 6.23: Variation of the objective function with the number of design cycles for a bypass channel design.

$T_{\Omega_r} = 300K$ throughout the domain, while maintaining a reference power dissipation ($\bar{P} = 1.58 \times 10^{-7}$) (corresponding to a straight cooling channel) and the volume fraction occupied by the fluid ($\beta = 0.4$).

The optimal result is a channel with branches that maintain connectivity and occupy the entire domain. The maximum temperature decreases and becomes more uniform throughout the domain compared to the initial state.

Figure 6.25 shows the variation of the objective function with the number of cycles. Note that the minimum of the objective function occurs around the twentieth cycle. However, the evolution of the topology in the different cycles indicates that the optimal topology must satisfy the constraints, as well as a clear separation between the solid and liquid parts.

In all subsequent cases, we consider the constraints on the reference power dissipation ($\bar{P} = 1.58 \times 10^{-7}$) and the volume fraction occupied by the fluid ($\beta = 0.4$).

Now we present a special case of thermal control. Let us consider the

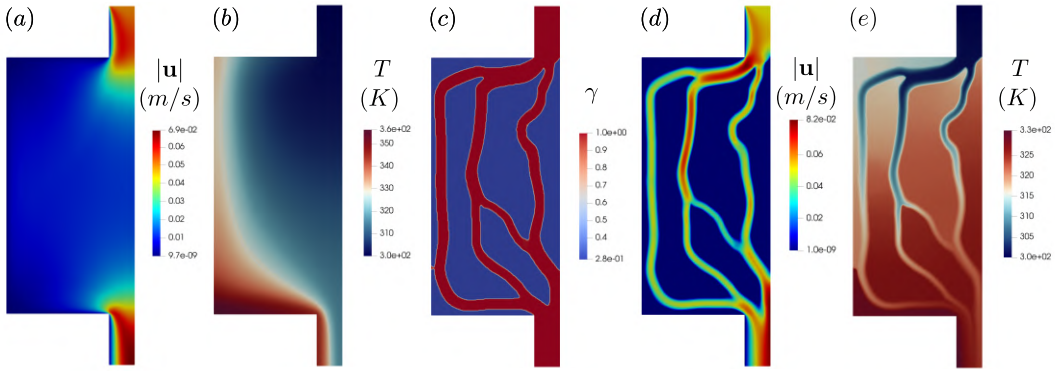

Figure 6.24: Topology optimization of a thermo-fluid flow system (design of a heat sink) to minimize the error function and total pressure loss. Initial (a) primal velocity and (b) temperature fields with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (c) pseudo-density, (d) primal velocity and (e) temperature fields.

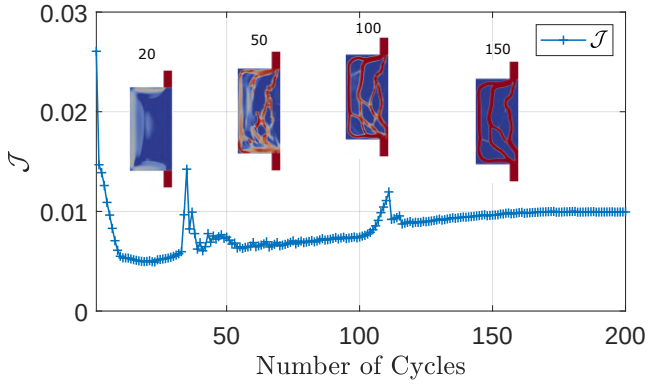

Figure 6.25: Variation of the objective function with the number of design cycles for heat sink design.

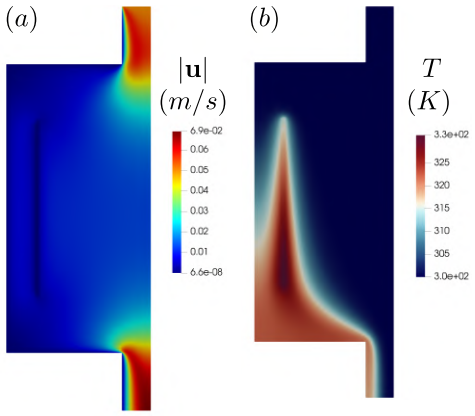

Figure 6.26: Heat-fluid flow system with a heat line source. Initial (a) velocity and (b) temperature fields with $\gamma = 0.4$ uniform throughout the domain.

case of a single-line heat source from Fig. 6.20. Figure 6.26 shows the initial velocity and temperature fields. The line heat source has a uniform value of $f_{\Omega_a} = 500\,W/m^2$, and the objective function is the line error functional located

in the same region of the source.

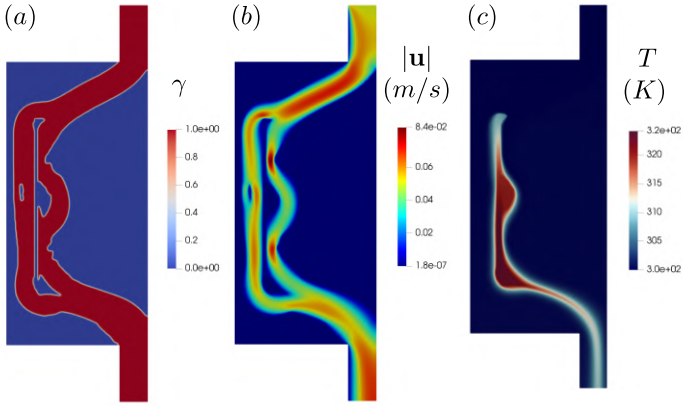

Figure 6.27: Topology optimization of a heat-fluid flow system with a heat conduction source and line error cost function with $T_{\Omega_r} = 300K$. Optimal results, (a) pseudo-density, (b) primal velocity, and (c) temperature fields.

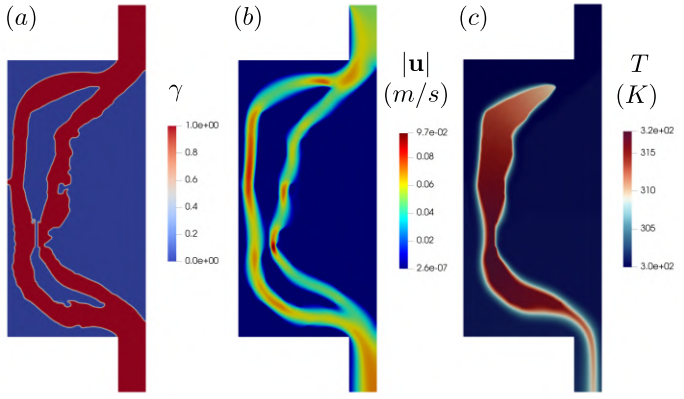

Figure 6.28: Topology optimization of a heat-fluid flow system with a heat conduction source and line error cost function with $T_{\Omega_r} = 315K$. Optimal results, (a) pseudo-density, (b) primal velocity, and (c) temperature fields.

Figures 6.27 and 6.28 show the results when the reference temperature T_{Ω_r} of the line error functional is equal to $300K$ and $315K$, respectively. When the reference temperature is $300K$, we see that the final topology of the branched channel passes very close to the source to dissipate the heat, and the shape that the channel takes serves to keep the temperature uniform, since this was specified in the objective function. In this case, the assigned reference temperature is inappropriate to achieve a uniform temperature of $300K$ in the source region. When the reference temperature is $315K$, we see that the source region strongly approaches $315K$. However, the final topology of the channel has many shortcomings.

Next, we let the two heat sources from Fig. 6.20 vary linearly with temperature as: $f_{\Omega_a} = 10T - 2800$. Figure 6.29 shows the optimized results, where we see that the fluid circulating in the channel forms a kind of

encapsulation to ensure that the region occupied by the two sources is kept at a uniform temperature equal to $315K$.

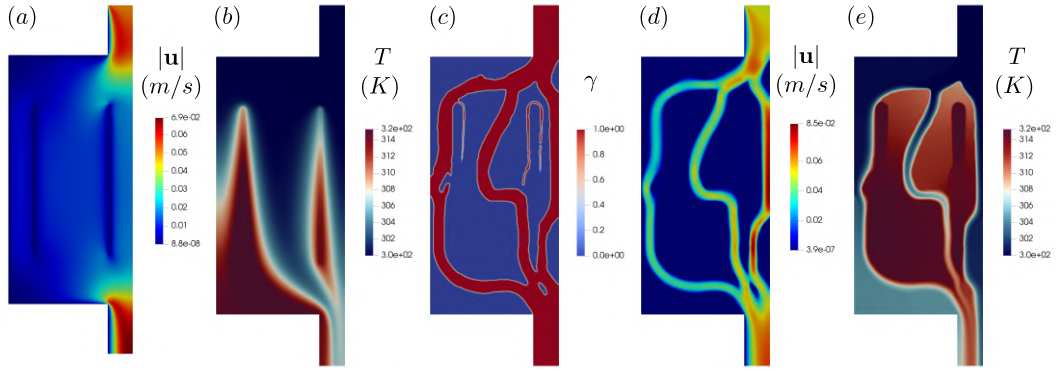

Figure 6.29: Topology optimization of a heat-fluid flow system with two non-uniform heat line sources to minimize the line error at $T_{\Omega_r} = 315K$. Initial (a) primal velocity and (b) temperature fields with $\gamma = 0.4$ uniform throughout the domain. Optimal results (c) pseudo-density, (d) primal velocity and (e) temperature fields.

In the next configurations, we work with the error cost function encompassing the entire domain, with the goal of bringing the temperature uniformly to $300K$ throughout the domain. Figure 6.30 shows the result of topology optimization of a heat-fluid system with two uniform sources equal to $f_{\Omega_a} = 500\,W/m^2$. We see that the topology configuration is such that the fluid occupies regions of higher temperature, which leads to a decrease in the maximum temperature and results in a more homogeneous temperature field compared to the initial temperature.

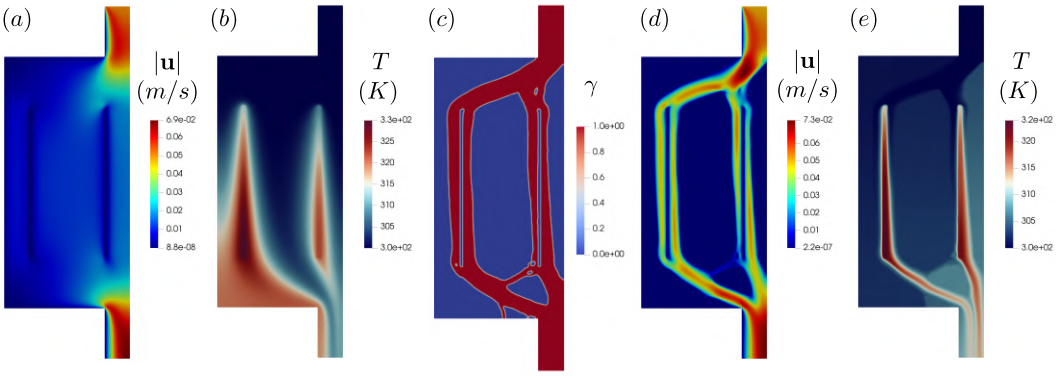

Figure 6.30: Topology optimization of a heat-fluid flow system with two heat line sources to minimize the error function . Initial (a) primal velocity and (b) temperature fields with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (b) pseudo-density, (c) primal velocity and (d) temperature fields.

The solid material considered in the previous case is the aluminum with high thermal conductivity, which contributes to the heat transfer through the solid part. Heat exchangers are often lined with insulating materials or

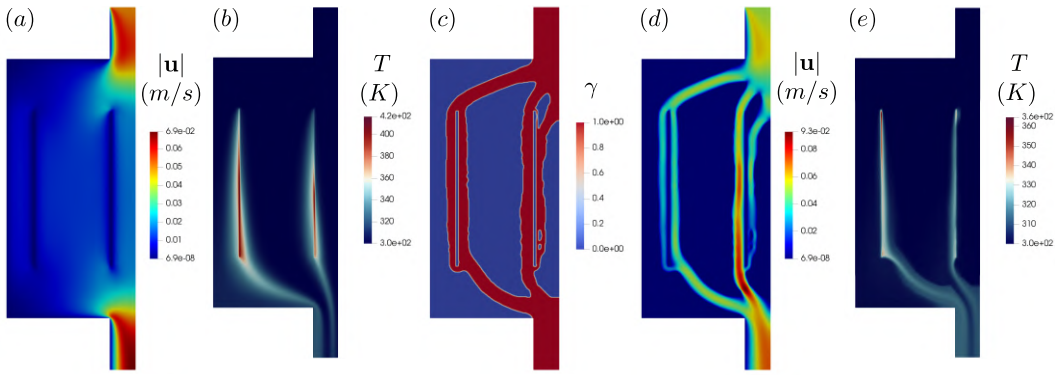

Figure 6.31: Topology optimization of a thermo-fluid flow system with two heat line sources and a low conductivity solid to minimize the error function. Initial (a) primal velocity and (b) temperature fields with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (b) pseudo-density, (c) primal velocity and (d) temperature fields.

exposed to air. Therefore, in this case, we consider a solid material with a very low conductivity, equivalent to that of air ($0.024\,W/mK$). Figure 6.31 shows an initial high temperature concentrated in each line source region. The final topology is very similar to the previous case and the temperature drops sharply.

Finally, we consider the case where part of the wall boundary is exposed to convective heat transfer, i.e., the convective boundary condition is given by:

$$\boldsymbol{n} \cdot k\nabla T|_{\Gamma_\omega} = h_{ext}(T - T_\infty) \tag{6-4}$$

and the corresponding adjoint boundary results:

$$\boldsymbol{n} \cdot k\nabla \varphi|_{\Gamma_\omega} = h_{ext}\varphi \tag{6-5}$$

where $\boldsymbol{n}$ is the external unit normal, T_∞ is the external temperature, and h_{ext} is the convective heat transfer coefficient.

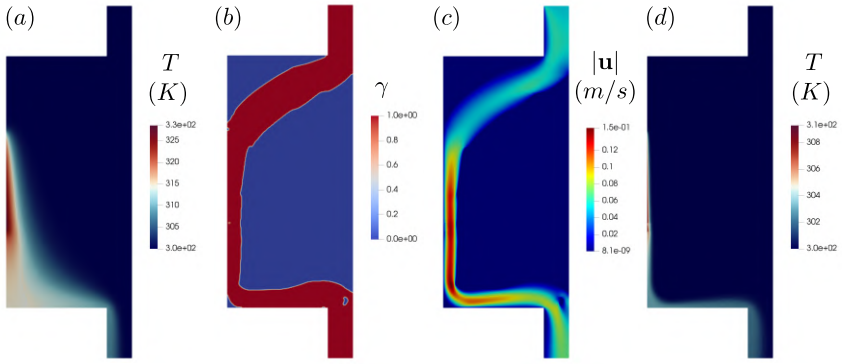

Figure 6.32: Topology optimization of a thermal-fluid flow system with a convective heat flux to minimize the error function. Initial (a) primal temperature field with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (b) pseudodensity, (c) primal velocity and (d) temperature fields.

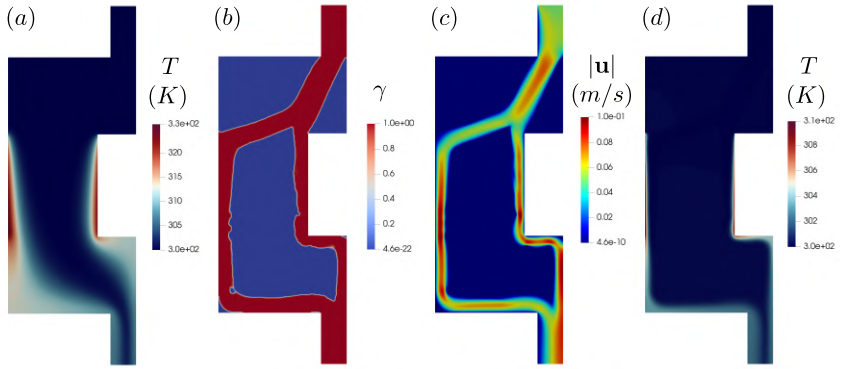

Figure 6.33: Topology optimization of a thermo-fluid flow system with two convective heat flows to minimize the error function. Initial (a) primal temperature field with $\gamma = 0.4$ uniform throughout the domain. Optimal results, (b) pseudo-density, (c) primal velocity and (d) temperature fields.

Figure 6.32 shows the case of a boundary subjected to convective heat transfer. The conduit adopts a final topology such that the fluid accelerates and occupies the region where convective heat transfer occurs, with the goal of improving heat transfer. Figure 6.33 shows the case of two boundaries exposed to convective heat transfer outlined in Fig. 6.21. The result shows a significant decrease in the maximum temperature. The conduit adopts a final topology where the fluid flows past the walls due to the established objective function, which requires a uniform temperature of $300K$ throughout the region.

7
Conclusion and future work

7.1
Conclusions

For the case of the backward facing step configuration in laminar regime and steady state, we demonstrated the equivalence of the continuous and discrete adjoint methods, when tested separately using FEM and FVM. We saw that using FVM and the SIMPLE algorithm to solve the adjoint equations with the same discretization and matrix solution method used to solve the primal equations, local and global convergence problems arose, so appropriate settings must be sought to obtain a convergent solution. However, we have shown that since the adjoint equations are linear, they can be easily solved using FEM.

We performed a sensitivity analysis using the adjoint method with respect to the following parameters: the size and position of the sources, the velocity components of the inlets, and the pseudo-density for the material distribution. Then, the sensitivities were compared with those calculated by the finite difference method and showed good agreement in all cases except for the parameters corresponding to the velocity components of the inlets. The optimal location of discrete sources representing solid regions in the domain was studied. We have presented the derivation of sensitivities for the cost functions of error and total pressure loss with respect to the coordinates of the sources using the continuous adjoint method. When optimizing the position of the sources, we found that the main constraint in optimizing the shape of the inner contour in the domain is to keep the distance between the adjacent sources. Therefore, it is necessary to formulate a condition that allows us to represent an inner impermeable wall while maintaining an appropriate distance between sources.

We could see the importance of knowing the physical phenomenon of the problem in order to formulate objective functions that can be satisfied. For example, the objective function of error in a certain region of the domain is sensitive to some parameters that are not so far from it, and also shows oscillations when the objective function is near the minimum. On the other hand, the objective function for the total pressure loss behaves uniformly since it acts in the whole domain.

In the topology optimization of heat exchangers, the fluid occupies the

hottest regions, aiming for a uniform temperature over the whole domain, as specified by the objective function.

7.2 Suggestions for Future Work

In the following, we address suggestions for future work arising from this thesis:

- To use FEM and FVM together and take their strengths to determine adjoint variables and sensitivities;
- To consider new geometric configurations as well as new objective functions in the study cases presented;
- To assign an individual pseudo-density value $\gamma \in [0, 1]$ to each discrete source in the domain. These pseudo-density values can be related to the inverse of the local permeability (α) by a generic, monotone and continuously differentiable function of γ, similar to topology optimization problems;
- To implement a most robust optimizer like the Method of Moving Asymptotes (MMA) in the case of source position optimization, to improve the optimization process;
- To include heat sources in the case of source position optimization;
- To perform optimization in turbulence regime for active control and detection of hazardous releases using the adjoint method.

8
Bibliography

Aage, N.; Lazarov, B. S. Parallel Framework for Topology Optimization Using the Method of Moving Asymptotes. **Structural and Multidisciplinary Optimization**, v. 47, n. 4, p. 493–505, apr 2013.

Abbas, M. **The simpleFoam Implementation — Explained**. 2020. 1–6 p.

Alexandersen, J.; Andreasen, C. S. A Review of Topology Optimisation for Fluid-Based Problems. **Fluids**, v. 5, n. 1, p. 29, mar 2020.

Alonso, D. H. et al. Topology optimization based on a two-dimensional swirl flow model of tesla-type pump devices. **Computers & Mathematics with Applications**, Elsevier, v. 77, n. 9, p. 2499–2533, 2019.

Andreassen, E. et al. Efficient Topology Optimization in MATLAB Using 88 Lines of Code. **Structural and Multidisciplinary Optimization**, v. 43, n. 1, p. 1–16, jan 2011.

Armaly, B. F. et al. Experimental and theoretical investigation of backward-facing step flow. **Journal of Fluid Mechanics**, v. 127, n. -1, p. 473, feb 1983.

Battista, N. A.; Strickland, W. C.; Miller, L. A. IB2d: A Python and MATLAB Implementation of the Immersed Boundary Method. **Bioinspiration and Biomimetics**, v. 12, n. 3, 2017.

Behrou, R.; Ranjan, R.; Guest, J. K. Adaptive topology optimization for incompressible laminar flow problems with mass flow constraints. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 346, p. 612–641, 2019.

Bendsøe, M. P.; Sigmund, O. **Topology Optimization**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004.

Boffi, D.; Gastaldi, L. The Immersed Boundary Method: A Finite Element Approach. **Computational Fluid and Solid Mechanics 2003**, p. 1263–1266, 2003.

Boffi, D. et al. The finite element immersed boundary method: model, stability, and numerical results. **S. Papadrakakis, Onate, editor, Computational Methods for Coupled Problems in Science and Engineering**, Citeseer, 2005.

Borrvall, T.; Petersson, J. Topology optimization of fluids in Stokes flow. v. 107, n. August 2002, p. 77–107, 2003.

Byrd, R. et al. A Limited Memory Algorithm for Bound Constrained Optimization. **Journal of Scientific Computing**, v. 16, n. 5, p. 1190–1208, 1995.

Cai, S. G. et al. Moving Immersed Boundary Method. **International Journal for Numerical Methods in Fluids**, v. 85, n. 5, p. 288–323, 2017.

Carvalho, M.; Valério, J. **Introdução ao Método de Elementos Finitos: Aplicação em Dinâmica dos Fluidos**. [S.l.]: SBMAC-Sociedade Brasileira de Matemática Aplicada e Computacional, 2012.

Chen, J.; Xia, B.; Zhao, C. Topology Optimization for Heat Transfer Enhancement in Thermochemical Heat Storage. **International Journal of Heat and Mass Transfer**, Elsevier Ltd, v. 154, p. 119785, jun 2020.

Darwish, M.; Moukalled, F. **The Finite Volume Method in Computational Fluid Dynamics: An Advanced Introduction with OpenFOAM® and Matlab®**. [S.l.]: Springer, 2021.

Dede, E. M. Multiphysics topology optimization of heat transfer and fluid flow systems. In: **proceedings of the COMSOL Users Conference**. [S.l.: s.n.], 2009.

Dilgen, S. B. et al. Density based topology optimization of turbulent flow heat transfer systems. **Structural and Multidisciplinary Optimization**, Springer, v. 57, n. 5, p. 1905–1918, 2018.

Dong, X.; Liu, X. Bi-objective topology optimization of asymmetrical fixed-geometry microvalve for non-newtonian flow. **Microsystem Technologies**, Springer, v. 25, n. 6, p. 2471–2479, 2019.

Dong, X.; Liu, X. Multi-objective optimal design of microchannel cooling heat sink using topology optimization method. **Numerical Heat Transfer, Part A: Applications**, Taylor & Francis, v. 77, n. 1, p. 90–104, 2020.

Fanni, M.; Shabara, M.; Alkalla, M. A comparison between different topology optimization methods.(dept m (production)). **MEJ. Mansoura Engineering Journal**, Mansoura University, Faculty of Engineering, v. 38, n. 4, p. 13–24, 2020.

Fuchsberger, J. et al. On the incorporation of obstacles in a fluid flow problem using a navier–stokes–brinkman penalization approach. **Journal of Computational Science**, Elsevier, v. 57, p. 101506, 2022.

Gaymann, A.; Montomoli, F.; Pietropaoli, M. Fluid topology optimization: Bio-inspired valves for aircraft engines. **International Journal of Heat and Fluid Flow**, Elsevier, v. 79, p. 108455, 2019.

Gersborg-hansen, A.; Sigmund, O.; Haber, R. Topology Optimization of Channel Flow Problems. **Structural and Multidisciplinary Optimization**, v. 30, n. 3, p. 181–192, sep 2005.

Goeke, S.; Wünsch, O. Adjoint Based Topology Optimization of Conjugate Heat Transfer Systems. **PAMM**, v. 17, n. 1, p. 771–772, dec 2017.

Goldstein, D.; Handler, R.; Sirovich, L. Modeling a no-slip flow boundary with an external force field. **Journal of computational physics**, Elsevier, v. 105, n. 2, p. 354–366, 1993.

Golub, G. H.; Loan, C. F. V. **Matrix computations**. [S.l.]: JHU press, 2013.

Granzow, B. **A Matlab Implementation of L-BFGS-B**. 2017. 1–10 p.

Gross, L. K. **Essential Mathematical Methods for Physicists**. [S.l.]: JSTOR, 2005.

Gunzburger, M. D. **Perspectives in flow control and optimization**. [S.l.]: SIAM, 2002.

Gunzburger, M. D.; Svobodny, T. P. A numerical Method Suction and Injection for Drag Minimization Via the of Mass Through the Boundary. **In Stabilization of Flexible Structures**, p. 312–321, 1990.

Helgason, E. **Development of Adjoint-Based Optimization Methods for Ducted Flows in Vehicles**. 54 p. Tese (Doutorado), 2015.

Helgason, E.; Krajnović, S. Aerodynamic shape optimization of a pipe using the adjoint method. In: American society of mechanical engineers. **ASME International Mechanical Engineering Congress and Exposition**. [S.l.], 2012. v. 45233, p. 259–267.

Helgason, E.; Krajnović, S. Implementation of an adjoint-based optimization with scalar transport. In: American society of mechanical engineers. **ASME International Mechanical Engineering Congress and Exposition**. [S.l.], 2014. v. 46545, p. V007T09A089.

Herzog, R.; Kunisch, K. Algorithms for PDE-Constrained Optimization. **GAMM-Mitteilungen**, v. 33, n. 2, p. 163–176, 2010.

Hinterberger, C.; Olesen, M. Industrial application of continuous adjoint flow solvers for the optimization of automotive exhaust systems. **CFD & Optimization, Antalya, Turkey**, v. 30, 2011.

Hu, D.; Zhang, Z.; Li, Q. Numerical study on flow and heat transfer characteristics of microchannel designed using topological optimizations method. **Science China Technological Sciences**, Springer, v. 63, n. 1, p. 105–115, 2020.

Joslin, R. D. et al. Self-Contained Automated Methodology for Optimal Flow Control. **AIAA Journal**, v. 35, n. 5, p. 816–824, may 1997.

Kanwal, R. P. **Generalized functions theory and technique: Theory and technique**. [S.l.]: Springer Science & Business Media, 1998.

Karpouzas, G. K. **A Hybrid Method for Shape and Topology Optimization in Fluid Mechanics**. 221 p. Tese (Doutorado), 2019.

Karpouzas, G. K.; Villiers, E. D. Level-set Based Topology Optimization using the continuous adjoint method. In: **An International Conference on Engineering and Applied Sciences Optimization. OPT-I.** [S.l.: s.n.], 2014. p. 4–6.

Kast, S. M. **Methods for Optimal Output Prediction in Computational Fluid Dynamics**. 250 p. Tese (Doutorado), 2016.

Katopodes, N. D., & Piasecki, M. Site and Size Optimization of Contaminant Sources in Surface Water Systems. **Journal of Environmental Engineering**, v. 122, n. October, p. 917–923, 1996.

Kavvadias, I. S. et al. **Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences**. Cham: Springer International Publishing, 2015. v. 36. 159–173 p. (Computational Methods in Applied Sciences, v. 36).

Kenway, G. K. et al. Effective adjoint approaches for computational fluid dynamics. **Progress in Aerospace Sciences**, Elsevier, v. 110, p. 100542, 2019.

Kian, J. D. M. **Topology Optimization Method applied to Design Channels Considering Non-Newtonian Fluid Flow**. 77 p. Tese (Doutorado), 2017.

Koga, A. A. **Projeto de Dispositivos de Microcanais Utilizando o Método de Otimização Topológica**. 134 p. Tese (Doutorado), 2010.

Kontoleontos, E. et al. Adjoint-based constrained topology optimization for viscous flows, including heat transfer. **Engineering Optimization**, Taylor & Francis, v. 45, n. 8, p. 941–961, 2013.

Kopera, M. A. et al. Direct Numerical Simulation of Turbulent Flow Over a Backward-Facing Step. **J. Fluid Mech**, n. 3, p. 329–355, 2014.

Kouhi, M. et al. Implementation of discrete adjoint method for parameter sensitivity analysis in chemically reacting flows. In: 57th aiaa/asce/ahs/asc structures, structural dynamics, and materials .... [S.l.], 2016.

Kutz, J. N. et al. **Dynamic mode decomposition: data-driven modeling of complex systems**. [S.l.]: SIAM, 2016.

Lazarov, B. S.; Sigmund, O. Filters in Topology Optimization Based on Helmholtz-type Differential Equations. **International Journal for Numerical Methods in Engineering**, v. 86, n. 6, p. 765–781, may 2011.

Le, D. V.; Khoo, B. C. A Moving-Least-Square Immersed Boundary Method for Rigid and Deformable Boundaries in Viscous Flow. **Communications in Computational Physics**, v. 22, n. 4, p. 913–934, 2017.

Le, H.; Moin, P.; Kim, J. Direct Numerical Simulation of Turbulent Flow Over a Backward-Facing Step. **Journal of Fluid Mechanics**, v. 330, n. January, p. 349–374, 1997.

Lee, T.; Mateescu, D. Experimental and Numerical Investigation of 2-D Backward-Facing Step Flow. **Journal of Fluids and Structures**, v. 12, n. 6, p. 703–716, aug 1998.

Li, D. et al. An Improved Moving-Least-Squares Reconstruction for Immersed Boundary Method. **International Journal for Numerical Methods in Engineering**, v. 104, n. 8, p. 789–804, 2015.

Li, H. et al. Optimal design and thermal modelling for liquid-cooled heat sink based on multi-objective topology optimization: An experimental and numerical study. **International Journal of Heat and Mass Transfer**, Elsevier, v. 144, p. 118638, 2019.

Liu, K.; Tovar, A. An Efficient 3D Topology Optimization Code Written in Matlab. **Structural and Multidisciplinary Optimization**, v. 50, n. 6, p. 1175–1196, dec 2014.

Liu, W. **Inverse Design of Inverse Design of Enclosed Environment by CFD-Based Adjoint Method**. 182 p. Tese (Doutorado), 2017.

Liu, W. et al. Optimization of Air Supply Location, Size, and Parameters in Enclosed Environments Using a Computational Fluid Dynamics-Based Adjoint method. **Journal of Building Performance Simulation**, v. 9, n. 2, p. 149–161, mar 2016.

Menge, D. et al. Topology optimized heat transfer using the example of an electronic housing. In: University of texas at austin. **2018 International Solid Freeform Fabrication Symposium**. [S.l.], 2018.

Nabi, S.; Grover, P.; Caulfield, C. Adjoint-Based Optimization of Displacement Ventilation Flow. **Building and Environment**, v. 124, p. 342–356, nov 2017.

Nadarajah, S.; Jameson, A. A Comparison of the Continuous and Discrete Adjoint Approach to Automatic Aerodynamic Optimization. In: **38th Aerospace Sciences Meeting and Exhibit**. Reston, Virigina: American Institute of Aeronautics and Astronautics, 2000.

Nadarajah, S. K.; Jameson, A. Optimum Shape Design for Unsteady Flows with Time-Accurate Continuous and Discrete Adjoint Method. **AIAA Journal**, v. 45, n. 7, p. 1478–1491, jul 2007.

Nocedal, J.; Wright, S. J. **Numerical optimization**. [S.l.]: Springer, 1999.

Nogueira, D. S. L. **Topology Optimization Method applied to Laminar Flow Machine Rotor Design**. 110 p. Dissertação (Mestrado), Engenharia de Controle e Automação Mecânica, USP, 2016.

Nørgaard, S. A. et al. Applications of automatic differentiation in topology optimization. **Structural and Multidisciplinary Optimization**, Springer, v. 56, n. 5, p. 1135–1146, 2017.

Openfoam. **Mesh Generation with the BlockMesh Utility**. 2021. 1–11 p.

Openfoam. **OpenFOAM**. 2021.

Othmer, C. A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows. **International Journal for Numerical Methods in Fluids**, v. 58, n. 8, p. 861–877, nov 2008.

Patankar, S. Numerical Heat Transfer and Fluid Flow. Hemishere Publishing Corporation. 1980.

Patankar, S. V. **Heat and Mass Transfer in Turbulent Boundary Layers**. 1–222 p. Tese (Doutorado) — University of London, UK, 1967.

Pereira, A. et al. Fluid Flow Topology Optimization in PolyTop: Stability and Computational Implementation. **Structural and Multidisciplinary Optimization**, Structural and Multidisciplinary Optimization, v. 54, n. 5, p. 1345–1364, 2016.

Persson, P.-O.; Strang, G. A Simple Mesh Generator in MATLAB. **SIAM review**, v. 46, n. June, p. 329–345, 2004.

Peskin, C. S. Flow patterns around heart valves: a numerical method. **Journal of computational physics**, Elsevier, v. 10, n. 2, p. 252–271, 1972.

Pietropaoli, M.; Montomoli, F.; Gaymann, A. Three-dimensional fluid topology optimization for heat transfer. **Structural and Multidisciplinary Optimization**, Structural and Multidisciplinary Optimization, v. 59, n. 3, p. 801–812, mar 2019.

Prieler, R. et al. CFD-Based Optimization of a Transient Heating Process in a Natural Gas Fired Furnace Using Neural Networks and Genetic Algorithms. **Applied Thermal Engineering**, Elsevier, v. 138, n. September 2017, p. 217–234, jun 2018.

Rimer, S. P. **Controlling Hazardous Releases While Protecting Passengers in Civil Infrastructure Systems**. 157 p. Tese (Doutorado), 2016.

Roache, P. J. Code Verification by the Method of Manufactured Solutions. **Journal of Fluids Engineering**, v. 124, n. 1, p. 4–10, mar 2002.

Roy, C. J. et al. Verification of Euler/Navier–Stokes codes using the method of manufactured solutions. **International Journal for Numerical Methods in Fluids**, v. 44, n. 6, p. 599–620, feb 2004.

Ruberto, E. **An Adjoint Based Topology Optimization for Flows Including Heat Transfer**. 95 p. Dissertação (Mestrado), Ingegneria Aeronautica, Politécnico Di Milano, 2017.

Salari, K.; Knupp, P. **Code Verification by the Method of Manufactured Solutions**. Albuquerque, NM (US), 2000. 125 p.

Schneider, R.; Jimack, P. K. On the Evaluation of Finite Element Sensitivities to Nodal Coordinates. **Electronic Transactions on Numerical Analysis**, v. 32, p. 134–144, 2008.

Schramm, M.; Stoevesandt, B.; Peinke, J. Optimization of Airfoils Using the Adjoint Approach and the Influence of Adjoint Turbulent Viscosity. **Computation**, v. 6, n. 1, p. 5, jan 2018.

Sharaborin, E. L.; Rogozin, O. A.; Kasimov, A. R. The coupled volume of fluid and brinkman penalization methods for simulation of incompressible multiphase flows. **Fluids**, MDPI, v. 6, n. 9, p. 334, 2021.

Shih, T. M.; Tan, C. H.; Hwang, B. C. Effects of Grid Staggering on Numerical Schemes. **International Journal for Numerical Methods in Fluids**, v. 9, n. 2, p. 193–212, 1989.

Sigmund, O. A 99 line topology optimization code written in Matlab. **Structural and Multidisciplinary Optimization**, v. 21, n. 2, p. 120–127, apr 2001.

Silva, G. H. et al. Exact and Efficient Interpolation Using Finite Elements Shape Functions. **European Journal of Computational Mechanics**, v. 18, n. 3-4, p. 307–331, jan 2009.

Skaar, K. E. **Topology Optimization for Unsteady Flow with Applications in Biomedical Flows**. 87 p. Tese (Doutorado), 2017.

Skinner, S. N.; Zare-behtash, H. State-of-the-art in aerodynamic shape optimisation methods. **Applied Soft Computing**, Elsevier, v. 62, p. 933–962, 2018.

Soto, O.; Löhner, R.; Yang, C. An Adjoint-Based Design Methodology for CFD Problems. **International Journal of Numerical Methods for Heat & Fluid Flow**, v. 14, n. 6, p. 734–759, sep 2004.

Spietz, H. J.; Hejlesen, M. M.; Walther, J. H. Iterative brinkman penalization for simulation of impulsively started flow past a sphere and a circular disc. **Journal of Computational Physics**, Elsevier, v. 336, p. 261–274, 2017.

Srinath, D.; Mittal, S. An Adjoint Method for Shape Optimization in Unsteady Viscous Flows. **Journal of Computational Physics**, Elsevier Inc., v. 229, n. 6, p. 1994–2008, mar 2010.

Stolpe, M.; Svanberg, K. An Alternative Interpolation Scheme for Minimum Compliance Topology Optimization. **Structural and Multidisciplinary Optimization**, v. 22, n. 2, p. 116–124, sep 2001.

Subramaniam, V.; Dbouk, T.; Harion, J.-L. Topology Optimization of Conductive Heat Transfer Devices: An Experimental Investigation. **Applied Thermal Engineering**, Elsevier Ltd, v. 131, p. 390–411, feb 2018.

Subramaniam, V.; Dbouk, T.; Harion, J.-L. Topology Optimization of Conjugate Heat Transfer Systems: A Competition Between Heat Transfer Enhancement and Pressure Drop Reduction. **International Journal of Heat and Fluid Flow**, Elsevier, v. 75, n. December 2018, p. 165–184, feb 2019.

Sukumar, N.; Tabarraei, A. Conforming Polygonal Finite Elements. **International Journal for Numerical Methods in Engineering**, v. 61, n. 12, p. 2045–2066, nov 2004.

Svanberg, K. The Method of Moving Asymptotes — A New Method for Structural Optimization. **International Journal for Numerical Methods in Engineering**, v. 24, n. 2, p. 359–373, 1987.

Svanberg, K. The Method of Moving Asymptotes - Modelling Aspects and Solution Schemes. **Lecture Notes for the DCAMM course Advanced Topics in Structural Optimization**, p. 1–23, 1998.

Svanberg, K. A Class of Globally Convergent Optimization Methods Based on Conservative Convex Separable Approximations. **SIAM Journal on Optimization**, v. 12, n. 2, p. 555–573, 2002.

Talischi, C. et al. PolyMesher: A General-Purpose Mesh Generator for Polygonal Elements Written in Matlab. **Structural and Multidisciplinary Optimization**, v. 45, n. 3, p. 309–328, mar 2012.

Talischi, C. et al. PolyTop: A Matlab Implementation of a General Topology Optimization Framework Using Unstructured Polygonal Finite Element Meshes. **Structural and Multidisciplinary Optimization**, v. 45, n. 3, p. 329–357, mar 2012.

Tawk, R.; Ghannam, B.; Nemer, M. Topology optimization of heat and mass transfer problems in two fluids—one solid domains. **Numerical Heat Transfer, Part B: Fundamentals**, Taylor & Francis, v. 76, n. 3, p. 130–151, 2019.

Tran, S. B. et al. Modeling Deformable Capsules in Viscous Flow Using Immersed Boundary Method. **Physics of Fluids**, v. 32, n. 9, 2020.

Versteeg, H. K.; Malalasekera, W. **An introduction to computational fluid dynamics: the finite volume method**. [S.l.]: Pearson education, 2007.

Wachspress, E. L. **A Rational Finite Element Basis**. [S.l.: s.n.], 1975. v. 114. 347 p.

Wang, B. **Real-time Control of Solute Plume in Closed Conduit Flow**. 187 p. Tese (Doutorado), 2015.

Wang, L.; Mavriplis, D. J.; Anderson, W. K. Adjoint Sensitivity Formulation for Discontinuous Galerkin Discretizations in Unsteady Inviscid Flow Problems. **AIAA Journal**, v. 48, n. 12, p. 2867–2883, dec 2010.

Warnock, A. M. **Automatic Detection and Control of Hazardous Plumes in Wall-Bounded Flow Systems**. 288 p. Tese (Doutorado), 2013.

Xue, Y.; Zhai, Z. J.; Chen, Q. Inverse Prediction and Optimization of Flow Control Conditions for Confined Spaces Using a CFD-Based Genetic Algorithm. **Building and Environment**, Elsevier Ltd, v. 64, p. 77–84, 2013.

Yamaleev, N. K.; Diskin, B.; Nielsen, E. J. Local-in-Time Adjoint-Based Method for Design Optimization of Unsteady Flows. **Journal of Computational Physics**, Elsevier Inc., v. 229, n. 14, p. 5394–5407, jul 2010.

Yoon, G. H. Topology optimization for turbulent flow with spalart–allmaras model. **Computer Methods in Applied Mechanics and Engineering**, Elsevier, v. 303, p. 288–311, 2016.

Yoon, M.; Koo, B. Topology Design Optimization of Conductive Thermal Problems Subject to Design-Dependent Load Using Density Gradients. **Advances in Mechanical Engineering**, v. 11, n. 5, may 2019.

Yu, M. et al. Three-dimensional Topology Optimization of Thermal-Fluid-Structural Problems for Cooling System Design. **Structural and Multidisciplinary Optimization**, Structural and Multidisciplinary Optimization, v. 62, n. 6, p. 3347–3366, 2020.

Zhao, X. et al. Optimal Design of an Indoor Environment by the CFD-Based Adjoint Method with Area-Constrained Topology and Cluster Analysis. **Building and Environment**, Elsevier, v. 138, n. February, p. 171–180, jun 2018.

Zhao, X. et al. Inverse Design of an Indoor Environment Using a CFD-Based Adjoint Method with the Adaptive Step Size for Adjusting the Design Parameters. **Numerical Heat Transfer, Part A: Applications**, Taylor & Francis, v. 71, n. 7, p. 707–720, apr 2017.

A
The Finite Element Method Discretization

In this section we present the finite element discretization procedure of the primal and adjoint equations. For more details on the theory and implementation of the Matlab base code used in this work, see (Carvalho; Valério, 2012). The finite element method is based on the weak formulation of the PDE, where an approximate solution is sought in a suitable subspace of finite dimension. The approximate solution is a projection of the exact solution into this subspace and can be represented as a linear combination of a finite number of known functions forming the basis of this subspace.

A.1
FEM of Primal Equations

First, in this section, for simplicity of notation, we assume that the variables $\boldsymbol{u}, p$ and ϕ that appear in the governing equations $(2\text{-}3)$ are variables that approximate the exact solution. Thus, applying the method of weighted residuals to force the residuals of the governing equations $(2\text{-}3)$ to be equal to zero over the entire range Ω yields,

$$\mathbf{R}_m = \int_\Omega [-\nabla \cdot \mathbf{T} + \boldsymbol{u} \cdot \nabla \boldsymbol{u} - \boldsymbol{f}_{\Omega_a}] \cdot \mathbf{W} \, \mathrm{d}\Omega = 0 \tag{A-1}$$

$$R_c = \int_\Omega [\nabla \cdot \boldsymbol{u}] \zeta \, \mathrm{d}\Omega = 0 \tag{A-2}$$

$$R_s = \int_\Omega [-\nabla \cdot (D \nabla \phi) + \boldsymbol{u} \cdot \nabla \phi - f_{\Omega_a}] \omega \, \mathrm{d}\Omega = 0 \tag{A-3}$$

where $\mathbf{W}$, ζ and ω are the weight functions.

A.1.1
Weak formulation

To obtain the weak form of the above equations, integration by parts is used to distribute the differential operators equally between the dependent variables and the weight functions. Recall, then, that the weak form of the Eqs. A-1, A-4 and A-3 consists of $(\boldsymbol{u}, p, \phi) \in \mathcal{U} \times \mathcal{Q} \times \mathcal{S}$ such that:

$$\begin{aligned}\mathbf{R}_m = &\int_\Omega \mathbf{T} : \nabla \mathbf{W} \, \mathrm{d}\Omega - \int_\Gamma (\boldsymbol{n} \cdot \mathbf{T}) \cdot \mathbf{W} \, \mathrm{d}\Gamma + \int_\Omega (\boldsymbol{u} \cdot \nabla \boldsymbol{u}) \cdot \mathbf{W} \, \mathrm{d}\Omega \\ &- \int_\Omega \boldsymbol{f}_{\Omega_a} \cdot \mathbf{W} \, \mathrm{d}\Omega \end{aligned} \tag{A-4}$$

$$\begin{aligned} R_c = &\int_\Omega [\nabla \cdot \boldsymbol{u}] \zeta \, \mathrm{d}\Omega = 0 \\ R_s = &\int_\Omega \nabla \omega \cdot (D \nabla \phi) \, \mathrm{d}\Omega - \int_\Gamma \boldsymbol{n} \cdot (\omega D \nabla \phi) \, \mathrm{d}\Gamma + \int_\Omega \boldsymbol{u} \cdot \nabla \phi \omega \, \mathrm{d}\Omega \\ &- \int_{\Omega_a} f \omega \, \mathrm{d}\Omega \end{aligned} \tag{A-5}$$

where the velocity trial and test spaces are given by:

$$\mathcal{U} = \left\{ \boldsymbol{u} \in \left[H^1(\Omega) \right]^2 : \boldsymbol{u}|_{\Gamma_D} = \mathbf{g} \right\}, \tag{A-6}$$

$$\mathcal{W} = \left\{ \mathbf{W} \in \left[H^1(\Omega) \right]^2 : \mathbf{W}|_{\Gamma_D} = \mathbf{0} \right\} \tag{A-7}$$

The pressure space is $\mathcal{Q} = L^2(\Omega)$, and the scalar trial and test spaces are given by:

$$\mathcal{S} = \left\{ \phi \in \left[H^1(\Omega) \right] : \phi|_{\Gamma_D} = \mathrm{g} \right\}, \tag{A-8}$$

$$\mathcal{T} = \left\{ \omega \in \left[H^1(\Omega) \right] : \omega|_{\Gamma_D} = 0 \right\} \tag{A-9}$$

Given a partition $\mathcal{P}_h = \{\Omega_l\}_{l=1}^M$ of Ω consisting of elements of the mesh domain, the approximation of this system is obtained by restricting the variational problem to finite dimensional subspaces $\mathcal{U}_h$, $\mathcal{W}_h$, $\mathcal{S}_h$, $\mathcal{T}_h$ and $\mathcal{Q}_h$.
The vector weight function $\mathbf{W}$ and the velocity $\boldsymbol{u}$ can be written in terms of their components: $\mathbf{W} = [W_1, W_2]$ and $\boldsymbol{u} = [u, v]$, respectively. $\mathbf{W}$ is a vector function belonging to the function space generated by the following weight functions:

$$\left\langle \begin{bmatrix} \psi_1 \\ 0 \end{bmatrix}, \begin{bmatrix} \psi_2 \\ 0 \end{bmatrix}, \cdots, \begin{bmatrix} \psi_N \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ \psi_1 \end{bmatrix}, \begin{bmatrix} 0 \\ \psi_2 \end{bmatrix}, \cdots, \begin{bmatrix} 0 \\ \psi_N \end{bmatrix} \right\rangle \tag{A-10}$$

Moreover, ζ and ω are scalar functions that belongs to the function space generated by the following weight functions:

$$\zeta = \{\chi_1, \chi_2, \cdots \chi_M\}, \tag{A-11}$$

$$\omega = \{\psi_1, \psi_2, \cdots \psi_N\} \tag{A-12}$$

where M means the number of mesh elements and N the number of mesh nodes.

A.1.2
Two dimensional case

In the special case of the Cartesian two-dimensional system, each weighted remainder term of the momentum equation is written as follows:

1. $\mathbf{T} : \nabla \mathbf{W} = \dfrac{\partial W_1}{\partial x}\Big(-\dfrac{p}{\rho} + 2\nu\dfrac{\partial u}{\partial x}\Big) + \dfrac{\partial W_1}{\partial y}\Big[\nu\Big(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\Big)\Big]$
$+\dfrac{\partial W_2}{\partial y}\Big(-\dfrac{p}{\rho} + 2\nu\dfrac{\partial v}{\partial y}\Big) + \dfrac{\partial W_2}{\partial x}\Big[\nu\Big(\dfrac{\partial u}{\partial y} + \dfrac{\partial v}{\partial x}\Big)\Big]$

2. $(n \cdot \mathbf{T}) \cdot \mathbf{W} = f_x W_1 + f_y W_2$

3. $(\boldsymbol{u} \cdot \nabla \boldsymbol{u}) \cdot \mathbf{W} = W_1\Big(u\dfrac{\partial u}{\partial x} + \nu\dfrac{\partial u}{\partial y}\Big) + W_2\Big(u\dfrac{\partial v}{\partial x} + \nu\dfrac{\partial v}{\partial y}\Big)$

Each component of the weight vector function can be written as a linear combination of the scalar basis functions ψ_i. Thus, we can split $\mathbf{R}_m$ into R^i_{mx} and R^i_{my}.

$$
\begin{aligned}
R^i_{mx} = & \int_\Omega \Big\{\psi_i\Big(u\frac{\partial u}{\partial x} + \nu\frac{\partial u}{\partial y}\Big) + \frac{\partial \psi_i}{\partial x}\Big(-\frac{p}{\rho} + 2\nu\frac{\partial u}{\partial x}\Big) + \frac{\partial \psi_i}{\partial y}\Big[\nu\Big(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\Big)\Big]\Big\}\,\mathrm{d}\Omega \\
& - \int_\Gamma \psi_i\Big[n_x\Big(-\frac{p}{\rho} + 2\nu\frac{\partial u}{\partial x}\Big) + n_y\nu\Big(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\Big)\Big]\,\mathrm{d}\Gamma - \int_{\Omega_a} \psi_i f_{\Omega x}\,\mathrm{d}\Gamma \\
& i = 1, \cdots, N.
\end{aligned}
\tag{A-13}
$$

$$
\begin{aligned}
R^i_{my} = & \int_\Omega \Big\{\psi_i\Big(u\frac{\partial v}{\partial x} + \nu\frac{\partial v}{\partial y}\Big) + \frac{\partial \psi_i}{\partial y}\Big(-\frac{p}{\rho} + 2\nu\frac{\partial v}{\partial x}\Big) + \frac{\partial \psi_i}{\partial x}\Big[\nu\Big(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\Big)\Big]\Big\}\,\mathrm{d}\Omega \\
& - \int_\Gamma \psi_i\Big[n_y\Big(-\frac{p}{\rho} + 2\nu\frac{\partial v}{\partial y}\Big) + n_x\nu\Big(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\Big)\Big]\,\mathrm{d}\Gamma - \int_{\Omega_a} \psi_i f_{\Omega y}\,\mathrm{d}\Gamma \\
& i = 1, \cdots, N.
\end{aligned}
\tag{A-14}
$$

The continuity equation results

$$
R^i_c = \int_\Omega \Big(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y}\Big)\chi_i\,\mathrm{d}\Omega = 0; \qquad i = 1, \cdots, M. \tag{A-15}
$$

and the scalar equation gives

$$
\begin{aligned}
R^i_s = & \int_\Omega \Big[\psi_i\Big(u\frac{\partial \phi}{\partial x} + \nu\frac{\partial \phi}{\partial y}\Big) + \frac{\partial \psi_i}{\partial x}\Big(D\frac{\partial \phi}{\partial x}\Big) + \frac{\partial \psi_i}{\partial y}\Big(D\frac{\partial \phi}{\partial x}\Big)\Big]\,\mathrm{d}\Omega \\
& - \int_\Gamma \psi_i\Big(n_x D\frac{\partial \phi}{\partial x} + n_y\frac{\partial \phi}{\partial y}\Big)\,\mathrm{d}\Gamma - \int_{\Omega_a} f\psi_i\,\mathrm{d}\Omega; \qquad i = 1, \cdots, N.
\end{aligned}
\tag{A-16}
$$

A.1.3
Expansion of unknown fields

Within each element, we assume a constant pressure field, while the velocity field is represented by an isoparametric transformation of the Wachspress basis. Note that with this choice, each $p \in \mathcal{Q}_h$ can be written as a linear extension of the basis function $\chi_i \in \mathcal{Q}_h$.

$$p = \sum_{j=1}^{M} P_j \chi_j \tag{A-17}$$

Suppose that $\boldsymbol{u} \in \mathcal{U}_h$ and $\phi \in \mathcal{S}_h$ have a linear evolution of the basis function $\psi_i \in \mathcal{U}_h$ and $\psi_i \in \mathcal{S}_h$, respectively.

$$\boldsymbol{u} = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{N} U_j \psi_j \\ \sum_{j=1}^{N} V_j \psi_j \end{bmatrix} \tag{A-18}$$

$$\phi = \sum_{j=1}^{N} \Phi_j \psi_j \tag{A-19}$$

A.1.4
Newton's method

To obtain a second-order convergence rate, Newton's method is used to solve the nonlinear coupled system, and it is explained in Algorithm 1.

Algorithm 1: Newton Method

```
Set the residual R(c) = 0; where c = (U, V, P, Φ)^T
Choose an initial attempt c = c_0;
while ||R(c)|| > ε do
    J Δc = −R
    Update c;
    c = c + Δc
end
```

The Jacobian matrix $\mathbf{J}$ of Newton's method is given by:

$$\mathbf{J} = \begin{bmatrix} \dfrac{\partial R^i_{mx}}{\partial U_j} & \dfrac{\partial R^i_{mx}}{\partial V_j} & \dfrac{\partial R^i_{mx}}{\partial P_j} & \dfrac{\partial R^i_{mx}}{\partial \Phi_j} \\ \dfrac{\partial R^i_{my}}{\partial U_j} & \dfrac{\partial R^i_{my}}{\partial V_j} & \dfrac{\partial R^i_{my}}{\partial P_j} & \dfrac{\partial R^i_{my}}{\partial \Phi_j} \\ \dfrac{\partial R^i_{c}}{\partial U_j} & \dfrac{\partial R^i_{c}}{\partial V_j} & \dfrac{\partial R^i_{c}}{\partial P_j} & \dfrac{\partial R^i_{c}}{\partial \Phi_j} \\ \dfrac{\partial R^i_{s}}{\partial U_j} & \dfrac{\partial R^i_{s}}{\partial V_j} & \dfrac{\partial R^i_{s}}{\partial P_j} & \dfrac{\partial R^i_{s}}{\partial \Phi_j} \end{bmatrix}$$

The expression for each component of this structure is presented as follows:
Components for R^i_{mx}

$$\begin{aligned}
\frac{\partial R^i_{mx}}{\partial U_j} &= \int_\Omega \left\{ \psi_i \Big(\psi_i \frac{\partial u}{\partial x} + u \frac{\partial \psi_i}{\partial x} + v \frac{\partial \psi_i}{\partial y} \Big) + \frac{\partial \psi_i}{\partial x} 2\nu \frac{\partial \psi_j}{\partial x} + \nu \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial y} \right\} \mathrm{d}\Omega \\
&\quad - \int_\Gamma \psi_i \Big[n_x \Big(2\nu \frac{\partial \psi_j}{\partial x} \Big) + n_y \nu \Big(\frac{\partial \psi_j}{\partial y} \Big) \Big] \mathrm{d}\Gamma \\
\frac{\partial R^i_{mx}}{\partial V_j} &= \int_\Omega \Big(\psi_i \psi_j \frac{\partial u}{\partial y} + \nu \frac{\partial \psi_i}{\partial y} \frac{\partial \psi_j}{\partial x} \, \mathrm{d}\Omega - \int_\Gamma \psi_i \Big[n_y \nu \Big(\frac{\partial \psi_j}{\partial x} \Big) \Big] \mathrm{d}\Gamma \\
\frac{\partial R^i_{mx}}{\partial P_j} &= \int_\Omega \Big(- \frac{\partial \psi_i}{\partial x} \chi_j \Big) \mathrm{d}\Omega \\
\frac{\partial R^i_{mx}}{\partial \Phi_j} &= 0
\end{aligned}$$

Components for R^i_{my}

$$\begin{aligned}
\frac{\partial R^i_{my}}{\partial U_j} &= \int_\Omega \Big(\psi_i \psi_j \frac{\partial v}{\partial x} + \nu \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial y} \Big) \mathrm{d}\Omega - \int_\Gamma \psi_i \Big[n_x \nu \Big(\frac{\partial \psi_j}{\partial x} \Big) \Big] \mathrm{d}\Gamma \\
\frac{\partial R^i_{my}}{\partial V_j} &= \int_\Omega \Big[\psi_i \Big(u \frac{\partial \psi_j}{\partial x} + \psi_j \frac{\partial v}{\partial y} + v \frac{\partial \psi_j}{\partial y} \Big) + \frac{\partial \psi_j}{\partial y} 2\nu \frac{\partial \psi_j}{\partial y} + \nu \frac{\partial \psi_i}{\partial x} \frac{\partial \psi_j}{\partial x} \Big] \mathrm{d}\Omega \\
&\quad - \int_\Gamma \psi_i \Big[n_y \Big(2\nu \frac{\partial \psi_j}{\partial y} \Big) + n_x \nu \Big(\frac{\partial \psi_j}{\partial x} \Big) \Big] \mathrm{d}\Gamma \\
\frac{\partial R^i_{my}}{\partial P_j} &= \int_\Omega \Big(- \frac{\partial \psi_i}{\partial y} \chi_j \Big) \mathrm{d}\Omega \\
\frac{\partial R^i_{my}}{\partial \Phi_j} &= 0
\end{aligned}$$

Components for R^i_c

$$\frac{\partial R_c^i}{\partial U_j} = \int_\Omega \left(\frac{\partial \psi_j}{\partial x}\chi_i\right) \mathrm{d}\Omega$$
$$\frac{\partial R_c^i}{\partial V_j} = \int_\Omega \left(\frac{\partial \psi_j}{\partial y}\chi_i\right) \mathrm{d}\Omega$$
$$\frac{\partial R_c^i}{\partial P_j} = 0$$
$$\frac{\partial R_c^i}{\partial \Phi_j} = 0$$

Components for R_s^i

$$\frac{\partial R_s^i}{\partial U_j} = \int_\Omega \left(\psi_i \psi_j \frac{\partial \phi}{\partial x}\right) \mathrm{d}\Omega$$
$$\frac{\partial R_s^i}{\partial V_j} = \int_\Omega \left(\psi_i \psi_j \frac{\partial \phi}{\partial y}\right) \mathrm{d}\Omega$$
$$\frac{\partial R_s^i}{\partial P_j} = 0$$
$$\frac{\partial R_s^i}{\partial \Phi_j} = \int_\Omega \left[\psi_i\left(u\frac{\partial \psi_j}{\partial x} + v\frac{\partial \psi_j}{\partial y}\right) + \frac{\partial \psi_j}{\partial x}\left(D\frac{\partial \psi_j}{\partial x}\right) + \frac{\partial \psi_j}{\partial y}\left(D\frac{\partial \psi_j}{\partial x}\right)\right] \mathrm{d}\Omega - \int_\Gamma \psi_i\left(n_x D\frac{\partial \psi_j}{\partial x} + n_y D\frac{\partial \psi_j}{\partial y}\right) \mathrm{d}\Gamma$$

Boundary Conditions: The boundary conditions are implemented as follows:

Dirichlet boundary condition: We specify a value for the state variables (U_i, V_i, Φ_i) with reference values (U_r, V_r, Φ_r), on the boundaries e.g. Γ_i, Γ_a and Γ_ω. Furthermore, the element Jacobian matrix turns to the identity matrix, like this:

$$R_{mx}^i = U_i - U_r,$$
$$R_{my}^i = V_i - V_r,$$
$$R_s^i = \Phi_i - \Phi_r,$$
$$J^{(e)}(iu, 1:ndof) = 0,$$
$$J^{(e)}(iv, 1:ndof) = 0,$$
$$J^{(e)}(i\phi, 1:ndof) = 0,$$
$$J^{(e)}(iu, iu) = 1,$$
$$J^{(e)}(iv, iv) = 1,$$
$$J^{(e)}(i\phi, i\phi) = 1$$

where iu,iv and $i\phi$ represent the row position on the element matrix of the state variables u,v and ϕ, respectively, and $ndof$ is the number of degrees of freedom of each element. Note that on these boundaries the normal pressure gradient is generally zero .

Neumann boundary condition: We specify the value of the momentum and scalar flux at the boundary, e.g., Γ_o. For evolved flows, for example, the flux is zero, and the value of the pressure P should be given.

Symmetry boundary condition: Sometimes we are dealing with symmetry flows. For example, in a horizontal flow where $\boldsymbol{n} \cdot \boldsymbol{u} = v = 0$ (i.e. $R^i_{my} = V_i - 0$) and the shear stress $\boldsymbol{t} \cdot (\boldsymbol{n} \cdot \mathbf{T}) = 0$.

A.1.5 Shape functions

We use the shape function following the work presented by Talischi et al. (2012). The Wachspress shape function corresponding to node i, $1 \leq i \leq n$, is defined as (Sukumar; Tabarraei, 2004):

$$\psi_i(\boldsymbol{\xi}) = \frac{\alpha_i(\boldsymbol{\xi})}{\sum_{j=1}^{n} \alpha_j(\boldsymbol{\xi})} \tag{A-20}$$

where α_i are the interpolants of the form:

$$\alpha_i(\boldsymbol{\xi}) = \frac{A(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})}{A(\mathbf{p}_{i-1}, \mathbf{p}_i, \boldsymbol{\xi}) A(\mathbf{p}_i, \mathbf{p}_{i+1}, \boldsymbol{\xi})} \tag{A-21}$$

Here A denotes the area of the triangle formed by its arguments (see Fig. A.1a). Since the reference element is a regular polygon, $A(\mathbf{p}_{i-1}, \mathbf{p}_i, \mathbf{p}_{i+1})$ is the same for all i and therefore can be extracted from the expression Eq. (A-20). If we adopt the notation $A_i(\boldsymbol{\xi}) = A(\mathbf{p}_{i-1}, \mathbf{p}_i, \boldsymbol{\xi})$, we obtain the simplified formula for interpolation:

$$\alpha_i(\boldsymbol{\xi}) = \frac{1}{A_i(\boldsymbol{\xi}) A_{i+1}(\boldsymbol{\xi})} \tag{A-22}$$

The area of the triangles formed by $\boldsymbol{\xi}$ and the vertices, as well as their gradient with respect to it, are calculated using expressions:

$$A_i(\boldsymbol{\xi}) = \frac{1}{2} \begin{vmatrix} \xi_1 & \xi_2 & 1 \\ p_{1,i-1} & p_{2,i-1} & 1 \\ p_{1,i} & p_{2,i} & 1 \end{vmatrix}, \tag{A-23}$$

$$\frac{\partial A_i}{\partial \xi_1} = \frac{1}{2}(p_{2,i-1} - p_{2,i}); \; \frac{\partial A_i}{\partial \xi_2} = \frac{1}{2}(p_{1,i} - p_{1,i-1}) \tag{A-24}$$

The derivatives of the interpolant are simply given by

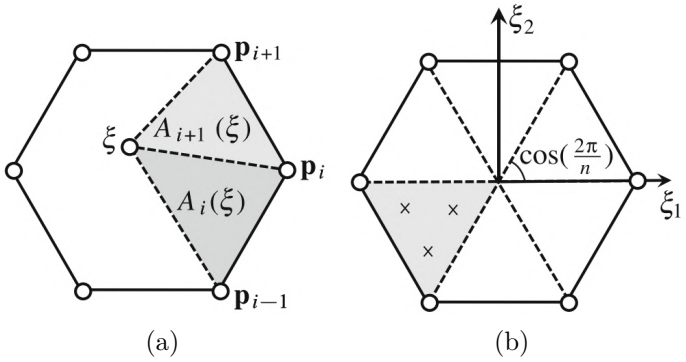

Figure A.1: (a) Triangular areas used to define α_i, (b) Triangulation of the reference regular polygon and integration points defined on each triangle (Talischi et al., 2012b).

$$\frac{\partial \alpha_i}{\partial \xi_k} = -\alpha_i \left(\frac{1}{A_i} \frac{\partial A_i}{\partial \xi_k} + \frac{1}{A_{i+1}} \frac{\partial A_{i+1}}{\partial \xi_k} \right);\ k = 1, 2 \tag{A-25}$$

and from Eq. (A-20) we have the following expression for the shape function gradients:

$$\frac{\partial \psi_i}{\partial \xi_k} = \frac{1}{\sum_{j=1}^{n} \alpha_j} \left(\frac{\partial \alpha_i}{\partial \xi_k} - \psi_i \sum_{j=1}^{n} \frac{\partial \alpha_j}{\partial \xi_k} \right);\ k = 1, 2 \tag{A-26}$$

A directed triangulation of the reference n-gon is done by connecting its vertices with the input point $\boldsymbol{\xi}$, which lies in its interior. As shown in Fig. A.1b, the vertices of the reference n-gon are located at $\mathbf{p}_i = (cos\, 2\pi i/n, sin\, 2\pi i/n)$. This function is used in the definition of polygonal shape functions as well as in the quadrature rule.

One way to perform integration over the n-gon reference is to divide it into n-triangles (by connecting the origin to the vertices) and apply known quadrature rules to each triangle. We used three integration points per triangle (see Fig. A.1b).

A.1.6 Numerical Integration

The isoparametric mapping is used to transform the global coordinates (x, y) into local coordinates (ξ_1, ξ_2) so that the integrals along the element are written using the Gaussian quadrature method as follows:

$$\int_{\Omega} F(x, y)\, \mathrm{d}x\, \mathrm{d}y = \int_{\overline{\Omega}} \overline{F}(\xi_1, \xi_2) |\mathbf{J}|\, \mathrm{d}\xi_1\, \mathrm{d}\xi_2 = \sum_{k=1}^{ngp} \overline{F}(\xi_{1k}, \xi_{2k}) |\mathbf{J}^k| W_k \tag{A-27}$$

where ngp is the number of integration points in the element and $|\mathbf{J}^k|$ is the Jacobian of the transformation evaluated at each integration point k.

A.2
FEM of Adjoint Equations

Again, we use the weighted residue method to enforce that the residue of the adjoint equations $(2\text{-}52)(2\text{-}53)(2\text{-}54)$, over the entire domain Ω, is zero.

$$\mathbf{R}_{ma} = \int_{\Omega} \Big[-\nabla\cdot\widehat{\mathbf{T}} - \nabla\boldsymbol{v}\cdot\boldsymbol{u} - (\boldsymbol{u}\cdot\nabla)\boldsymbol{v} - \nabla\varphi\phi - \frac{\partial\mathcal{J}_{\Omega}}{\partial\boldsymbol{u}}\Big|_{\Omega_s}\Big]\cdot\mathbf{W}\,\mathrm{d}\Omega = 0 \tag{A-28}$$

$$R_{ca} = \int_{\Omega} \Big[-\nabla\cdot\boldsymbol{v} - \frac{\partial\mathcal{J}_{\Omega}}{\partial p}\Big|_{\Omega_s}\Big]\zeta\,\mathrm{d}\Omega = 0 \tag{A-29}$$

$$R_{sa} = \int_{\Omega} \Big[-\nabla\cdot(D\nabla\varphi) - \nabla\cdot(\boldsymbol{u}\varphi) - \frac{\partial\mathcal{J}_{\Omega}}{\partial\phi}\Big|_{\Omega_s}\Big]\omega\,\mathrm{d}\Omega = 0 \tag{A-30}$$

where $\mathbf{W}$, ζ and ω are the weight functions.

A.2.1
Weak formulation

The weak form of the above adjoint equations is obtained by integration by parts to distribute the differential operators equally between the dependent variables and the weight functions.

$$\begin{aligned}\mathbf{R}_{ma} = & \int_{\Omega}\widehat{\mathbf{T}} : \nabla\mathbf{W}\,\mathrm{d}\Omega - \int_{\Gamma}(\boldsymbol{n}\cdot\widehat{\mathbf{T}})\cdot\mathbf{W}\,\mathrm{d}\Gamma - \int_{\Omega}\boldsymbol{u}\cdot[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathsf{T}}]\cdot\mathbf{W}\,\mathrm{d}\Omega \\ & - \int_{\Omega}\nabla\varphi\phi\cdot\mathbf{W}\,\mathrm{d}\Omega - \int_{\Omega}\frac{\partial\mathcal{J}_{\Omega}}{\partial\boldsymbol{u}}\Big|_{\Omega_s}\cdot\mathbf{W}\,\mathrm{d}\Omega\end{aligned} \tag{A-31}$$

$$\begin{aligned}R_{sa} = & \int_{\Omega}\nabla\omega\cdot(D\nabla\varphi)\,\mathrm{d}\Omega - \int_{\Gamma}\boldsymbol{n}\cdot(\omega D\nabla\varphi)\,\mathrm{d}\Gamma - \int_{\Omega}\nabla\cdot(\boldsymbol{u}\varphi)\omega\,\mathrm{d}\Omega \\ & - \int_{\Omega_s}\frac{\partial\mathcal{J}_{\Omega}}{\partial\phi}\omega\,\mathrm{d}\Omega\end{aligned} \tag{A-32}$$

Substituting the adjoint boundary conditions,

$$\begin{aligned}\mathbf{R}_{ma} = & \int_{\Omega}\widehat{\mathbf{T}} : \nabla\mathbf{W}\,\mathrm{d}\Omega - \int_{\Gamma_o}\Big[\frac{\partial\mathcal{J}_{\Gamma}}{\partial\boldsymbol{u}}\Big|_{\Gamma_s} - \boldsymbol{v}(\boldsymbol{u}\cdot\boldsymbol{n}) - \boldsymbol{n}(\boldsymbol{v}\cdot\boldsymbol{u}) - \varphi\phi\boldsymbol{n}\Big]\cdot\mathbf{W}\,\mathrm{d}\Gamma \\ & - \int_{\Omega}\boldsymbol{u}\cdot[\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathsf{T}}]\cdot\mathbf{W}\,\mathrm{d}\Omega - \int_{\Omega}\nabla\varphi\phi\cdot\mathbf{W}\,\mathrm{d}\Omega \\ & - \int_{\Omega}\frac{\partial\mathcal{J}_{\Omega}}{\partial\boldsymbol{u}}\Big|_{\Omega_s}\cdot\mathbf{W}\,\mathrm{d}\Omega\end{aligned} \tag{A-33}$$

$$R_{sa} = \int_{\Omega} \nabla\omega \cdot (D\nabla\varphi)\,\mathrm{d}\Omega - \int_{\Gamma_{\omega s}} \frac{\partial \mathcal{J}_{\Gamma}}{\partial \phi}\omega\,\mathrm{d}\Gamma - \int_{\Gamma_i - \Gamma_o} \omega\Big(\frac{\partial \mathcal{J}_{\Gamma}}{\partial \phi}\Big|_{\Gamma_s} - \boldsymbol{u}\cdot\boldsymbol{n}\varphi\Big)\,\mathrm{d}\Gamma$$
$$- \int_{\Omega} \nabla\cdot(\boldsymbol{u}\varphi)\omega\,\mathrm{d}\Omega - \int_{\Omega_s} \frac{\partial \mathcal{J}_{\Omega}}{\partial \phi}\omega\,\mathrm{d}\Omega \tag{A-34}$$

A.2.2
Two dimensional case

The vectorial weight function $\mathbf{W}$ and the adjoint velocity $\boldsymbol{v}$ can be written in terms of its components: $\mathbf{W} = [W_1, W_2]$ and $\boldsymbol{v} = [\hat{u}, \hat{v}]$ respectively. Hence, we have

1. $\widehat{\mathbf{T}} : \nabla\mathbf{W} = \dfrac{\partial W_1}{\partial x}\Big(-\dfrac{q}{\rho} + 2\nu\dfrac{\partial \hat{u}}{\partial x}\Big) + \dfrac{\partial W_1}{\partial y}\Big[\nu\Big(\dfrac{\partial \hat{u}}{\partial y} + \dfrac{\partial \hat{v}}{\partial x}\Big)\Big]$
$+\dfrac{\partial W_2}{\partial y}\Big(-\dfrac{q}{\rho} + 2\nu\dfrac{\partial \hat{v}}{\partial y}\Big) + \dfrac{\partial W_2}{\partial x}\Big[\nu\Big(\dfrac{\partial \hat{u}}{\partial y} + \dfrac{\partial \hat{v}}{\partial x}\Big)\Big]$

2. $(n\cdot\widehat{\mathbf{T}})\cdot\mathbf{W} = \Big[\dfrac{\partial\Gamma}{\partial u}\Big|_{\Gamma_s} - 2n_x\hat{u}u - v(n_y\hat{u} + n_x\hat{v}) - \varphi\phi n_x\Big]W_1$
$+\Big[\dfrac{\partial\Gamma}{\partial v}\Big|_{\Gamma_s} - 2n_y\hat{v}v - u(n_y\hat{u} + n_x\hat{v}) - \varphi\phi n_y\Big]W_2$

3. $\boldsymbol{u}\cdot[\nabla v + (\nabla v)^{\mathsf{T}}]\cdot\mathbf{W} = W_1\Big[2u\dfrac{\partial \hat{u}}{\partial x} + v\Big(\dfrac{\partial \hat{u}}{\partial y} + \dfrac{\partial \hat{v}}{\partial x}\Big)\Big] + W_2\Big[u\Big(\dfrac{\partial \hat{u}}{\partial y} + \dfrac{\partial \hat{v}}{\partial x}\Big) + 2v\dfrac{\partial \hat{v}}{\partial y}\Big]$

4. $\nabla\varphi\phi\cdot\mathbf{W} = W_1\Big(\phi\dfrac{\partial\varphi}{\partial x}\Big) + W_2\Big(\phi\dfrac{\partial\varphi}{\partial y}\Big)$

5. $\dfrac{\partial \mathcal{J}_{\Omega}}{\partial u}\Big|_{\Omega_s}\cdot\mathbf{W} = W_1\Big(\dfrac{\partial \mathcal{J}_{\Omega}}{\partial u}\Big|_{\Omega_s}\Big) + W_2\Big(\dfrac{\partial \mathcal{J}_{\Omega}}{\partial v}\Big|_{\Omega_s}\Big)$

Each component of the weight vector function can be written as a linear combination of the scalar basis functions ψ_i. Thus, we can split $\mathbf{R}_{ma}$ into R^i_{mxa} and R^i_{mxa}.

$$R^i_{mxa} = \int_{\Omega}\Big\{-\psi_i\Big[2u\frac{\partial \hat{u}}{\partial x} + v\Big(\frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x}\Big)\Big] + \frac{\partial\psi_i}{\partial x}\Big(-\frac{q}{\rho} + 2\nu\frac{\partial \hat{u}}{\partial x}\Big)$$
$$+ \frac{\partial\psi_i}{\partial y}\Big[\nu\Big(\frac{\partial \hat{u}}{\partial y} + \frac{\partial \hat{v}}{\partial x}\Big)\Big] - \psi_i\Big(\phi\frac{\partial\varphi}{\partial x}\Big)\Big\}\,\mathrm{d}\Omega - \int_{\Gamma_o}\psi_i\Big[\frac{\partial\mathcal{J}_{\Gamma}}{\partial u}\Big|_{\Gamma_s}$$
$$- 2n_x\hat{u}u - v(n_y\hat{u} + n_x\hat{v}) - \varphi\phi n_x\Big]\,\mathrm{d}\Gamma - \int_{\Omega_s}\psi_i\frac{\partial\mathcal{J}_{\Omega}}{\partial u}\,\mathrm{d}\Gamma; \tag{A-35}$$
$$i = 1, \cdots, N.$$

$$
\begin{aligned}
R^i_{mya} = \int_\Omega \Big\{ & -\psi_i\Big[u\Big(\frac{\partial \hat{u}}{\partial y}+\frac{\partial \hat{v}}{\partial x}\Big)+2v\frac{\partial \hat{v}}{\partial y}\Big]+\frac{\partial \psi_i}{\partial y}\Big(-\frac{q}{\rho}+2\nu\frac{\partial \hat{v}}{\partial y}\Big) \\
& +\frac{\partial \psi_i}{\partial x}\Big[\nu\Big(\frac{\partial \hat{u}}{\partial y}+\frac{\partial \hat{v}}{\partial x}\Big)\Big]-\psi_i\Big(\phi\frac{\partial \varphi}{\partial y}\Big)\Big\}\,\mathrm{d}\Omega-\int_\Gamma \psi_i\Big[\frac{\partial \mathcal{J}_\Gamma}{\partial v}\Big|_{\Gamma_s} \\
& -2n_y\hat{v}v-u(n_y\hat{u}+n_x\hat{v})-\varphi\phi n_y\Big]\,\mathrm{d}\Gamma-\int_{\Omega_s}\psi_i\frac{\partial \mathcal{J}_\Omega}{\partial v}\,\mathrm{d}\Gamma; \\
& i=1,\cdots,N.
\end{aligned}
\tag{A-36}
$$

The adjoint continuity equation results:

$$
R^i_{ca}=-\int_\Omega\Big(\frac{\partial \hat{u}}{\partial x}+\frac{\partial \hat{v}}{\partial y}\Big)\chi_i\,\mathrm{d}\Omega-\int_{\Omega_s}\frac{\partial \mathcal{J}_\Omega}{\partial p}\chi_i\,\mathrm{d}\Omega;\quad i=1,\cdots,M.
$$

and the adjoint scalar equation gives:

$$
\begin{aligned}
R^i_{sa} = & \int_\Omega\Big[-\psi_i\Big(u\frac{\partial \varphi}{\partial x}+v\frac{\partial \varphi}{\partial y}\Big)+\frac{\partial \psi_i}{\partial x}\Big(D\frac{\partial \varphi}{\partial x}\Big)+\frac{\partial \psi_i}{\partial y}\Big(D\frac{\partial \varphi}{\partial y}\Big)\Big]\,\mathrm{d}\Omega \\
& -\int_{\Gamma_i-\Gamma_o}\psi_i\Big(\frac{\partial \mathcal{J}_\Gamma}{\partial \phi}\Big|_{\Gamma_s}-\varphi(n_xu+n_yv)\Big)\,\mathrm{d}\Gamma-\int_{\Gamma_{\omega s}}\frac{\partial \mathcal{J}_\Gamma}{\partial \phi}\psi_i\,\mathrm{d}\Gamma \\
& -\int_{\Gamma_s}\frac{\partial \mathcal{J}_\Omega}{\partial \phi}\psi_i\,\mathrm{d}\Omega;\quad i=1,\cdots,N.
\end{aligned}
\tag{A-37}
$$

A.2.3
Expansion of unknown fields

The adjoint velocity, pressure and scalar fields must be written as linear expansions of the basis functions of the spaces of each field with ψ_i and χ_i as follows:

$$
\begin{aligned}
\boldsymbol{v} &= \begin{bmatrix}\hat{u}\\ \\ \hat{v}\end{bmatrix}=\begin{bmatrix}\sum_{j=1}^{N}\hat{U}_j\psi_j\\ \\ \sum_{j=1}^{N}\hat{V}_j\psi_j\end{bmatrix} \\
q &= \sum_{j=1}^{M}Q_j\chi_j \\
\varphi &= \sum_{j=1}^{N}\Psi_j\psi_j
\end{aligned}
\tag{A-38}
$$

A.2.4
Direct method

The direct method is used to solve the linear coupled system and is explained in Algorithm 2.

Algorithm 2: Direct Method

1 Set the adjoint residual $\mathbf{R}_a(\hat{\boldsymbol{c}}) = 0$; where $\hat{\boldsymbol{c}} = (\hat{U}, \hat{V}, Q, \Psi)^{\mathsf{T}}$
2 Set the algebraic equation:
3 $\underline{\mathbf{A}}\hat{\boldsymbol{c}} = \mathbf{b}$;
4 Solve for $\hat{\boldsymbol{c}}$:
5 $\hat{\boldsymbol{c}} = \underline{\mathbf{A}}^{-1}\mathbf{b}$

A.3
Interpolation using shape functions

Point sources appearing in the primal and adjoint equations are represented by Lagrangian markers superimposed on an Eulerian background mesh. Two approaches are used in this work, iterative inverse mapping and the Moving Least Square, for FEM and FVM, respectively.

A.3.1
Iterative inverse mapping

Silva et al. (2009), presented interpolation techniques using finite element shape functions, see Fig. A.2. The strategy consists of two parts: First, the element containing each data point is identified (the owner element). Then, the coordinates of the data points are transformed into the reference coordinate system of the finite element (referred to as inverse mapping). The reference coordinates are used to calculate the shape function coefficients of the element that perform the interpolation.

The finite element shape functions ψ_j are generally known as functions of the local coordinates of the element $\boldsymbol{\xi}$. Therefore, to evaluate these functions, it is generally necessary to compute $\boldsymbol{\xi}$ from $\boldsymbol{x}$. The equation (A-39) defines a nonlinear mapping function from the reference coordinates $\boldsymbol{\xi}$ to the global coordinates $\boldsymbol{x}$,

$$\boldsymbol{x} = \begin{bmatrix} x \\ \\ y \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^{n} X_j \psi_j \\ \\ \sum_{j=1}^{n} Y_j \psi_j \end{bmatrix} \tag{A-39}$$

The inverse mapping is numerically obtained by minimizing the error measure $\delta_{\boldsymbol{x}} = \|\boldsymbol{x}^* - \boldsymbol{x}\|$. Where we obtain $\boldsymbol{x}^*$ by mapping: $\boldsymbol{\xi}^* \rightarrow \boldsymbol{x}^*$. The iterative process begins with an initial estimate of the reference coordinates, $\boldsymbol{\xi}_0$. In this work, we use

the midpoint within the element, i.e., $(\xi_1, \xi_2) = (0, 0)$. If the inversion is successful, the error, δ_x, should approach machine accuracy.

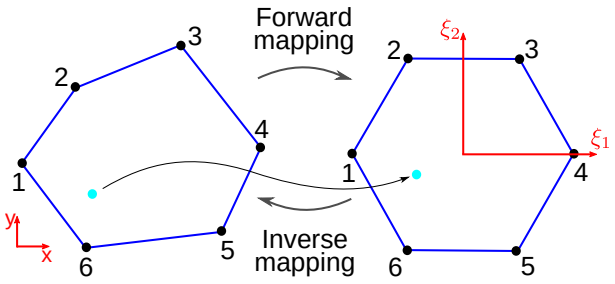

Figure A.2: Forward and inverse mapping scheme in polygonal element.

A.3.2
Moving Least Square

We describe the Moving Least Square (MLS) method on an unstructured mesh following the work of Tran et al. (2020). MLS is used in applications of the Immersed Boundary Method (Boffi; Gastaldi, 2003; Boffi et al., 2005; Li et al., 2015; Battista; Strickland; Miller, 2017; Le; Khoo, 2017; Cai et al., 2017). Figure A.3 outlines the Lagrangian markers (light blue dots) and the centers of the Eulerian mesh (black cross spots). The blue dashed circle with radius r_s is the support domain of the Lagrangian marker L.

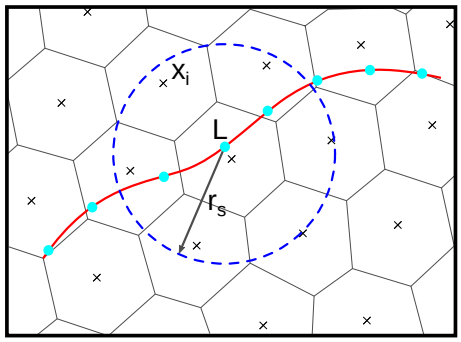

Figure A.3: Schematic diagram of a Eulerian unstructured mesh and Lagrangian grids.

To compute a state variable called ϕ in Lagrangian marker positions by the same state variable ϕ of Eulerian nodes in the support domain, we proceed as follows:

$$\phi(\mathbf{X}_L) = \mathbf{p}^T(\mathbf{X}_L)\mathbf{a}(\mathbf{X}_L) \tag{A-40}$$

where $\mathbf{p}(\mathbf{X}_L) = [1 X_L Y_L Z_L]^T$ is the weighted orthogonal basis function vector for the linear space and $\mathbf{a}(\mathbf{X}_L)$ is its coefficient vector determined by minimizing the weighted residual function.

$$R(\mathbf{X}_L, \mathbf{x}_i) = \sum_{i=1}^{n} W(\mathbf{X}_L - \mathbf{x}_i)[\mathbf{p}^T(\mathbf{x}_i)\mathbf{a}(\mathbf{X}_L) - \phi(\mathbf{x}_i)]^2 \tag{A-41}$$

where $W(\mathbf{X}_L - \mathbf{x}_i)$ is a weight function that depends on the distance between $\mathbf{X}_L$ and $\mathbf{x}_i$. We use a cubic spline for the weight function as follows:

$$W(\mathbf{X}_L - \mathbf{x}_i) = \begin{cases} \frac{2}{3} - 4r^2 + 4r^3, & r \leq 0.5 \\ \frac{4}{3} - 4r + 4r^2 - \frac{4}{3}r^3, & 0.5 \leq r \leq 1.0 \\ 0, & otherwise, \end{cases} \tag{A-42}$$

where $r = (\mathbf{X}_L - \mathbf{x}_i)/r_s$ and r_s is the radius of the support domain, which should be between two and five local grid sizes. After minimizing the residual function $R(\mathbf{X}_L, \mathbf{x}_i)$ with respect to $\mathbf{a}(\mathbf{X}_L)$ leads to:

$$\mathbf{a}(\mathbf{X}_L) = \mathbf{A}^{-1}(\mathbf{X}_L)\mathbf{B}(\mathbf{X}_L)\phi_s(\mathbf{x}_i) \tag{A-43}$$

$$\mathbf{A}(\mathbf{X}_L) = \sum_{i=1}^{n} W(\mathbf{X}_L - \mathbf{x}_i)\boldsymbol{p}(\mathbf{x}_i)\boldsymbol{p}(\mathbf{x}_i)^T \tag{A-44}$$

$$\mathbf{B}(\mathbf{X}_L) = [W(\mathbf{X}_L - \mathbf{x}_1)\mathbf{p}(\mathbf{x}_1)W(\mathbf{X}_L - \mathbf{x}_2)\mathbf{p}(\mathbf{x}_2)...W(\mathbf{X}_L - \mathbf{x}_n)\mathbf{p}(\mathbf{x}_n)] \tag{A-45}$$

and $\phi_s(\mathbf{x}_i) = [\phi(\mathbf{x}_1)\phi(\mathbf{x}_2)...\phi(\mathbf{x}_n)]^T$ is the state variable of n Eulerian nodes in the support domain.

Inserting Eq. A-43 into Eq. A-40, we have:

$$\phi(\mathbf{X}_L) = \mathbf{\Phi}_{iL}^T(\mathbf{X}_L, \boldsymbol{x}_i)\phi(\boldsymbol{x}_i) \tag{A-46}$$

where

$$\mathbf{\Phi}_{iL}^T(\mathbf{X}_L, \boldsymbol{x}_i) = \mathbf{p}^T(\mathbf{X}_L)\mathbf{A}(\mathbf{X}_L)^{-1}\mathbf{B}(\mathbf{X}_L) \tag{A-47}$$

Point sources can be obtained by a distribution scheme using the shape function $\mathbf{\Phi}_{iL}^T$ bridging the node i in the Eulerian mesh and the marker L in the Lagrangian mesh, as follows:

$$f_i = \frac{1}{V_i}\sum_{L=1}^{m} \Phi_i^L(\mathbf{X}_L, \mathbf{x}_i)\mathbf{F}_L \tag{A-48}$$

where $f_i, \mathbf{x}_i$ and V_i are the sources, the spatial coordinate and the volume of the Eulerian node i, respectively. $\mathbf{F}_L$ and $\mathbf{X}_L$ are the size of the source and the spatial coordinates of the Lagrangian marker L and m is the number of Lagrangian markers surrounding the Eulerian node i.

B
The Finite Volume Method Discretization

Here we follow the work of Darwish; Moukalled (2016) (and the nomenclature used there) to illustrate the process of finite volume discretization. The governing equations $(2\text{-}3)$ are integrated over the elements (or finite volumes) into which the domain is partitioned. Then, Gauss' theorem is applied to convert the volume integrals of the convection and diffusion terms into surface integrals. Then, the surface and volume integrals are converted to discrete integrals and integrated numerically by using integration points (ip)

The conservation equation for a general scalar variable ϕ in stationary form can be expressed as follows:

$$\underbrace{\nabla \cdot (\rho \boldsymbol{u} \phi)}_{convective\,term} = \underbrace{\nabla \cdot (\Gamma^{\phi} \nabla \phi)}_{diffusion\,term} + \underbrace{Q^{\phi}}_{source\,term} \tag{B-1}$$

By integrating the above equation over the element C shown in Fig. B.1; Eq.B-1 is transformed to

$$\int_{V_C} \nabla \cdot (\rho \boldsymbol{u} \phi)\, \mathrm{d}V = \int_{V_C} \nabla \cdot (\Gamma^{\phi} \nabla \phi)\, \mathrm{d}V + \int_{V_C} Q^{\phi}\, \mathrm{d}V \tag{B-2}$$

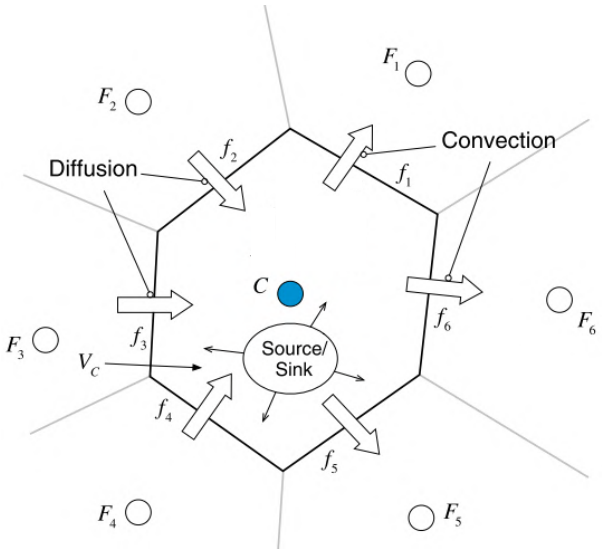

Figure B.1: Conservation in a discrete element (Darwish; Moukalled, 2021).

Replace the volume integrals of the convection and diffusion terms with surface integrals using the divergence theorem, the above equation becomes:

$$\oint_{\partial V_C} (\rho \boldsymbol{u} \phi) \cdot \mathrm{d}\mathbf{S} = \oint_{\partial V_C} (\Gamma^{\phi} \nabla \phi) \cdot \mathrm{d}\mathbf{S} + \int_{V_C} Q^{\phi}\, \mathrm{d}V \tag{B-3}$$

where Q^{ϕ} is the source term, $\mathbf{S}$ is the surface vector, $\boldsymbol{u}$ is the velocity vector, ϕ is the obtained quantity, and $\oint_{\partial V_C}$ is the surface integral over the volume V_C.

B.1
Flux Integration Over Element Faces

Defining the total flux $\mathbf{J}^{\phi}$ as the sum of convection and diffusion fluxes, it can be written as follows:

$$\mathbf{J}^{\phi} = \rho \boldsymbol{u}\phi - \Gamma^{\phi}\nabla\phi \tag{B-4}$$

Replacing the surface integral over the cell C by a summation of the flux terms over the faces of the element C, the surface integral of the total flux becomes:

$$\oint_{\partial V_C} \mathbf{J}^{\phi} \cdot \mathrm{d}\mathbf{S} = \sum_{f \sim faces(V_C)} \left(\int_f \mathbf{J}^{\phi} \cdot \mathrm{d}\mathbf{S} \right) \tag{B-5}$$

In Eq.B-5, the surface fluxes are evaluated at the faces of the element instead of being integrated into the element. This transformation has important implications for the properties of the FVM. One of them is that it makes the method conservative.

The surface integral at each face of the element must be evaluated. Using a Gaussian quadrature, the integral at the face f of the element becomes:

$$\int_f \mathbf{J}^{\phi} \cdot \mathrm{d}\mathbf{S} = \int_f (\mathbf{J}^{\phi} \cdot \boldsymbol{n}) \mathrm{d}S = \sum_{ip \sim ip(f)} (\mathbf{J}^{\phi} \cdot \boldsymbol{n})_{ip} \omega_{ip} S_f \tag{B-6}$$

where ip refers to an integration point and $ip(f)$ is the number of integration points along the surface f. A number of options are available, the accuracy of which depends on the number of integration points used and the weighing function ω_{ip}.

B.2
Source Term Volume Integration

Volume integration is used for the source term. Using a Gaussian quadrature integration, the volume integral of the source term is calculated as follows:

$$\int_{V_C} Q^{\phi} \, \mathrm{d}V = \sum_{ip \sim ip(V_C)} \left(Q^{\phi}_{ip} \omega_{ip} V_C \right) \tag{B-7}$$

B.3
Flux Linearization

The area flux can be divided into a linear part, a function of the ϕ values at the nodes spanning the face (i.e., ϕ_C and ϕ_F), and a nonlinear part, which includes the remaining part that cannot be expressed in terms of ϕ_C and ϕ_F. The resulting equation can be written as follows:

$$\mathbf{J}^{\phi}_f \cdot \mathbf{S}_f = FluxT_f = FluxC_f \phi_C + FluxF_f \phi_F + FluxV_f \tag{B-8}$$

where $FluxT_f$ represents the total flux through face f and is decomposed into three terms. The first two terms represent the contributions of the two elements sharing the face and are written over the linearization coefficients $FluxC_f$ and $FluxF_f$. The last term describes the nonlinear contribution that cannot be expressed in terms of ϕ_C and ϕ_F and is given by the nonlinear term $FluxV_f$. The values of $FluxC_f$, $FluxF_f$, and $FluxV_f$ depend, of course, on the discretized term and the scheme used for its discretization. Repeating for all cell faces yields:

$$\begin{aligned}\sum_{f\sim nb(C)}\left(\mathbf{J}_f^{\phi}\cdot\mathbf{S}_f\right) &= \sum_{f\sim nb(C)}(FluxT_f)\\ &= \sum_{f\sim nb(C)}(FluxC_f\phi_C + FluxF_f\phi_F + FluxV_f)\end{aligned} \tag{B-9}$$

The linearization of the volume flow is done by expressing it as a linear function of the element node value ϕ_C and given by:

$$Q_C^{\phi}V_C = FluxC\phi_C + FluxV \tag{B-10}$$

Using the flux Linearization Eq.B-9 and B-10) leads to the algebraic relationship we are looking for:

$$a_C\phi_C + \sum_{F\sim NB(C)}(a_F\phi_F) = b_C \tag{B-11}$$

where the relations between equation coefficients and flux linearization coefficients are expressed as:

$$a_C = \sum_{f\sim nb(C)} FluxC_f - FluxC \tag{B-12}$$

$$a_F = FluxF_f \tag{B-13}$$

$$b_C = -\sum_{f\sim nb(C)} FluxV_f + FluxV \tag{B-14}$$

B.4
Boundary Conditions

The use of different boundary conditions leads to different solutions, even if the general equation remains the same. Boundary conditions are applied to boundary elements that have one or more faces on the boundary. Denote by C the centroid of the boundary element shown in Fig. B.2 or Fig. B.3 with a boundary face of centroid b and outward facing surface vector $\mathbf{S}_b$. As before, the discretization process over cell C yields the result:

$$\sum_{f\sim nb(C)} \left(\mathbf{J}_b^{\phi} \cdot \mathbf{S}\right)_f = Q_C^{\phi} V_C \tag{B-15}$$

The fluxes on the interior faces are discretized as before, while the boundary flux is discretized with the aim of constructing a linearization with respect to ϕ_C, thus:

$$\mathbf{J}_b^{\phi} \cdot \mathbf{S}_b = FluxC_b\phi_C + FluxV_b \tag{B-16}$$

The specification of boundary conditions involves either the specification of the unknown boundary value ϕ_b or, alternatively, the boundary flux $\mathbf{J}_b^{\phi}$. There are a variety of types of boundary conditions. However, two of the most commonly used for general scalars are the Dirichlet and Neumann boundary conditions.

B.4.1
Dirichlet Boundary Condition (Value Specified)

A Dirichlet boundary condition is a type of boundary condition that specifies the value of ϕ at the boundary, i.e.,

$$\phi_b = \phi_{b,specified} \tag{B-17}$$

For the boundary face shown in Fig.B.2, the boundary flux is evaluated using the known value of ϕ_b according to Eq.B-16. Thus, the value of the boundary flux is not unknown, but can be calculated directly. Thus, in Eq.B-11 the coefficients are:

$$a_C = FluxC_b + \sum_{f\sim nb(C)} FluxC_f - FluxC \tag{B-18}$$

$$a_F = FluxF_f \tag{B-19}$$

$$b_C = -(FluxV_b + \sum_{f\sim nb(C)} FluxV_f) + FluxV \tag{B-20}$$

The following important observations can be made about the discretized boundary equation:

- The coefficient a_b is larger than other neighbor coefficients because b is closer to C and consequently has a larger influence on ϕ_C.
- The coefficient a_C is still the sum of all neighboring coefficients including a_b. This means that for the boundary element $\sum_{F\sim NB(C)} |a_F|/|a_C| < 1$ is the second necessary condition for satisfying the Scarborough criterion, and thus in each iteration the convergence of the linear system of equations is guaranteed by an iterative solution method.
- The $a_b\phi_b$-product $(= FluxV_b)$ is now on the right side of the equation, i.e. it is part of b_C, since it contains no unknowns.

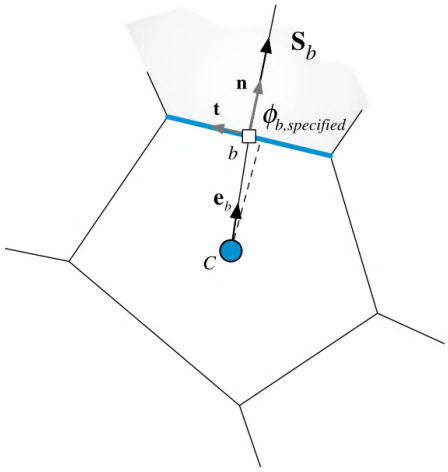

Figure B.2: Dirichlet boundary condition (Darwish; Moukalled, 2021).

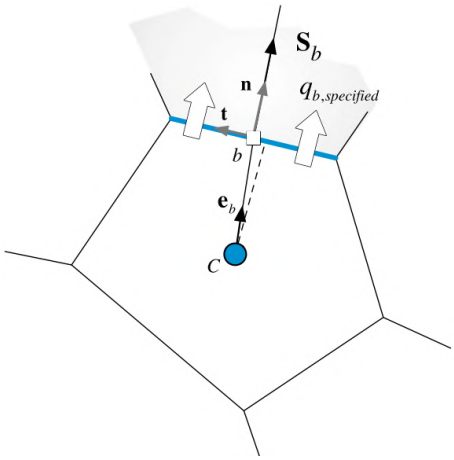

Figure B.3: Neumann boundary condition (Darwish; Moukalled, 2021).

B.4.2
Flux Specified (Neumann Boundary Condition)

If the flux (or normal gradient to the face) of ϕ is given at the boundary (Fig.B.3), then the boundary condition is called the Neumann boundary condition. In this case, the specified flux is given by:

$$\mathbf{J}_b^{\phi} \cdot \mathbf{S}_b = \underbrace{\mathbf{J}_b^{\phi} \cdot \boldsymbol{n}_b}_{specified\,flux} \quad S_b = q_{b,specified} S_b = FluxC_b \phi_C + FluxV_b \tag{B-21}$$

In the above equation, $q_{b,specified}$ is a known user-specified quantity representing flux per unit area, and now $FluxC_b = 0$. So from Eq.B-11 we get the following coefficients:

$$a_C = \sum_{f \sim nb(C)} FluxC_f - FluxC \tag{B-22}$$

$$a_F = FluxF_f \tag{B-23}$$

$$b_C = -(FluxV_b + \sum_{f \sim nb(C)} FluxV_f) + FluxV \tag{B-24}$$

To the above discretized equation the following important points can be made:

- A Von Neumann boundary condition does not lead to a dominant a_C-coefficient.
- If both q_b and S_b^{ϕ} are zero, then ϕ_C is bounded by its neighbors. Otherwise, ϕ_C may exceed (or fall below) the neighbor values of ϕ, which is allowed. For example, if ϕ is the temperature, then q_b represents the heat flux applied to the boundary. Thus, if heat is added at the boundary, then the temperature in the region near the boundary is expected to be higher than in the interior.
- Once ϕ_C is calculated, the boundary value ϕ_b can be calculated:

$$\phi_b = \frac{FluxC_b \phi_C - q_b}{FluxC_b} \tag{B-25}$$

- Finally, the Von Neumann condition can be considered as a natural boundary condition for the finite volume method, since for the case where the specified flux is zero, nothing needs to be done in terms of discretization for the face, while a specified value of zero (Dirichlet condition) still requires the discretization to be performed.

B.5 Interpolation schemes

The convection and diffusion terms of the equation sets were discretized with the second-order upwind scheme and the central difference scheme, respectively.

C
The SIMPLE Algorithm

The Navier-Stokes equation (N-S) is a nonlinear equation. It also has more variables (u, v, w, p) than equations. The system cannot be solved directly because there is no explicit equation for the pressure. Therefore, we need an algorithm to do this. The Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) technique (Patankar, 1980; Versteeg; Malalasekera, 2007) is an algorithm where pressure and velocity are coupled. The solution procedure is based on reformulating the Navier-Stokes equations in terms of a momentum predictor and a pressure correction equation that enforces the satisfaction of the continuity condition, see (Abbas, 2020) for more details.

The first step is to express the momentum equations in the general matrix form:

$$\mathcal{M}U = -\nabla p \tag{C-1}$$

where $\mathcal{M}$ is the coefficient matrix of velocities. Next, this sparse matrix $\mathcal{M}$ is transformed into a diagonal $\mathcal{A}$ and an off-diagonal $\mathcal{H}$ matrix. The decomposition is performed as follows:

$$\mathcal{A}U - \mathcal{H} = -\nabla p \tag{C-2}$$

By rearranging Eq.C-2

$$U = \frac{\mathcal{H}}{\mathcal{A}} - \frac{\nabla p}{\mathcal{A}} \tag{C-3}$$

and substitute it into the continuity equation $\nabla \cdot U = 0$. This gives the Poisson equation for pressure:

$$\nabla \cdot \left(\frac{\nabla p}{\mathcal{A}}\right) = \nabla \cdot \left(\frac{\mathcal{H}}{\mathcal{A}}\right) \tag{C-4}$$

The discretized equations are solved sequentially in an iterative procedure that can be summarized in the algorithm **??**.

The SIMPLE algorithm is easily extendable to adjoint equations, since they have the same structure as their primal counterparts. For a deeper insight into the SIMPLE algorithm, we recommend (Versteeg; Malalasekera, 2007) and (Darwish; Moukalled, 2021).

Algorithm 3: SIMPLE Method

```
Choose an initial attempt of pressure, velocity and scalar fields;
while velocity not satisfy momentum equation do
    Solve the momentum equation Eq.(C-1) for the velocity field;
    Solve the Poisson equation for the pressure Eq.(C-4);
    Use the pressure field to correct the velocity field so that it satisfy the
     continuity equation, Eq.(C-3);
    Solve the scalar equation using the current value of velocity;
end
```

D
Optimization Algorithms

The complexity of an optimization problem depends on several factors, such as the number of design variables involved, the degree of convexity/nonconvexity and linearity/nonlinearity of the equations, and the associated numerical difficulties (Subramaniam; Dbouk; Harion, 2019). Therefore, the choice of an efficient numerical technique and an equally reliable optimization algorithm is very important to ensure stable convergence to an optimal solution. In this regard, the optimization problem could be solved using various approaches such as the quasi-Newton BFGS, Optimality Criteria (OC), the Method of Moving Asymptotes (MMA), and others.

D.1
The Quasi-Newton BFGS Method

The gradient-based Quasi-Newton BFGS method is a limited-memory Quasi-Newton algorithm, named for its discoverers Broyden, Fletcher, Goldfarb, and Shanno (Byrd et al., 1995), that produces superlinear convergence and is now described.

In a gradient-based method, we perform a line search $\mathbf{y}_{k+1} = \mathbf{y}_k + a_k \mathbf{p}_k$ for a directional step size a_k. We search for the search direction $\mathbf{p}$, at each iteration k:

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla \mathcal{J}_k \tag{D-1}$$

where $\mathbf{B}_k$ depends on the algorithm used. The secant method provides us with a method to iteratively construct the approximate Hessian such that:

$$\mathbf{B}_{k+1}(\mathbf{y}_{k+1} - \mathbf{y}_k) = \nabla \mathcal{J}_{k+1} - \nabla \mathcal{J}_k \tag{D-2}$$

There exist many different ways to approximate the Hessian, which for the BFGS method is the following:

$$\mathbf{B}_{k+1} = \mathbf{B}_k - \frac{\mathbf{B}_k \mathbf{s}_k \mathbf{s}_k^\mathsf{T} \mathbf{B}_k}{\mathbf{s}_k^\mathsf{T} \mathbf{B}_k \mathbf{s}_k} + \frac{\mathbf{r}_k \mathbf{r}_k^\mathsf{T}}{\mathbf{s}_k^\mathsf{T} \mathbf{r}_k} \tag{D-3}$$

where $\mathbf{s}_k = \mathbf{y}_{k+1} - \mathbf{y}_k$ and $\mathbf{r}_k = \nabla \mathcal{J}_{k+1} - \nabla \mathcal{J}_k$. Equation (D-3) can be substituted into Eq.(D-1).

The algorithm for the BFGS quasi-Newton method can be seen in Algorithm (4). In this work, we use the L-BFGS-B algorithm for simple boundaries implemented in Matlab by Granzow (2017). It is based on the gradient projection method and uses a BFGS matrix with limited memory to approximate the Hessian of the objective function. This makes it well suited for optimization problems with a large number of design variables bounded by lower and upper bounds. The L-BFGS-B approach,

unlike the OC and MMA approaches, does not consider restriction functions. It is therefore particularly suitable for parametric optimization problems For more background on gradient-based optimization methods, the reader is referred to the text of Nocedal; Wright (2006) and Herzog; Kunisch (2010).

Algorithm 4: Quasi-Newton BFGS

```
Choose an initial feasible design y^(k); set k ← 0;
Define the inverse Hessian approximation;
while (convergence criteria are not met) do
    Compute the search direction according to Eq.(D-1);
    Carry out a line search where y_{k+1} = y_k + a_k p_k for a direction step
     size a_k;
    Determine the new approximate Hessian using Eq.(D-3);
    Set k ← k + 1;
end
```

D.2 The Optimality Criteria Method

The method of optimality criteria (OC) still finds its place in topology optimization due to its numerical simplicity and efficiency. Its optimality condition can be expressed as follows:

$$B_e = \frac{-\frac{\partial \mathcal{J}}{\partial \gamma_e}}{\lambda \frac{\partial v}{\partial \gamma_e}} \tag{D-4}$$

where λ is a Lagrange multiplier associated with the constraint v, which can be found by a bi-sectioning algorithm satisfying that $v(\gamma) = 0$ and $B_e = 1$.
Following the work of Bendsøe; Sigmund (2004), a heuristic updating scheme for the design variables can be formulated as follows:

$$\gamma_e^{new} = \begin{cases} \max(\gamma_{min}, \gamma_e - m) & \text{if } \gamma_e B_e^{\eta} \leq \max(\gamma_{min}, \gamma_e - m), \\ \gamma_e B_e^{\eta} & \text{if } \max(\gamma_{min}, \gamma_e - m) < \gamma_e B_e^{\eta} < \min(1, \gamma_e + m), \\ \min(1, \gamma_e + m) & \text{if } \min(1, \gamma_e + m) \leq \gamma_e B_e^{\eta}, \end{cases} \tag{D-5}$$

where m is a positive motion limit and $\eta (= 1/2)$ is a numerical damping coefficient. The algorithm OC is explained in Algorithm (**??**). Admittedly, the optimality

criteria-based optimization implemented here is only good for a single constraint and it is based on a fixed-point type heuristic update scheme.

Algorithm 5: Optimality Criteria

```
Choose an initial design y^(k); set k ← 0;
while (convergence criteria are not met) do
    Solve the primal equations to obtain the corresponding (u, p, φ) fields;
    Compute objective function;
    Perform sensitivity analysis by using the primal and adjoint equations;
    Apply filter techniques;
    Update design variables using Eq.(D-5) to obtain y^(k+1);
    Set y^(k+1) ← y^(k);
    Set k ← k + 1;
end
```

D.3
The Method of Moving Asymptotes

The method of moving asymptotes (MMA) remains one of the most popular algorithms in the structural optimization community. Svanberg (1987), introduced this method for nonlinear programming in general and structural optimization. Later, Svanberg (1998) presented some modeling aspects and solution schemes, Svanberg (2002) presented the globally convergent version of MMA, called GCMMA, and Aage; Lazarov (2013) implemented a parallelized algorithm using MMA on a fully parallel topology optimization framework in C++.

This gradient-based algorithm is based on generating and solving convex approximating subproblems. Given $\mathbf{y} = (\gamma_1, ..., \gamma_n)^{\mathsf{T}}$, the vector of design variables. The method generates subproblems where in each current iteration k both the objective function and the restrictions are obtained by linearization into variables of type $\frac{1}{\gamma_j - L_j}$ or $\frac{1}{U_j - \gamma_j}$ depending on the signs of their respective derivatives at $\mathbf{y}^{(k)}$. The mentioned lower and upper asymptotes $L_j^{(k)}$ and $U_j^{(k)}$ are chosen so that $L_j^{(k)} < \gamma_j^{(k)} < U_j^{(k)}$, and are iteratively updated to reduce oscillations and improve the convergence rate.

The heuristic rule proposed by Svanberg (1987) for updating L_j and U_j is as follows: For $k = 1$ and $k = 2$,

$$\begin{aligned} L_j^{(k)} &= \gamma_j^{(k)} - 0.5(\gamma_j^{max} - \gamma_j^{min}) \\ U_j^{(k)} &= \gamma_j^{(k)} + 0.5(\gamma_j^{max} - \gamma_j^{min}) \end{aligned} \tag{D-6}$$

In later iterations, when $k \geq 3$,

$$\begin{aligned} L_j^{(k)} &= \gamma_j^{(k)} - \vartheta_j^{(k)}(\gamma_j^{(k-1)} - \gamma_j^{(k-1)}) \\ U_j^{(k)} &= \gamma_j^{(k)} + \vartheta_j^{(k)}(\gamma_j^{(k-1)} - \gamma_j^{(k-1)}) \end{aligned} \tag{D-7}$$

where,

$$\vartheta_j^{(k)} = \begin{cases} 0.7, & \text{if } (\gamma_j^{(k)} - \gamma_j^{(k-1)})(x_j^{(k-1)} - \gamma_j^{(k-2)}) < 0 \\ 1.2, & \text{if } (\gamma_j^{(k)} - \gamma_j^{(k-1)})(x_j^{(k-1)} - \gamma_j^{(k-2)}) > 0 \\ 1, & \text{if } (\gamma_j^{(k)} - \gamma_j^{(k-1)})(x_j^{(k-1)} - \gamma_j^{(k-2)}) = 0 \end{cases} \tag{D-8}$$

The MMA approach is explained in Algorithm (6).

Algorithm 6: Method of Moving Asymptotes

```
Choose an initial feasible design y^(k); set k ← 0;
while (convergence criteria are not met) do
    if k=1 or k=2 then
        Update L_j^k and U_j^k using (D-6);
    else
        Update L_j^k and U_j^k using (D-7) and (D-8);
    end
    Compute both the objective function and restrictions, and their
     derivatives;
    Solve the MMA subproblem to obtain y^(k+1);
    Set y^(k-2) ← y^(k-1), y^(k-1) ← y^(k), y^(k) ← y^(k+1);
    Set k ← k + 1;
end
```

The general convergence/stopping criterion applied is based on the changes in the design variables in the last iteration being negligible and the square norm of the Karush-Kuhn-Tucker (KKT) conditions being less than a very small positive value.