TOPOLOGY OPTIMIZATION IN OPENFOAM USING A CONTINUOUS ADJOINT FRAMEWORK

Author:

Nadeem Kever

Advisors:

Antonio Huerta and Matteo Giacomini

Master's degree in Numerical Methods in Engineering
Escola Tècnica Superior d'Enginyeria de Camins, Canals i Ports de Barcelona

January 2023

UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

For the people that hold me up.

Acknowledgements

First and foremost I have to thank my mother and father. Without their constant support and love I would not be where I am today. My brothers, Aseem, Khamisa, you have been my constant companions through life. My dear family, you are a necessity in my life like the beating of my heart. You give me purpose. Without you, I would cease to be what I am. Thank you.

My dear partner Adriana, no one knows more than you the challenges I have faced in these recent years. Through all of the turmoil and tumult you have stuck by me. Not only that – you have nourished me with your kindness, your friendship, your presence. I am thankful for all of that. I especially cherish all of our great conversations. I am not sure where I would be if you hadn't joined me on this journey. Thank you.

My colleagues, my friends, my companions – thank you in no particular order Davide, Stefano, Rafel, Luan, Alvaro, Mariano, Fabiola, Christina, and Pau. Each of you has helped me in this journey whether it was through your wisdom when I needed advice or simply through your friendship. I feel lucky to have been surrounded by such a kind group of people. I appreciate all of you; thank you.

To my friends outside of UPC, thank you for making Barcelona feel like home. You have been my family. I smile thinking of all the moments we've shared together. Thank you.

Finally, I would like to thank my advisors, Dr. Antonio Huerta and Dr. Matteo Giacomini, for providing an interesting topic of study to which I could dedicate myself to. Thank you for providing the guidance, supervision, and direction that helped bring this project to its completion.

“I know not all that may be coming, but be it what it will, I’ll go to it laughing.”

HERMAN MELVILLE

Abstract

In this work we report on the addition of features to an existing OpenFOAM topology optimization solver. We begin with a detailed summary of the theoretical basis of topology optimization in the case of the incompressible steady-state Navier-Stokes equations under an adjoint framework. We then describe the theoretical basis for the implementation of our extended solver. Namely, we have implemented two new features: a power dissipation objective function and a volume constraint. The validity of the volume constraint is successfully tested via a toy problem. Following validation, outputs of the newly modified solver are benchmarked against literature for cases that include a duct system with one inlet and two outlets, an alternate configuration of the same duct system, and a dual bottleneck two-outlet problem. We also include a review of the current literature on topology optimization methods and the implementation of volume constraints in density-based methods.

Keywords: Density based topology optimization, adjoint method, constrained optimization, volume constraint, incompressible steady-state Navier-Stokes equations, OpenFOAM

Contents

List of Figures

Chapter 1

Introduction

Optimal control theory was first utilized to solve optimization problems in 1971 by Lions [35]. Utilized in this approach was an objective function tracking the deviation from optimality [7]. The first application of adjoint methods in the area of aerodynamics was done by Pironneau, in 1984, in the case of potential flows governed by elliptic differential equations [40]. This was the first development of shape optimization and it was introduced as a shape control technique to minimize an objective function [13]. Since their development, shape optimization techniques have been typically applied in order to optimize and refine geometries in such a way as to restrict the development of new topological entities i.e. the spontaneous creation of new boundaries. In addition, when dealing with problems in fluid dynamics, particularly in the case of internal or ducted flow, a geometry optimization case in which only the inlet and outlet are predefined, the selection of appropriate control points and parameters is non-trivial and in fact may influence or restrict the final design. Topology optimization is the solution to directly address and resolve these challenges [7].

Structural mechanics saw the first introduction of this idea, topology optimization, in the work by Bendsoe and Kikuchi [20]. Compared to the standard approach of modifying the boundary to search for an optimal shape of a structure, the method in that paper instead varied the material density distribution to find regions where, in order to increase structural stiffness, material should be added [7]. Topology optimization has been used to tackle many problems in structural mechanics, as seen in [22]. In addition, the method has been applied in coupled problems such as structural optimization dominated by heat conduction [23, 30, 41, 48].

The idea of topology optimization in fluid dynamics was first developed in the cornerstone work by Borvall and Petersson in the case of Stokes flows [1]. The approach to fluid dynamic topology optimization begins with a volume mesh which is chosen to encompass the entire design space. This is the domain used for the computation of fluid dynamic quantities. Then, with the introduction of a porosity term, the method restricts or allows flow through parts of the domain. In a given volume mesh, this term is defined cell wise. The porosity is defined such that at high values the cell becomes close to impermeable by the fluid and thus functions as a momentum penalty in a specific cell. At values of zero, however, the porosity does not penalize flow in a given cell and so the fluid is allowed to flow unimpeded. This porosity, typically labeled α_i for cell i, is modeled via Darcy's Law and its distribution defines the geometry. Particularly, each α_i is a design variable in the optimization problem which determines the values of α_i based on the minimization of a given objective function. Noting the large number of design variables that coincide with the total number of cells in a given mesh, the adjoint method becomes an apt choice, due to its relative independence on the number of design variables in terms of computational cost, to compute the sensitivities of the given objective function with respect to the design variables i.e the porosity in each cell.

Extensive development was done expanding the work of Borvall and Peterson on Stokes flow [16, 29, 32, 33, 43]. Olesen et al. [39] and Gersborg-Hansen et al. [31] extended the idea of topology optimization to laminar flows. In a majority of the literature one finds that topology optimization is applied to internal flows; however, some recent publications [21] have extended the idea to external flows in order to minimize lift over drag of airfoils in incompressible laminar flow regimes. Though the work of Borvall and Peterson used a density-based topology optimization, using a porosity term as a design variable, alternative approaches have been developed relatively recently. One method that has garnered particular interest in the community is the level set approach, in which a level set is defined to strictly differentiate between regions of material and non-material [17, 24, 26, 27, 28, 34, 44, 47]. The level set method's advantage is in its clear definition of boundary between materials. Density-based methods for Navier-Stokes problems face the issue of "grey" areas. These grey areas represent a transition zone between material and non-material and neither belongs to the flow nor the solidified domain, and thus causes ambiguity in the definition of the design boundary.

In addition to strictly flow problems, topology optimization has been utilized in several multiphysics applications: in thermal fluid interactions in the laminar regime [25, 36, 45], fluid-structure interactions [18, 19], aeroelastic problems [37, 42] and acoustic structure problems [46].

The topology optimization method was further extended to turbulent flows using the "frozen turbulence" assumption via the continuous adjoint method in the paper by Othmer [4]. The frozen turbulence assumption being that the variation of the turbulent viscosity is of order zero and is thus neglected. Much of the work in this manuscript is developed on the foundation of this paper by Othmer and the development of the OpenFOAM solver by Othmer et al. [2].

Topology optimization codes have traditionally existed in expensive commercial and proprietary software; however, the development of open source frameworks has largely improved accessibility – particularly with OpenFOAM in the field of computational fluid dynamics. In fact, a solver for fluid based topology optimization has already been developed in OpenFOAM by Othmer et al. [2]. However, it lacks two major components if we seek to compare it to benchmark cases in literature, that is: the power dissipation objective function and a volume constraint. The publicly available solver, which comes default in OpenFOAM, does not in fact implement the power dissipation objective function; instead the current solver is built to minimize pressure losses [3]. As many benchmark cases in the literature use power dissipation as an objective function with the addition of a volume constraint, there is sufficient motivation to add these additional features to the existing code. This will be the central task to which the work reported in this paper is dedicated.

Thus, the necessity of this work lies in the fact that the current solver is written to minimize pressure losses and lacks a volume constraint. In order for the OpenFOAM solver to be beneficial to future research and development, it must be adequately comparable to benchmark cases in literature. Hence, the goal of this work is to code additional functionality into the current solver. The specific additions will be: primarily, power dissipation as an objective function and secondarily, a volume constraint. The purpose of these additions are so that the OpenFOAM solver can be benchmarked with current literature and as a result be validated for future research.

In the literature there are several approaches to implementing a volume constraint in topology optimization. One approach is the augmented Lagrangian method [38, 7] in which the standard definition of the Lagrangian in topology optimization is modified with the addition of a volume constraint term. This method is discussed in the Appendix. Another approach is the method of moving asymptotes [11]. These two methods can be loosely categorized as "implicit" methods, in the sense that they modify the optimization model to implicitly include the volume constraint. An alternative approach is defining an explicit volume constraint, as was done by Popovac [10]. This approach goes cell-by-cell in a mesh analyzing the sensitivity in each cell, and if that sensitivity is above a certain threshold, blocks the cell entirely by placing the maximum amount of material there. Then, the method checks if the amount of material placed in the design domain satisfies the volume constraint. If it does not, the sensitivity threshold is increased and the algorithm repeats itself. Although some modifications are made, the method defined by Popovac is the foundation for the algorithm written in this paper. However, in the paper by Popovac, only the pressure loss objective function is utilized. In this work, with the implementation of the power dissipation objective function, we extend the utility of the solver written by Popovac.

This work is organized as follows: in chapter 2, the theoretical problem tackled in this work is described in detail. Then in chapter 3 the theory for adjoint optimization in the case of an incompressible, steady-state Navier-Stokes flow is discussed. This is reduced to the case of 2D internal flows, then the framework is used to derive the adjoint equations and adjoint boundary conditions for the power dissipation objective function. To conclude the theory section, the addition of a volume constraint is discussed. Then in chapter 4, the success of the volume constraint implementation is visualized. Next, the power dissipation objective function is utilized to optimize a case that is compared with the literature. In addition, the objective function is tracked to visualize its evolution and convergence in the optimization process. This was done using a bash script tied to a python executable that the author wrote in order to visualize this iterative evolution of the objective function. The details of how this was done are in the appendix, in which a short exposition is provided to explain the scripts. In the last section, some concluding remarks are made. In addition, an appendix is provided to summarize additional information including: details of the derivation of the adjoint method for the flow problem studied here, an alternate approach to the volume constraint, a tutorial for the suite of code developed in the duration of this work, and finally the addition of the energy equation in order to model temperature.

Chapter 2

Topology Optimization Problem Statement

The discussion of this text is focused on the incompressible, steady-state Navier-Stokes problem with the addition of the Darcy term. This is summarized for a domain of interest Ω in the governing equations below:

$$-\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{2.1a}$$

$$(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v} + \boldsymbol{\nabla}p - \boldsymbol{\nabla}\cdot\left\{2\nu\boldsymbol{\nabla}^S\boldsymbol{v}\right\} + \alpha\boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{2.1b}$$

where $\boldsymbol{v}$ and p are the velocity and pressure of the fluid respectively; $\boldsymbol{\nabla}^S\boldsymbol{v} = \frac{1}{2}\left(\boldsymbol{\nabla}\boldsymbol{v} + (\boldsymbol{\nabla}\boldsymbol{v})^T\right)$ is the strain rate tensor and ν is the effective kinematic viscosity (with laminar and turbulent contributions). The additional $\alpha\boldsymbol{v}$ is known as the Darcy term and penalizes the momentum in counterproductive cells. The magnitude of α determines the degree of penalization and it can vary from 0 (no penalty / no material) to a constant defined as the maximum value (no flow/ solid material). It is the core aspect of density based topology optimization. With these terms defined, equations 2.1a and 2.1b are canonically known as the mass and momentum conservation equations respectively.

The problem statement is completed with the inclusion of the Dirichlet and Neumann boundary conditions on $\partial\Omega = \Gamma_{\mathrm{D}} \cup \Gamma_{\mathrm{N}}$, that is

$$\begin{cases} \boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}} & \text{on } \Gamma_{\mathrm{D}}, \\ (-p\boldsymbol{I} + 2\nu\boldsymbol{\nabla}^S\boldsymbol{v})\boldsymbol{n} = \boldsymbol{q}_N & \text{on } \Gamma_{\mathrm{N}}. \end{cases} \tag{2.2}$$

Here Γ_{D} and Γ_{N} are respectively the Dirichlet and Neumann portions of the domain boundary; $\boldsymbol{u}_{\mathrm{D}}$ and $\boldsymbol{q}_N$ are the imposed velocity profile and stress on Γ_{D} and Γ_{N} respectively; $\boldsymbol{I}$ and $\boldsymbol{n}$ are the identity matrix and the outward normal unit vector to the boundary.

The topology optimization problem aims to minimize a cost function J subject to constraints $\boldsymbol{\mathcal{R}}$ – that is:

$$\begin{aligned} &\text{minimize } J = J(\alpha, \boldsymbol{v}, p) \\ &\text{subject to } \boldsymbol{\mathcal{R}}(\alpha, \boldsymbol{v}, p) = 0. \end{aligned} \tag{2.3}$$

In this case, the design variable α is the Darcy porosity term, and the constraints $\boldsymbol{\mathcal{R}}$ are given by the state equations defined in Eq. 2.1. Thus the incompressible, steady-state Navier-Stokes equations now read as

$$\boldsymbol{\mathcal{R}}_{\boldsymbol{v}} = (\boldsymbol{v} \cdot \boldsymbol{\nabla})\, \boldsymbol{v} + \boldsymbol{\nabla} p - \boldsymbol{\nabla} \boldsymbol{\cdot} \left\{ 2\nu \boldsymbol{\nabla}^S \boldsymbol{v} \right\} + \alpha \boldsymbol{v} = 0, \tag{2.4a}$$

$$\mathcal{R}_p = -\boldsymbol{\nabla} \boldsymbol{\cdot} \boldsymbol{v} = 0, \tag{2.4b}$$

where $\boldsymbol{\mathcal{R}}_{\boldsymbol{v}}$ and $\mathcal{R}_p$ are used as compact notation to refer to the momentum and mass conservation equations respectively such that the constraints abstractly defined in Eq. 2.4 can be written as

$$\boldsymbol{\mathcal{R}} = \{\boldsymbol{\mathcal{R}}_{\boldsymbol{v}}, \mathcal{R}_p\}. \tag{2.5}$$

Chapter 3

Adjoint-Based Topology Optimization

The optimization problem defined in Eq. 2.3 in view of Eq. 2.5 can be reformulated in terms of a Lagrange problem with functional L to be minimized as an effective cost function. That is

$$L \coloneqq J + \int_{\Omega} \boldsymbol{u} \cdot \boldsymbol{\mathcal{R}}_{\boldsymbol{v}} \, d\Omega + \int_{\Omega} q \mathcal{R}_p \, d\Omega, \tag{3.1}$$

where the Lagrange multipliers corresponding to the adjoint velocity $\boldsymbol{u}$ and adjoint pressure q are introduced.

We then need to calculate the total variation of the Lagrangian L in order to obtain the sensitivity of the objective function with respect to the design variable α. Note, that the Lagrangian functional is not exclusively dependent on α; it also has contributions from $\boldsymbol{v}$ and p – we must extract the dependence on these variables by applying the chain rule, doing so yields:

$$\delta L = \frac{\partial L}{\partial \alpha} \delta \alpha + \frac{\partial L}{\partial \boldsymbol{v}} \delta \boldsymbol{v} + \frac{\partial L}{\partial p} \delta p \tag{3.2}$$

At face value, calculating the sensitivities directly from the above system seems like a daunting problem to solve. A solution would imply that for each design variable α we need to solve the state equations. However, the simplicity of the adjoint method comes in defining the optimality equations so as to isolate the variables of interest and as such make the calculation of the sensitivity practically independent of the state equation [4][12].

To derive the adjoint problem and the sensitivity of the objective function J with respect to the design variable α, first order optimality conditions are written. The first optimality condition is the sensitivity equation:

$$\frac{\partial L}{\partial \alpha} = 0. \tag{3.3}$$

The next two will yield the adjoint Navier-Stokes equations:

$$\frac{\partial L}{\partial \boldsymbol{v}} = 0 \tag{3.4}$$

$$\frac{\partial L}{\partial p} = 0. \tag{3.5}$$

The final two simply recover the original constraints $\boldsymbol{\mathcal{R}_v}$ and $\mathcal{R}_p$

$$\frac{\partial L}{\partial \boldsymbol{u}} = 0 \tag{3.6}$$

$$\frac{\partial L}{\partial q} = 0. \tag{3.7}$$

Alternatively, in the approach taken by the standard adjoint method, starting from Eq. 3.1 we could directly choose the Lagrange multipliers $\boldsymbol{u}$ and q such that they vanish identically in the domain Ω with respect to the variation of the state variables $\boldsymbol{v}$ and p [4][5]:

$$\frac{\partial L}{\partial \boldsymbol{v}}\delta\boldsymbol{v} + \frac{\partial L}{\partial p}\delta p = 0 \tag{3.8}$$

Regardless of using Eqs 3.3 - 3.5 or Eq 3.8 above, we arrive at the same conclusion for calculating the sensitivity

$$\delta L = \frac{\partial L}{\partial \alpha}\,\delta\alpha \tag{3.9}$$

$$\Leftrightarrow \delta L = \frac{\partial J}{\partial \alpha}\,\delta\alpha + \int_\Omega \boldsymbol{u}\cdot\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial \alpha}\,\delta\alpha d\Omega + \int_\Omega q\frac{\partial \mathcal{R}_p}{\partial \alpha}\,\delta\alpha\, d\Omega. \tag{3.10}$$

This illustrates the central tenet of the adjoint method in that the above equation only relies on relatively inexpensive derivatives taken with respect to the porosity term. Adjoint optimization is most beneficial when optimization is performed with respect to a large number of design variables, as is the case here. Efficiency is maintained under such conditions because the time cost of sensitivity analysis is virtually independent of the number of design variables [12]. This is what leads to the computational affordability generally attributed to the adjoint method.

3.1 Calculating the Sensitivity

Then the sensitivity can be computed from Eq. 3.3 by considering the problem in the discretized framework; thus, the sensitivity of the Lagrangian L is calculated with respect to the porosity α_i at each cell i as follows:

$$\frac{\partial L}{\partial \alpha_i} = \frac{\partial J}{\partial \alpha_i} + \int_\Omega \boldsymbol{u}\cdot\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial \alpha_i}d\Omega + \int_\Omega q\frac{\partial \mathcal{R}_p}{\partial \alpha_i}d\Omega. \tag{3.11}$$

In the framework of topology optimization, there is typically no dependence of the objective function J on the porosity. The porosity describes the continuous transition from fluid to solid and is considered an auxiliary variable. As such: $\partial J/\partial \alpha_i = 0$. In addition, the only dependence on α in our constraints $\boldsymbol{\mathcal{R}}$ is in the Darcy term in $\boldsymbol{\mathcal{R}}_{\boldsymbol{v}}$; accordingly, from Eq. 2.4 we find

$$\frac{\partial \boldsymbol{\mathcal{R}}_{\boldsymbol{v}}}{\partial \alpha_i} = \boldsymbol{v}_i, \quad \frac{\partial \mathcal{R}_{\boldsymbol{p}}}{\partial \alpha_i} = 0, \tag{3.12}$$

which, for the momentum residual, yields the velocity in the barycenter of cell i. This simplifies our equations to

$$\frac{\partial L}{\partial \alpha_i} = \int_\Omega \boldsymbol{u}_i \cdot \boldsymbol{v}_i \, d\Omega. \tag{3.13}$$

Thus the sensitivity for each cell, with the use of one integration point per cell, is the scalar product of the adjoint and primal velocity in the cell times the volume of the given cell V_i, that is

$$\frac{\partial L}{\partial \alpha_i} = (\boldsymbol{u}_i \cdot \boldsymbol{v}_i)\, V_i. \tag{3.14}$$

Therefore, to calculate the sensitivity, we require the flow velocity $\boldsymbol{v_i}$ which is given by solving the state equations $\boldsymbol{\mathcal{R}}$, the volume for a given cell V_i which is easily calculated given a mesh, and the adjoint velocity $\boldsymbol{u}_i$. This variable $\boldsymbol{u}_i$, the final piece for the sensitivity equation, must be calculated using the adjoint equations and adjoint boundary conditions. These are discussed in the next section.

3.2 Adjoint Equations and Boundary Conditions

In order to perform this derivation it is helpful to define what the derivative of an objective function $J(x)$ looks like with respect an arbitrary variable x. Taking this derivative looks as follows:

$$\frac{\partial J(x)}{\partial x} = \frac{\partial}{\partial x}\left\{ \int_\Gamma J_\Gamma(x)\, d\Gamma + \int_\Omega J_\Omega(x)\, d\Omega \right\} \tag{3.15}$$

$$= \int_\Gamma \frac{\partial}{\partial x} J_\Gamma(x)\, d\Gamma + \int_\Omega \frac{\partial}{\partial x} J_\Omega(x)\, d\Omega. \tag{3.16}$$

Here we have deconstructed the objective function into surface contributions J_Γ evaluated on the boundary of the domain $\Gamma = \partial\Omega$ and volume contributions J_Ω evaluated in the domain Ω.

Then the adjoint equations and boundary conditions equations are derived from the conditions defined in equations 3.4 and 3.5. Recall that these conditions come from defining that our adjoint variables, $\boldsymbol{u}$ and q such that the variation of the Lagrangian

L is identically zero when varied with respect to our primary variables, $\boldsymbol{v}$ and p. As a side note, we could just as well use Eq. 3.8 as a starting point to derive the adjoint and adjoint boundary equations as is done in the available literature [4]. However, in this work we will arrive at the same conclusion using equations 3.4 and 3.5. The details of the derivation are tedious and require several long calculations, as such they have been appended at the end of this manuscript for the reader's interest. Therefore, using the results of Appendix A, the adjoint Navier-Stokes equations in the case of topology optimization are

$$-(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{v} + \boldsymbol{\nabla} q - \boldsymbol{\nabla}\cdot(2\nu\boldsymbol{\nabla}^s\boldsymbol{u}) + \alpha\boldsymbol{u} = -\frac{\partial J_\Omega}{\partial \boldsymbol{v}} \quad \text{in} \quad \Omega, \tag{3.17a}$$

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = \frac{\partial J_\Omega}{\partial p} \quad \text{in} \quad \Omega. \tag{3.17b}$$

In addition, the adjoint boundary conditions are as follow

$$\int_\Gamma \left[\boldsymbol{n}(\boldsymbol{u}\cdot\boldsymbol{v}) + \boldsymbol{u}(\boldsymbol{v}\cdot\boldsymbol{n}) + 2\nu\boldsymbol{n}\cdot\boldsymbol{\nabla}^s\boldsymbol{u} - q\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial \boldsymbol{v}}\right]\cdot\delta\boldsymbol{v}\, d\Gamma - \int_\Gamma \boldsymbol{n}\cdot[2\nu\boldsymbol{\nabla}^s(\delta\boldsymbol{v})]\cdot\boldsymbol{u}\, d\Gamma = 0 \tag{3.18a}$$

$$\boldsymbol{u}\cdot\boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \quad \text{on} \quad \Gamma. \tag{3.18b}$$

The above adjoint and boundary equations constitute the general adjoint system of equations for the steady-state, incompressible Navier–Stokes equations with the inclusion of the Darcy porosity and using the assumption of frozen turbulence. Note the similarities between our adjoint equations in 3.17 and our primary equations in 2.1. In fact, this similarity goes further as in many objective functions of interest we find $J_\Omega = 0$– as such, Eq. 3.17b reduces and we find that the adjoint flow field is also divergence free. The major dissimilarity between the two sets of equations is the sign of the convective term in Eq. 3.17a; in the adjoint equation the convective term has a negative prefactor meaning the adjoint velocity is convected in the opposite direction of the primal velocity.

3.3 The Special Case of Internal Flows

Now we wish to reduce our adjoint system in the case of internal flows. In these cases, the domain boundary Γ can be broken into three distinct regions: the inlet(s), outlet(s) and the wall. In most cases of interest, the standard boundary condition for the inlet is a prescribed flow and a zero pressure gradient, the outlet a zero velocity gradient and zero pressure, and finally for the walls we have a no-slip condition i.e. a prescribed velocity of zero and a zero pressure gradient.

It is interesting to note that as many objective functions of interest have no domain dependence ($J_\Omega = 0$) [4]; we find that our adjoint equations are actually independent from the choice of objective function. This is a convenient result as changing objective functions requires no change of the solver, only reprogramming for the boundary equations. Taking this into account, below the adjoint equations are summarized in their final form:

$$-(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{v} + \boldsymbol{\nabla} q - \boldsymbol{\nabla}\boldsymbol{\cdot}(2\nu\boldsymbol{\nabla}^s\boldsymbol{u}) + \alpha\boldsymbol{u} = 0, \tag{3.19}$$

$$\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{u} = 0. \tag{3.20}$$

Now we wish to turn our attention to the adjoint boundary equations defined in equations 3.18. Before we do so, it is useful for us to make a simplification noting that our fields $\boldsymbol{u}$ and $\delta\boldsymbol{v}$ are divergence free. Then, the third and last terms of Eq. 3.18a can be reduced in the following way:

$$\begin{aligned} &\int_\Gamma 2\nu\boldsymbol{n}\cdot\{-\boldsymbol{\nabla}^s(\delta\boldsymbol{v})\cdot\boldsymbol{u} + \boldsymbol{\nabla}^s\boldsymbol{u}\cdot\delta\boldsymbol{v}\}\ d\Gamma \\ &= \int_\Gamma \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})\boldsymbol{u}\cdot\delta\boldsymbol{v} - \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta\boldsymbol{v}\cdot\boldsymbol{u}\ d\Gamma + \int_\Omega -\boldsymbol{\nabla}\boldsymbol{\cdot}(2\nu\boldsymbol{\nabla}^s\boldsymbol{u})\cdot\delta\boldsymbol{v}\ d\Omega. \end{aligned} \tag{3.21}$$

Then substituting this result appropriately in Eq. 3.18a, our simplified boundary conditions are as follows:

$$\begin{aligned} \int_\Gamma \left[\boldsymbol{n}(\boldsymbol{u}\cdot\boldsymbol{v}) + \boldsymbol{u}(\boldsymbol{v}\cdot\boldsymbol{n}) + \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})\boldsymbol{u} - q\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial\boldsymbol{v}}\right]\cdot\delta\boldsymbol{v}\,d\Gamma & \\ - \int_\Gamma \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta\boldsymbol{v}\cdot\boldsymbol{u}\,d\Gamma = 0 & \end{aligned} \tag{3.22a}$$

$$\boldsymbol{u}\cdot\boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \tag{3.22b}$$

We use these equations to derive the adjoint boundary conditions in the three distinct regions of Γ previously mentioned for internal flows.

3.3.1 Prescribed Velocity and Zero Pressure Gradient Conditions (typically inlet and wall)

In this case, as the velocity is a fixed value, the variation of the velocity vanishes i.e. $\delta\boldsymbol{v} = 0$. Thus the first term of Eq. 3.22a is zero. Then we would like to simplify the second term in Eq. 3.22a. We begin by decomposing the variation into its normal ($\perp$) and tangential $\|$ components.

$$(\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta\boldsymbol{v} = (\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta\boldsymbol{v}_\perp + (\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta\boldsymbol{v}_\|. \tag{3.23}$$

Using the fact that the admissible perturbations of the primary velocity are divergence free we can simplify the above with the following relation:

$$\boldsymbol{\nabla}\boldsymbol{\cdot}\delta\boldsymbol{v} = (\boldsymbol{n}\cdot\boldsymbol{\nabla})\delta v_\perp + \boldsymbol{\nabla}_\|\cdot\delta\boldsymbol{v}_\| = 0. \tag{3.24}$$

Along the inlet where we prescribe a velocity and along the walls where we have a no-slip condition, of course, the tangential components of the velocity are zero: in these directions $\delta \boldsymbol{v}_\parallel = 0$. Then the above equation yields:

$$(\boldsymbol{n} \cdot \boldsymbol{\nabla})\delta v_\perp = 0. \tag{3.25}$$

Returning to Eq. 3.23 we now have

$$(\boldsymbol{n} \cdot \boldsymbol{\nabla})\delta \boldsymbol{v} = (\boldsymbol{n} \cdot \boldsymbol{\nabla})\delta \boldsymbol{v}_\parallel. \tag{3.26}$$

Then Eq. 3.22a is now restated as:

$$\int_\Gamma \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})\delta \boldsymbol{v}_\perp \cdot \boldsymbol{u}_\parallel \, d\Gamma = 0 \tag{3.27}$$

Then our final boundary conditions are summarized below:

$$\boldsymbol{u}_\parallel = 0 \tag{3.28}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \tag{3.29}$$

It is interesting to note that this solely places boundary conditions on the adjoint velocity and not the adjoint pressure. However, observing the nature of how the adjoint pressure q enters the adjoint state equations and recognizing it is in the same way as the primary pressure p enters the primary state equations, we conclude that we should apply the same zero pressure gradient boundary condition for q.

3.3.2 Zero velocity Gradient and Zero Pressure Conditions (typically outlet)

In the specified case, we have $p = 0$ and $(\boldsymbol{n} \cdot \boldsymbol{\nabla})\delta \boldsymbol{v} = 0$. As such, Eq. 3.22a reduces to only the first integral. Thus we can represent the boundary condition in strong form:

$$\boldsymbol{n}(\boldsymbol{u} \cdot \boldsymbol{v}) + \boldsymbol{u}(\boldsymbol{v} \cdot \boldsymbol{n}) + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})\boldsymbol{u} - q\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial \boldsymbol{v}} = 0 \tag{3.30}$$

Breaking this down into tangential and normal components we find:

$$\text{Tangential:} \quad \boldsymbol{u}_\parallel v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})\boldsymbol{u}_\parallel + \frac{\partial J_\Gamma}{\partial \boldsymbol{v}}_\parallel = 0 \tag{3.31}$$

$$\text{Normal:} \quad \boldsymbol{u} \cdot \boldsymbol{v} + u_\perp v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})u_\perp - q + \frac{\partial J_\Gamma}{\partial v_\perp} = 0 \tag{3.32}$$

The tangential equation gives us the tangential component of the adjoint velocity:

$$\left[v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})\right] \boldsymbol{u}_\parallel = -\frac{\partial J_\Gamma}{\partial \boldsymbol{v}}_\parallel. \tag{3.33}$$

From this, we can derive the gradient of the normal component of the adjoint velocity (using incompressibility):

$$(\boldsymbol{n} \cdot \boldsymbol{\nabla})u_\perp = \boldsymbol{\nabla} \boldsymbol{\cdot} \boldsymbol{u} - \boldsymbol{\nabla}_\parallel u_t \tag{3.34}$$

$$= -\boldsymbol{\nabla}_\parallel u_t \tag{3.35}$$

to then be used in the normal equation to give us the adjoint pressure:

$$q = \boldsymbol{u} \cdot \boldsymbol{v} + u_\perp v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})u_\perp + \frac{\partial J_\Gamma}{\partial v_\perp} \tag{3.36}$$

With this we complete the set of adjoint boundary conditions in the case of a zero pressure condition and zero velocity gradient.

Now that we have completely described our set of adjoint and adjoint boundary conditions in the case of internal flows, we can apply these equations to our objective function of choice: power dissipation.

We compactly summarize the relevant equations for topology optimization in the case of internal flows using an adjoint framework[4]:

Adjoint Navier-Stokes equations

$$-(\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\boldsymbol{u} \cdot \boldsymbol{v} + \boldsymbol{\nabla} q - \boldsymbol{\nabla} \boldsymbol{\cdot} (2\nu \boldsymbol{\nabla}^s \boldsymbol{u}) + \alpha \boldsymbol{u} = 0, \tag{3.37a}$$

$$\boldsymbol{\nabla} \boldsymbol{\cdot} \boldsymbol{u} = 0. \tag{3.37b}$$

Adjoint boundary conditions for the inlet and wall

$$n \cdot \boldsymbol{\nabla} q = 0 \tag{3.38a}$$

$$\boldsymbol{u}_\parallel = 0 \tag{3.38b}$$

$$\boldsymbol{u} \cdot \boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \tag{3.38c}$$

Adjoint boundary conditions for the outlet

$$[v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})]\, \boldsymbol{u}_\parallel = -\frac{\partial J_\Gamma}{\partial \boldsymbol{v}_{\ \parallel}} \tag{3.39a}$$

$$q = \boldsymbol{u} \cdot \boldsymbol{v} + u_\perp v_\perp + \nu(\boldsymbol{n} \cdot \boldsymbol{\nabla})u_\perp + \frac{\partial J_\Gamma}{\partial v_\perp} \tag{3.39b}$$

3.4 Topology Optimization of Power Dissipation Under a Volume Constraint

In this section, the topology optimization framework is specified in the case of dissipated power as an objective function with the addition of a volume constraint. The power dissipation objective can be written as follows:

$$J := -\int_{\Gamma} \left\{ p + \frac{1}{2}\boldsymbol{v}\cdot\boldsymbol{v} \right\} \boldsymbol{v}\cdot\boldsymbol{n}\, d\Gamma \tag{3.40}$$

It describes the net inward flux (total pressure) through the domain boundaries [4]. According to our definition in Eq. 3.15 we have $J_\Omega = 0$ and

$$J_\Gamma = -\left\{ p + \frac{1}{2}\boldsymbol{v}\cdot\boldsymbol{v} \right\} \boldsymbol{v}\cdot\boldsymbol{n} \tag{3.41}$$

In order to use Eq. 3.40 as an objective function, a new set of boundary conditions need to be introduced in accordance with Eqs. 3.38 and 3.39. Taking the derivative of the objective function with respect to the primary variables yields:

$$\frac{\partial J_\Gamma}{\partial \boldsymbol{v}} = -\left\{ p + \frac{1}{2}\boldsymbol{v}\cdot\boldsymbol{v} \right\} \boldsymbol{n} - \left\{ \boldsymbol{v}\cdot\boldsymbol{n} \right\} \boldsymbol{v} \tag{3.42}$$

$$\frac{\partial J_\Gamma}{\partial p} = -\boldsymbol{v}\cdot\boldsymbol{n} \tag{3.43}$$

Therefore, in the case of prescribed velocity and zero pressure gradient conditions we have:

$$\boldsymbol{u}_{\parallel} = 0 \tag{3.44}$$

$$\boldsymbol{u}\cdot\boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} = \boldsymbol{v}\cdot\boldsymbol{n} = v_{\perp} \tag{3.45}$$

These conditions are applied on the inlet of the geometry. Note that if this is applied on the wall of our domain (rather than the inlet) we have $v_{\perp} = 0$ due to the no slip condition.

And under the conditions of zero velocity gradient and pressure prescribed to zero we have:

$$\left[v_{\perp} + \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla}) \right] \boldsymbol{u}_{\parallel} = -\frac{\partial J_\Gamma}{\partial \boldsymbol{v}}_{\parallel} = v_{\perp}\boldsymbol{v}_{\parallel} \tag{3.46}$$

$$\begin{aligned} q &= \boldsymbol{u}\cdot\boldsymbol{v} + u_{\perp}v_{\perp} + \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})u_{\perp} + \frac{\partial J_\Gamma}{\partial v_{\perp}} \\ &= \boldsymbol{u}\cdot\boldsymbol{v} + u_{\perp}v_{\perp} + \nu(\boldsymbol{n}\cdot\boldsymbol{\nabla})u_{\perp} + \left\{ -\frac{1}{2}\boldsymbol{v}\cdot\boldsymbol{v} - v_{\perp}^2 \right\} \end{aligned} \tag{3.47}$$

These are the boundary conditions applied to the outlet of the problem.

This completes the description for the boundary conditions in the case of the power dissipation objective function.

3.4.1 Addition of a Volume Constraint

Finally, a volume constraint is introduced in the topology optimization problem. There are several ways adding a volume constraint to the OpenFOAM solver can be approached. One approach is the method of moving asymptotes as discussed in [11]; another is the augmented Lagrangian approach described in Appendix A. Both of these approaches are implicit in the sense that they require redefinition of the optimization algorithm to implicitly account for the volume constraint. Whereas the method implemented in this paper is considered an explicit volume constraint in the sense that it explicitly filters sensitivities such that the volume constraint is satisfied. The method discussed and implemented here is chosen because it avoids the complications and computational complexity of implicit volume constraints and as such guarantees a numerically robust approach [10].

However, this method is not without its own drawbacks. Particularly in the case when the sensitivities span many orders of magnitude, the main issue with this method is the definition of step size within the range of sensitivities. The paper by Popovac, which the algorithm implemented in this work is mainly based on, opts to handle this problem by logarithmically tracking the evolution of the sensitivity. However, as the standard OpenFOAM solver (and the modification of the solver done in this work) produce negative sensitivities – applying the logarithm directly to the sensitivity is not possible. Thus we have to conclude: Popovac's modification of the solver requires that the negative sensitivities are in some way handled so that a logarithm can be applied to them. To avoid mathematical complexity, the approach taken in this paper is to instead take the raw values of the sensitivities and divide those – meaning the step size between sensitivity values is taken linearly, rather than logarithmically . However, as Popovac states in his paper – "depending on the flow characteristics, [the sensitivity] can vary by several orders of magnitude," we must handle the wide range of sensitives carefully. If we simply take the range from the maximum sensitivity to the lowest (the difference between these values could be upwards of five orders of magnitude in certain cases) and divide equally then we will run into 2 possible issues. 1. Our division is far too coarse and thus we skip entirely over the domain of sensitivities that will allow us satisfy the volume constraint. 2. Alternatively our step size is small enough to provide the necessary resolution of the sensitivity we require to accurately satisfy our volume constraint – however, in the case where our sensitivity spans many orders of magnitude, the fineness of the step size will hinder our algorithm far too computationally expensive.

If a user runs into this issue and finds that the range of sensitivities span several orders of magnitude, a method is suggested here to handle it. A simple approach to addressing this issue is to manually define the range of sensitivities for the algorithm to search through. This fine tuning requires an idea in the first place of the sensitivity range relevant to satisfying the volume constraint and so cannot be done a priori i.e. tuning must be done based on a previous simulation.

Below the simplest form of the algorithm is presented.

Algorithm 1: Volume Constraint Algorithm

```
sensRange = range(maxSens, minSens, numSteps)

for sens_i in sensRange do
    /* sens_i = ith sensitivity in sensRange */

    foreach cell in mesh do
        if cellSens > sens_i then
            set $\alpha_{\text{cell}} = \alpha_{\max}$
        end
    end

    vol = $\int_{\Omega} \alpha d\Omega$

    if vol ≥ targetVolume then
        break
    end

end
```

Finally, it was found that one cannot simply start the volume constraint from iteration zero of the fluid simulation as it does not result in a converged or accurate final solution. This makes sense as it requires some number of iterations for the standard solver to place appropriate sensitivities in the design domain. This problem is addressed by creating a parameter that selects which iteration of the solver the volume constraint algorithm begins being applied. This is a user defined parameter so that it may be changed depending on the flow problem. The flow chart for this new solver is shown below following the diagram outlined in [10]

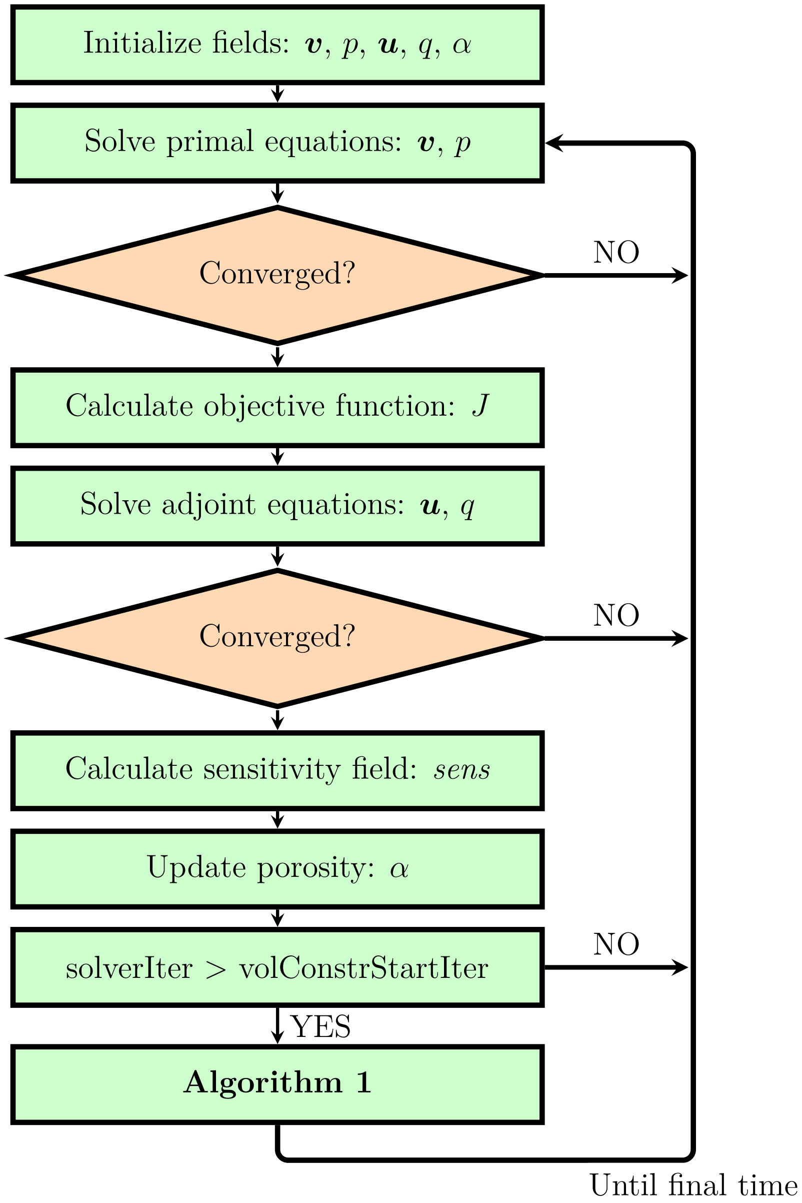

Initialize fields: $\boldsymbol{v}$, p, $\boldsymbol{u}$, q, α
Solve primal equations: $\boldsymbol{v}$, p
Converged?
NO
Calculate objective function: J
Solve adjoint equations: $\boldsymbol{u}$, q
Converged?
NO
Calculate sensitivity field: *sens*
Update porosity: α
solverIter > volConstrStartIter
NO
YES
Algorithm 1
Until final time

Figure 3.1: Flow chart of the implemented OpenFOAM solver for topology optimization. The flowchart here is based on my development of the solver. Based on the flowchart discussed in [10].

Chapter 4

Numerical Examples

4.1 Validation of Volume Constraint

To track the solver through successive stages of development it was useful to create a toy problem on which the solver was tested. As such, it is also instructive to include this toy problem as a first example to see in an obvious manner the success of the written solver. This problem is as follows: the design domain is a 1×1 box with an inlet of width 0.2 placed in the left top part of the box at a distance of 0.1 away from the top edge; an outlet is placed on the bottom left part of the domain at a distance of 0.1 from the right edge. The problem geometry is summarized in Fig 4.1

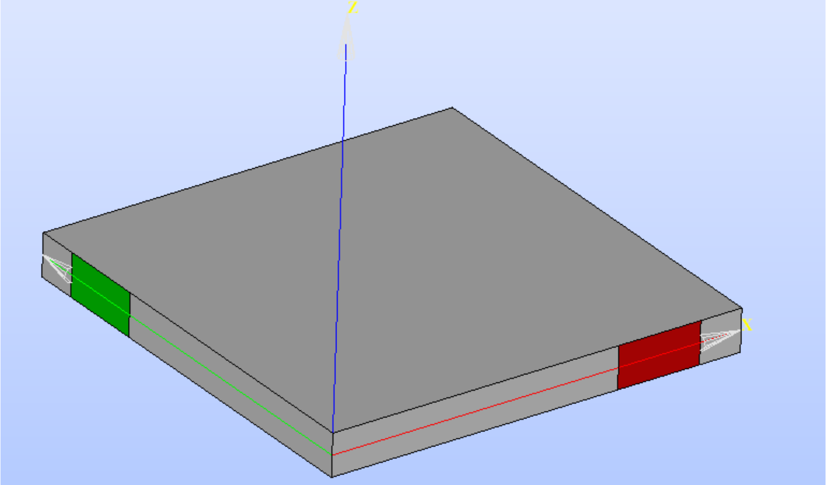
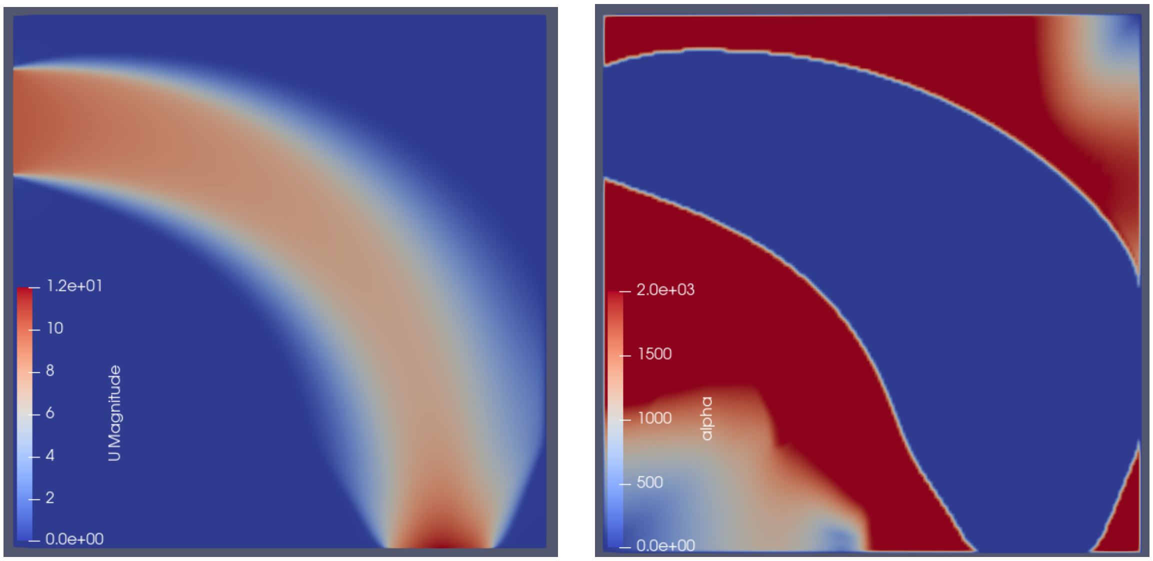


Figure 4.1: Geometry of the toy problem used to test the implementation of the volume constraint. The inlet is seen in green and the outlet in red.

The Reynolds number is 2000 of the results shown for this example. However, the solver converged for Reynolds numbers tested between 200 and $200,000$. The case is simulated for a total of 1200 iterations as that yields a converged solution in the standard solver. The objective function for each case here is power dissipation. The main purpose of this problem is to showcase the success of the volume constraint.

In Fig 4.2 we see the final solution of the standard solver. As a reminder, recall that the available solver lacks a volume constraint and is programmed to minimize the pressure losses as an objective function. Note the 'wispy' application of the porosity term alpha in this case outside the main portion of the flow. In addition, the fluid has a non-negligible velocity throughout much of the domain. These results have some obvious problems and we seek to refine these results with the use of a volume constraint.



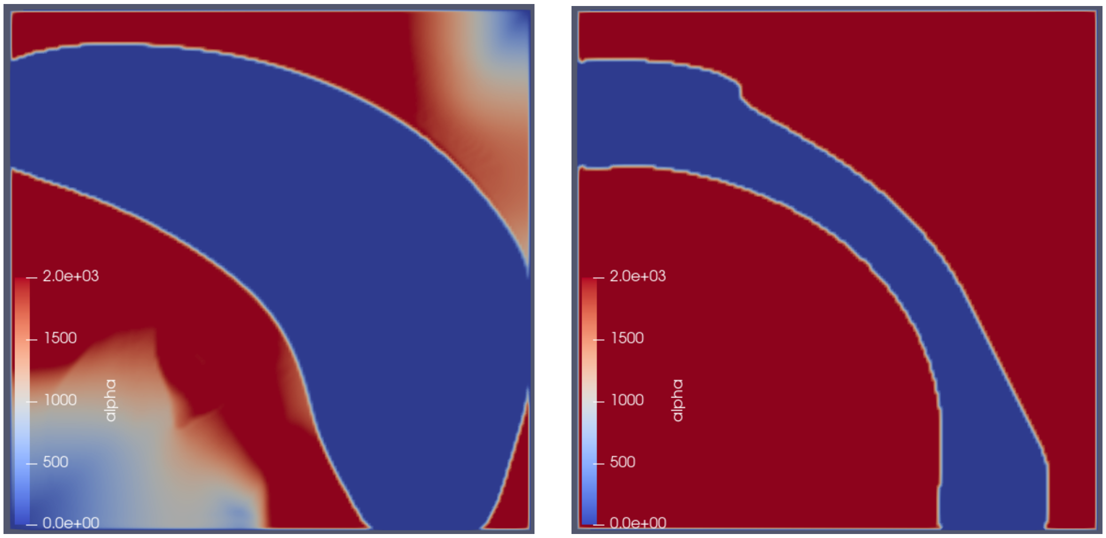
(a) Velocity profile

(b) Material distribution

Figure 4.2: Converged result of the modeled problem using the standard OpenFOAM solver.

To alleviate the issues seen in the previous case we rerun the problem with the application of a volume constraint. Let us set the volume constraint to 80% i.e. we want at least 80% of the volume of the design domain to be occupied by material. In this case we begin to apply the volume constraint starting at iteration 500.

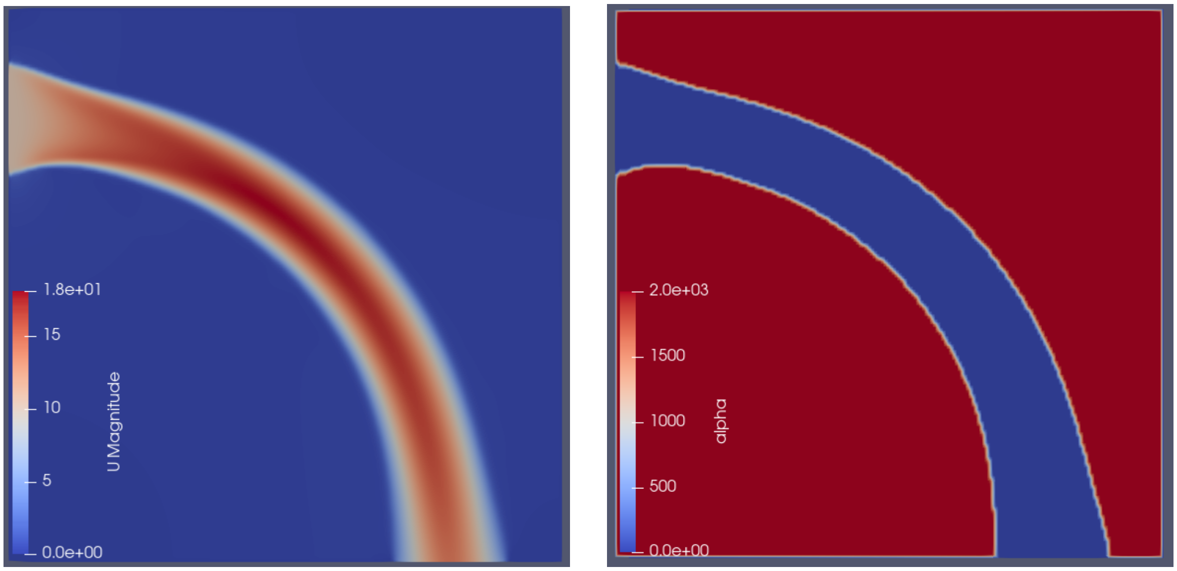
In Fig 4.3 above, we observe the application of the volume constraint using the explicit approach starting iteration 500. It is obvious that the result produced at iteration 500 is nowhere close to being optimized. This result is in fact due to the explicit nature of the volume constraint – essentially, only in this first step, it somewhat haphazardly places material in the design space to ensure via brute force satisfaction of the volume constraint. In Fig 4.4 we see the converged result. Indeed the algorithm has smoothed the surface of the pipe that it begun creating in iteration 500 to converge to the results shown.



(a) Material distribution on iteration 475

(b) Material distribution on iteration 500

Figure 4.3: On the left is the result of the problem at iteration 475, before the volume constraint is applied. On the right is the design domain immediately after the volume constraint is applied explicitly.



(a) Velocity profile

(b) Material distribution

Figure 4.4: Converged results of a volume constrained case. Here it is specified that at least 80% of the design space is material.

Next we visualize the evolution of the objective function in 4.5. Observe at iteration 500 there is a significant jump in the value of the objective function corresponding to the first iteration in which the volume constraint is applied. After this jump the method progressively begins optimizing the result once more eventually converging. One could ask at this point: what is the purpose of applying a volume constraint if it yields a higher objective function value? The answer is simple: indeed, the inclusion of a volume constraint will often yield less optimized geometries; however, these geometries are physically manifestable. Essentially, the volume constraint will create well-defined machinable geometries.

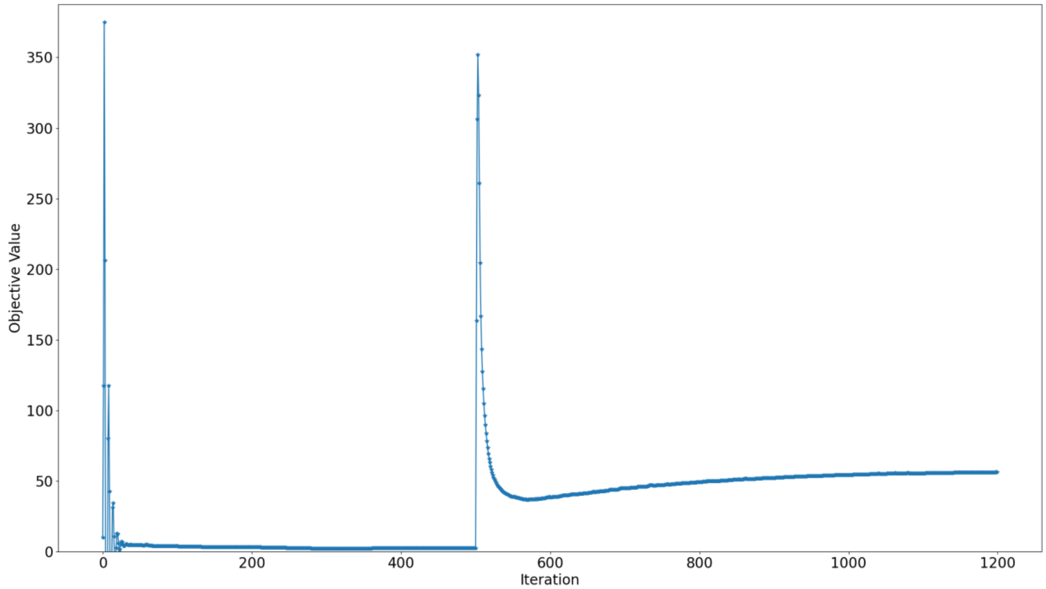
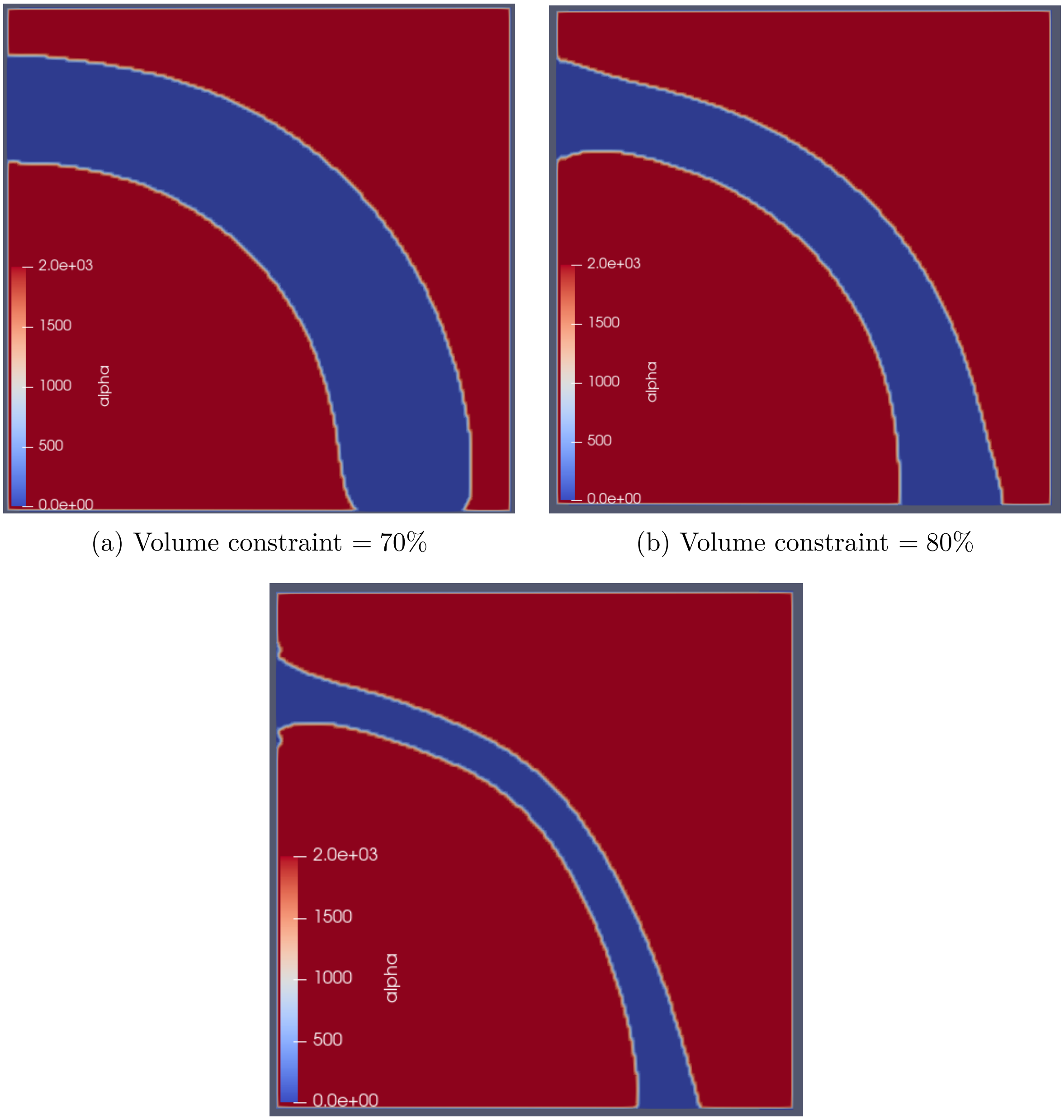


Figure 4.5: Objective function evolution in the case of a volume constraint of 80% applied at iteration 500.

Finally, we can illustrate the ability of the volume constraint parameter to accurately control the amount of material we seek in our design space. Below, we visualize the final domain in the cases where the constraint is set to 70%, 80%, 90%. It is apparent that the constraint is working as we observe the gradual increase of total material in the design domain.



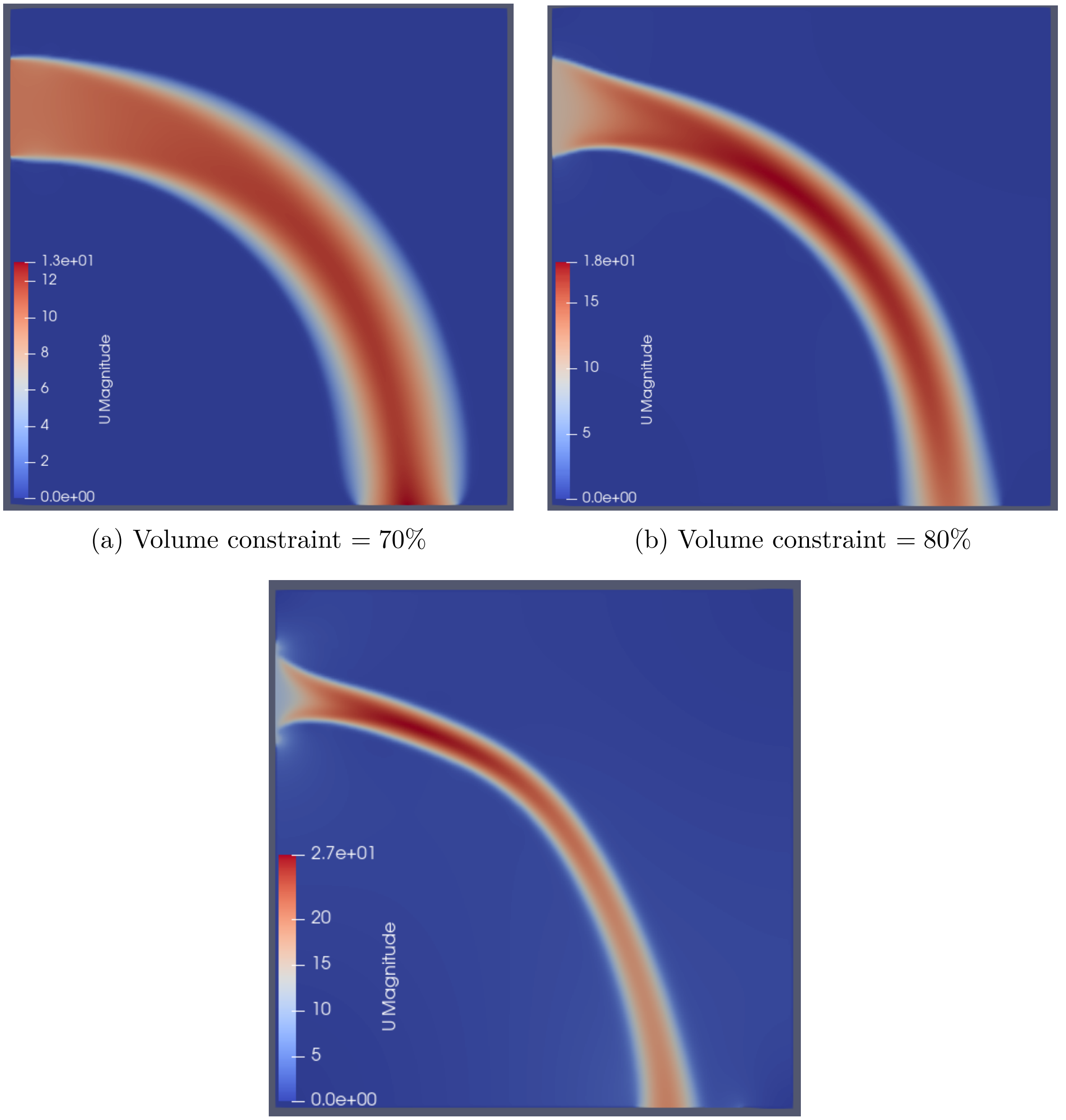
(a) Volume constraint = 70%

(b) Volume constraint = 80%

(c) Volume constraint = 90%

Figure 4.6: Material distribution of the geometry as the volume constraint is progressively increased.

In Fig. 4.7, the velocity profiles of the optimized cases are plotted. We observe, as expected, the maximum velocity increases as we increase the volume constraint on our optimized geometry (and so reduce the diameter of the created pipe).



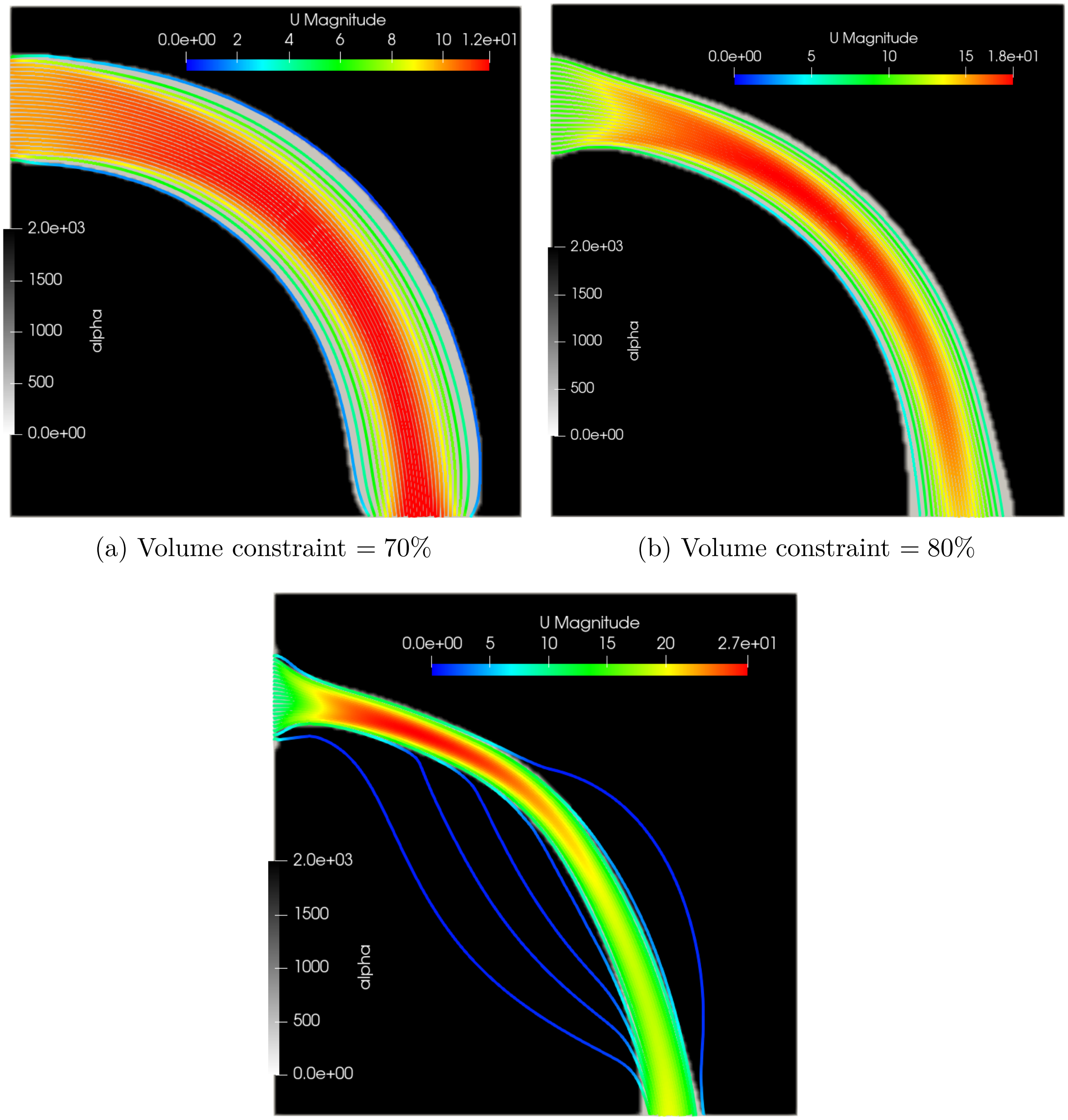
(a) Volume constraint = 70%

(b) Volume constraint = 80%

(c) Volume constraint = 90%

Figure 4.7: Velocity profile of the toy problem as the volume constraint is progressively increased.

Finally, in Fig 4.8 the streamlines for the problem are visualized and overlayed on the optimized geometry. In the case the volume constraint is set to 90%, we observe that some of the streamlines originating from the top and bottom edges of the inlet are strongly penalized suggesting the constraint is too harsh and should be reduced. In the case where the constraints are set to 80% and 90%, the streamlines nicely follow the contour of the optimized geometries.



(a) Volume constraint = 70%

(b) Volume constraint = 80%

(c) Volume constraint = 90%

Figure 4.8: Velocity streamlines of the toy problem overlaying the optimized topology.

4.2 Benchmarking Case 1: Duct System

In order for this solver to be applied to real problems, it is necessary to validate its results against cases published in the literature. The example here is based on the paper by Papoutsis-Kiachagias and Giannakoglou [7]. The case is as follows: we once again have a 1×1 box with the inlet placed on the left side of the domain and 2 outlets – one on the bottom part of the domain and another on the right part. The case is summarized in Fig. 4.9

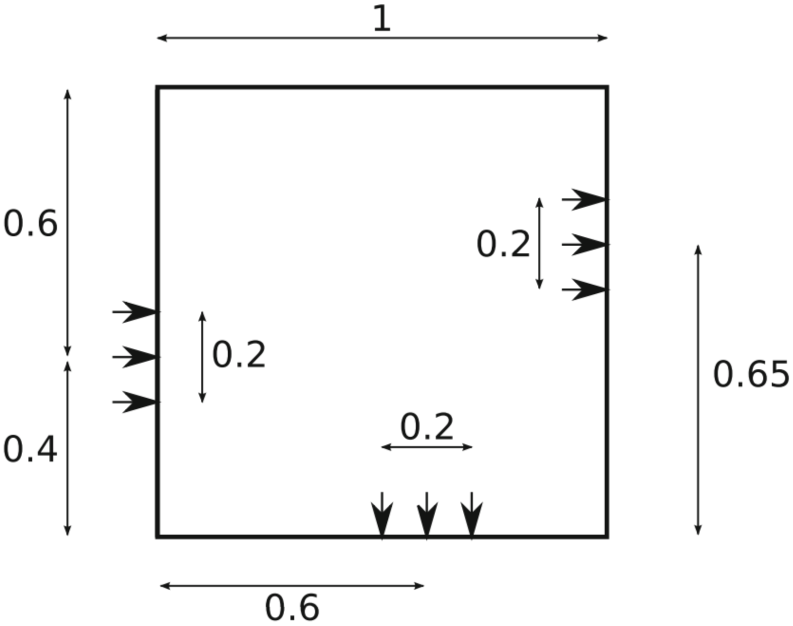
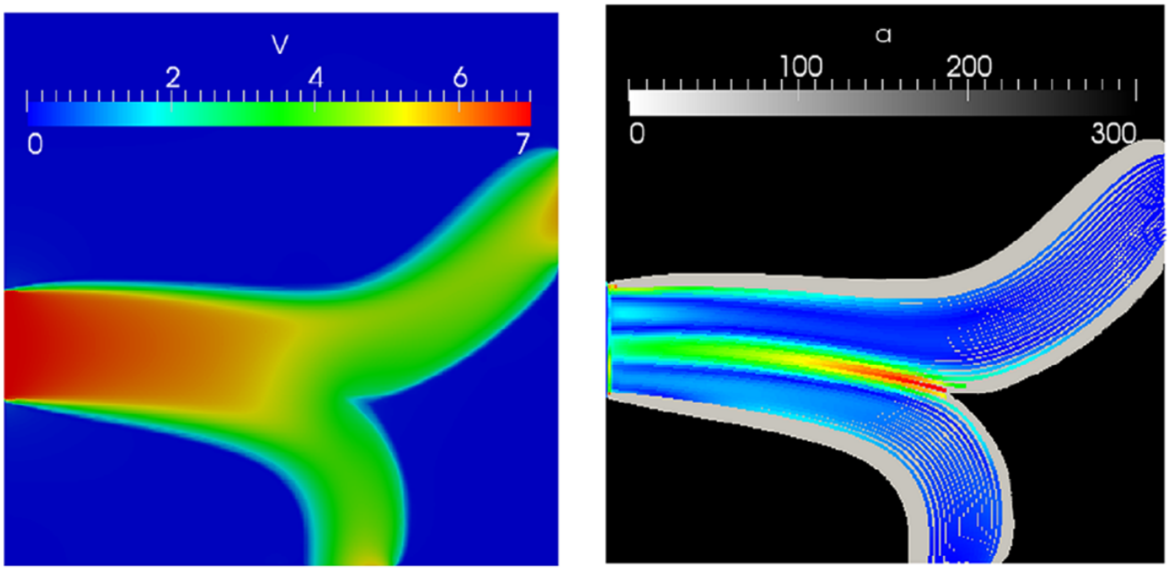


Figure 4.9: Topology optimization design space for benchmark case 1. A square domain with asymmetrically placed outlets and a single inlet. Image credit to [7].

The paper optimizes the domain with respect to two objective functions. The results from the implemented solver here are thus compared to the provided results for the power dissipation objective function so as to validate both the volume constraint and the power dissipation objective function implemented for the solver. The paper uses an adaptive approach to progressively refine the mesh where necessary – as this is not an option readily available to the author, a sufficiently fine, uniform, mesh (of approximately $45,000$ cells) is defined to allow the problem to converge and reproduce the macroscopic features we seek. Finally, the Reynolds number for this problem is 7000 and the volume constraint is set to 70%. The final results from the paper are displayed in Fig 4.10.

Before proceeding, it is necessary to make some comments about the image in Fig 4.10b: the streamlines displayed in the image are of the adjoint velocity. This was the only available image in the paper that had the optimized geometry visualized in the case of the power dissipation objective function, thus it is used here to communicate the final optimized geometry. In addition, the adjoint velocity calculated by the implemented solver and [7] are not comparable simply because the adjoint formulation used in the paper is different (for several reasons including the paper implements the augmented Lagrangian approach for the volume constraint), thus the calculated adjoint velocity will be different.



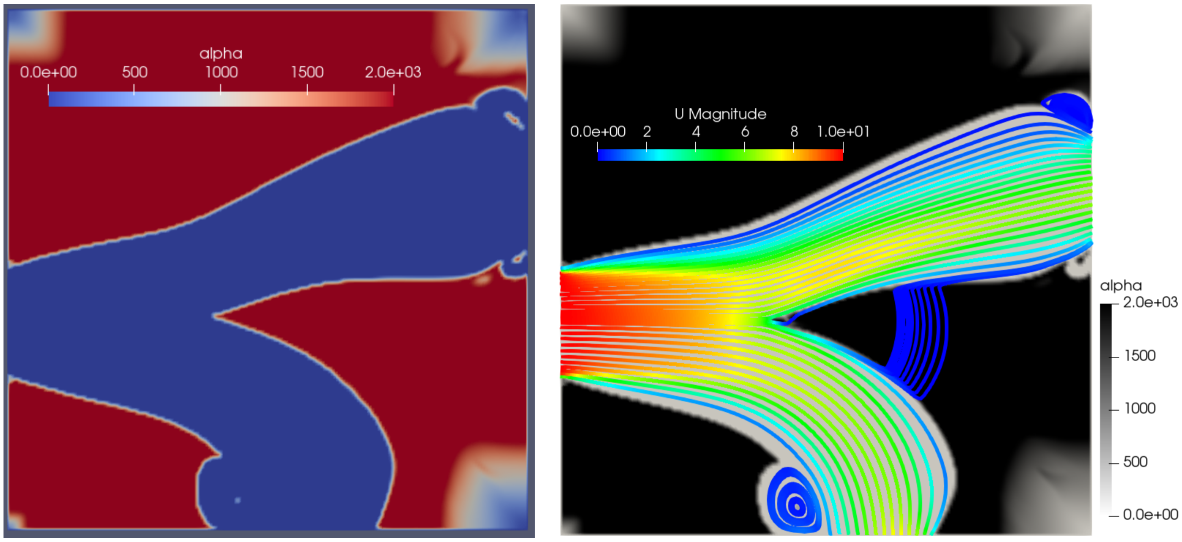
(a) Velocity profile

(b) Material distribution

Figure 4.10: Velocity profile and material distribution from published results in [7].

To showcase the development and accuracy of each stage of the solver, results will be progrssively displayed on the path to a correct result. First, we visualize the results of the standard topology solver default to OpenFOAM which uses pressure losses as an objective function. Next, are the results of the same solver, except with the addition of a volume constraint. Finally, the results of the correct objective function, power dissipation, are shown; these results will most closely match the results of the paper. In this approach we illustrate the correctness of both the volume constraint and the implementation of the power dissipation function.

In Fig. 4.11a, we have visualized the material distribution resulting from the use of the standard adjoint solver available in OpenFOAM. Once again, notice the “wispy” application of the porosity distribution and the odd artifacts near the outlets. Notice that these results do not at all match the published results shown in Fig. 4.10. In Fig 4.12c we display the velocity streamlines of the case and overlay them with the porosity distribution of the optimized geometry.

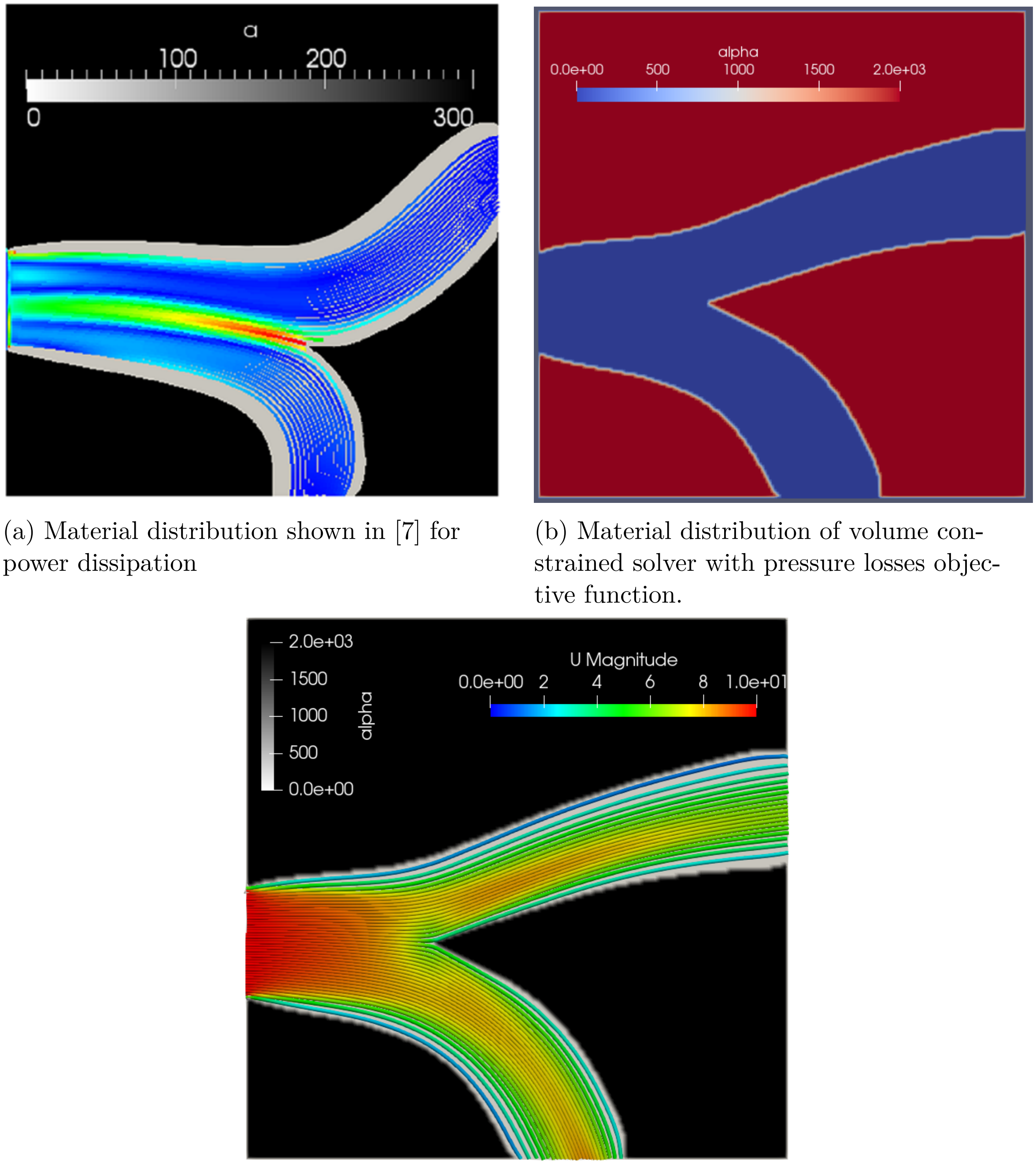


(a) Material distribution of benchmark case 1 using the default OpenFOAM topology optimization solver.

(b) Velocity streamlines overlayed on the material distribution

Figure 4.11: Output of the material distribution and velocity streamlines of benchmark case 1 using the currently available topology optimization solver in OpenFOAM.

Next, we improve this result by applying a volume constraint on the same solver (which uses pressure losses as an objective function). The results in Fig. 4.12 show that indeed the results have drastically improved, particularly in the "wispy" application of the porosity term. Notice, however, that there are obvious macroscopic differences between the published results and the output of the solver. This is expected given that the objective function used is pressure losses, in contrast with the published result which used power dissipation.



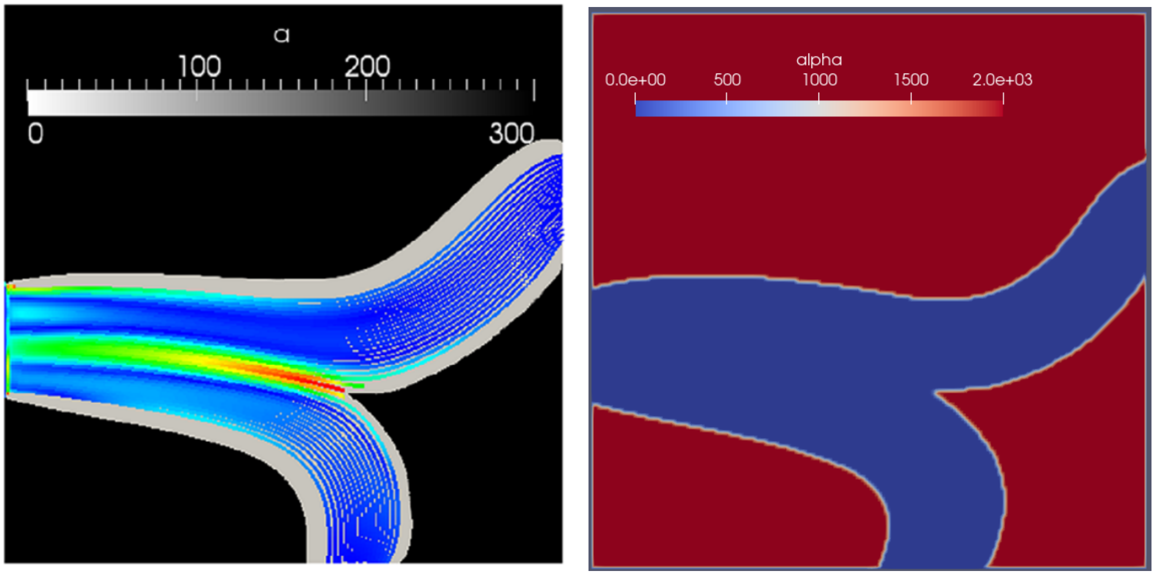
(a) Material distribution shown in [7] for power dissipation

(b) Material distribution of volume constrained solver with pressure losses objective function.

(c) Velocity streamlines overlaying the optimized porosity distribution.

Figure 4.12: Top row: Comparison of material distribution between the power dissipation case in [7] and the volume constrained solver using pressure losses as an objective function. Bottom: the velocity streamlines overlaying the optimal porosity distribution.

Finally, after implementing the power dissipation objective function, we produce the result shown in Fig. 4.13. Immediately, we observe the similarity of the macroscopic features between this result and the final result. Notice in particular the features near the outlets. For example, near the outlet on the right, we observe the similar continually upward slope on the bottom part of the created pipe.

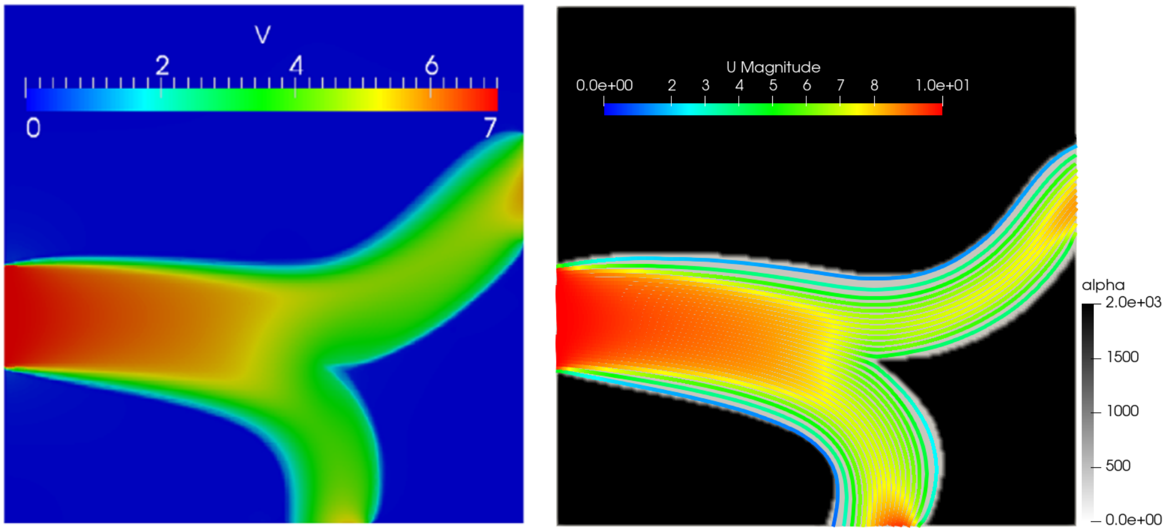


(a) Material distribution shown in [7] for power dissipation

(b) Material distribution of volume constrained solver with power dissipation objective function.

Figure 4.13: Comparison of material distribution between the power dissipation case in [7] and the volume constrained solver using power dissipation as an objective function

In Fig 4.14 we compare the velocity profile of this case from [7] to our solvers output by visualizing the velocity streamlines on the optimized geometry. We find the velocity profiles of the two cases have significant macroscopic similarities.



(a) Velocity profile from published results in [7]

(b) Velocity profile of volume constrained solver with power dissipation objective function.

Figure 4.14: Comparison of the velocity profile published in [7] to the implemented solver. Observe the almost perfect match between the profiles of the two cases. The mismatch in the magnitudes can be ignored as the paper [7] only specifies the Reynold number for the case. As such, the case run in this paper appropriately selects the inlet velocity and kinematic viscosity to match the Reynolds number.

4.3 Benchmarking Case 2: Duct System (Alternate Configuration)

This case is geometrically similar to the one above except that the inlet and outlets are extruded outward from the geometry. The example described here was first studied in the paper by Gkaragkounis, and Papoutsis-Kiachagias et al. [13], and was later extended to a density based topology optimization framework by Papoutsis-Kiachagias and Giannakoglou in [15]. The geometry is described in Fig. 4.15. This simulation is done in the laminar regime; the inlet velocity is prescribed to be 1 m/s and the kinetic viscosity is set to 0.001 m^2/s, thus resulting in a Reynolds number of 200. The objective function utilized is dissipated power. Finally, the volume percentage prescribed to be material is 53.8%.

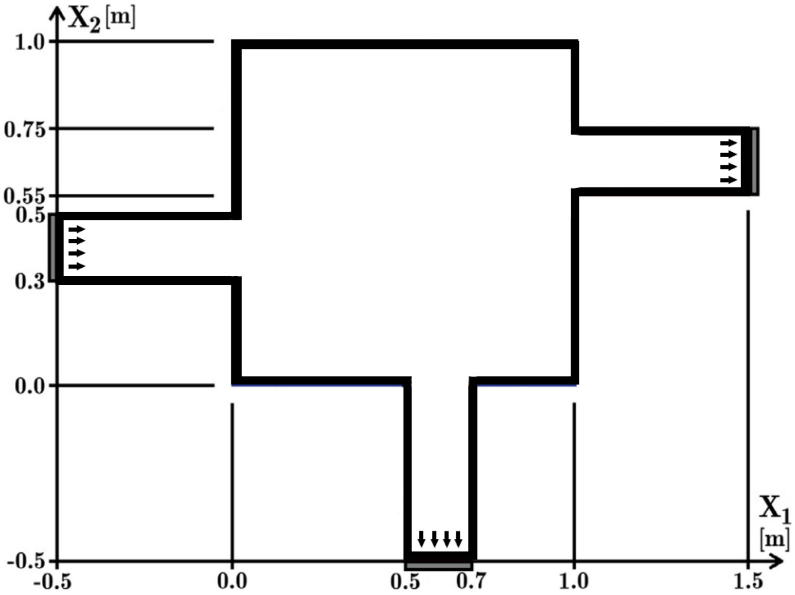
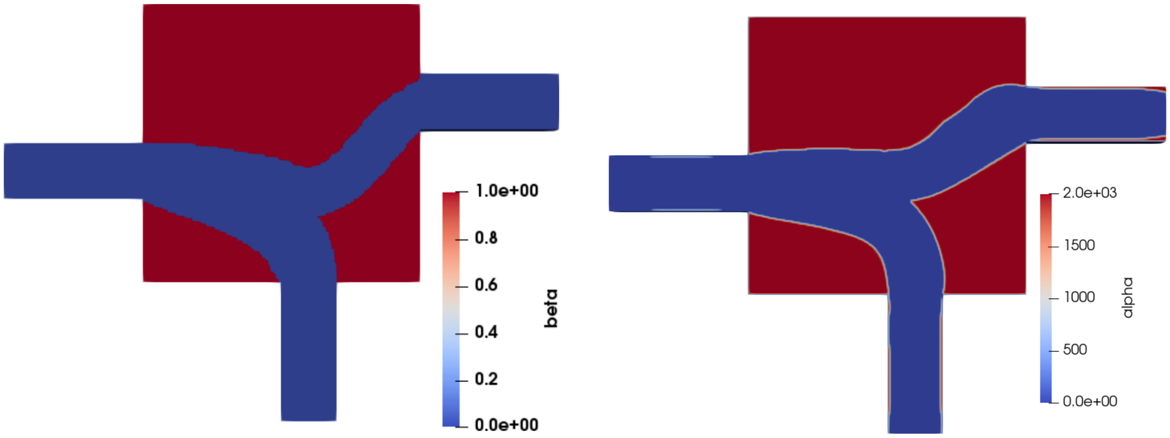


Figure 4.15: Problem description for benchmark case 2, an alternate configuration of the duct system described in the first benchmark. Image credits [15].

In Fig. 4.16 the material distribution of the result in [15] is compared with the result of the implemented solver. It is observed that the results are relatively similar by visual inspection. The implemented solver is able to recreate all the major macroscopic features of the benchmark case.

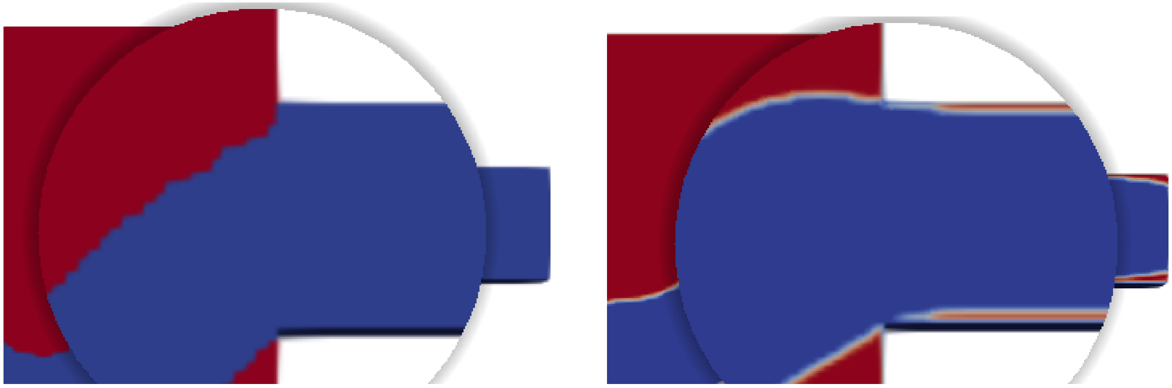


(a) Material distribution shown in [15]

(b) Material distribution of volume constrained solver with power dissipation objective function

Figure 4.16: Comparison of material distribution between results in [15] and the volume constrained solver developed in this paper

Observe that in the right outlet of the implemented solver there is non-zero material placed on the boundaries of the outlet, whereas in the outlet of the published case there is no material. This is visualized for clarity in zoomed in images of the right outlet shown in Fig 4.17. It is clear that the authors of [15] have restricted the accumulation of material in the outlets. There are a couple of possible ways this could be happening. One possible way is that the authors have an a priori approach that, as an implicit part of the model, excludes the placement of material on the boundary. This could account for the some of the minor differences in between the material distributions we see in Fig. 4.16. Alternatively, it is possible be that a post-processing step was taken to manually remove any material placed in the outlets to keep the main focus on the design space. Either way, it is clear that some step has been taken to ensure the final result does not have any material placed in the outlet – a clear giveaway of this is the sharp transition from material to non-material in the material distribution results of [15]. This is especially observed in the right outlet of Fig. 4.21a, to be discussed in the next section.



(a) Zoom in of right outlet of 4.16a.

(b) Zoom in of right outlet of 4.16b

Figure 4.17: Zoom in of the right outlets of Fig 4.16. Observe the outlet of the implemented solver, shown in b), had non zero material density.

Now the flow is visualized and compared between the two cases. We observe, for the flow, once again the macroscopic features are recreated.

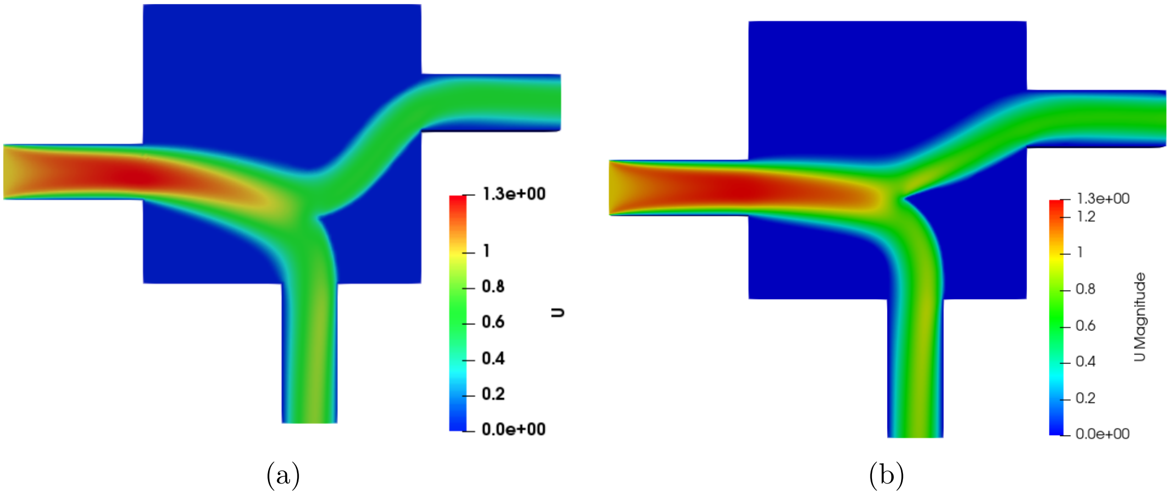


Figure 4.18: Comparison of velocity profiles between results in [15] and the volume constrained solver developed in this paper

Finally, in Fig. 4.19, the streamlines of velocity are visualized on top of the optimized porosity distribution.

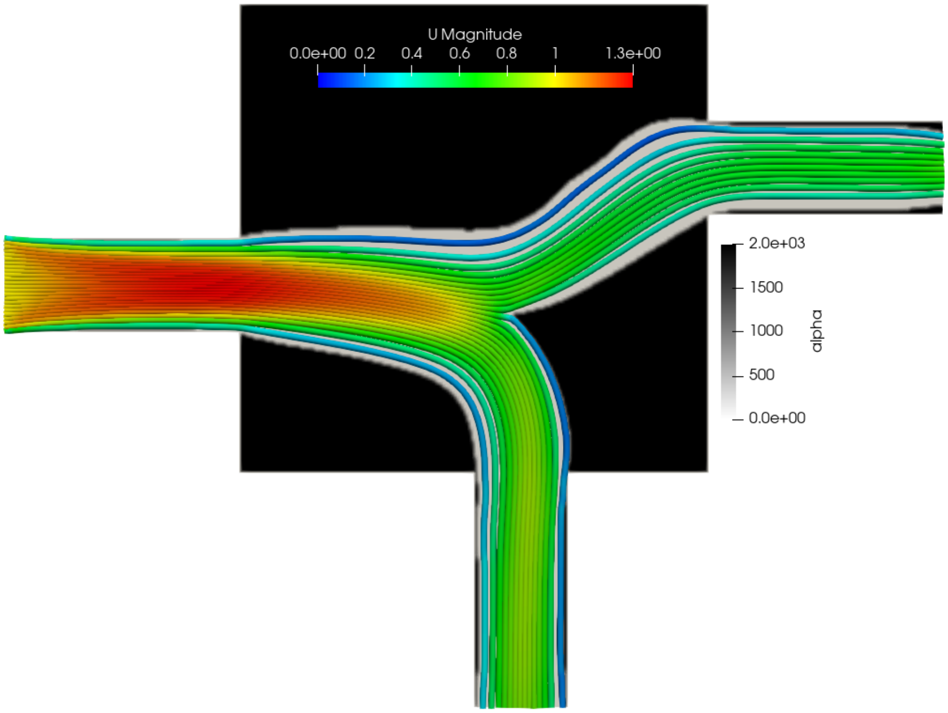


Figure 4.19: Streamlines overlaying the optimized geometry with material distribution

4.4 Benchmarking Case 3: Dual Bottleneck Duct

This case is once again described in the paper [15]. The geometry is visualized in Fig. 4.20. The flow properties are the same as the previous problem, that is, the inlet has a prescribed flow of 1 m/s and the kinematic viscosity is 0.001 m^2/s resulting in a Reynolds number of 200. Once again, the same objective function of dissipated power is minimized. The volume constraint is set such that 58.5% is occupied by material.

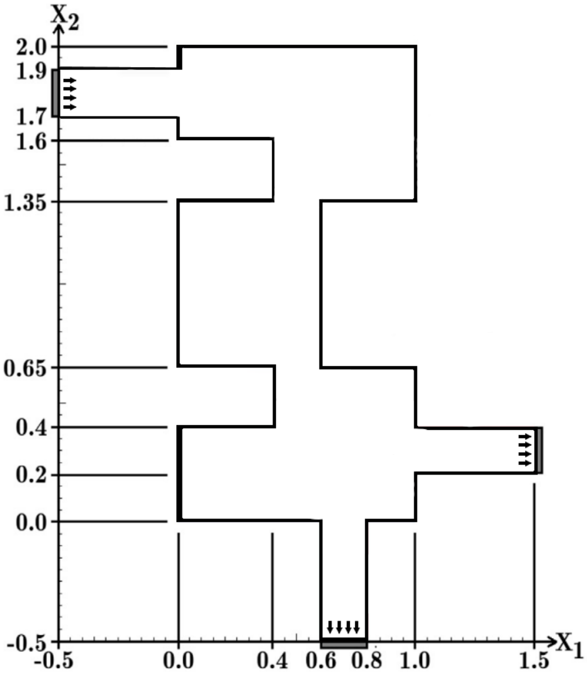
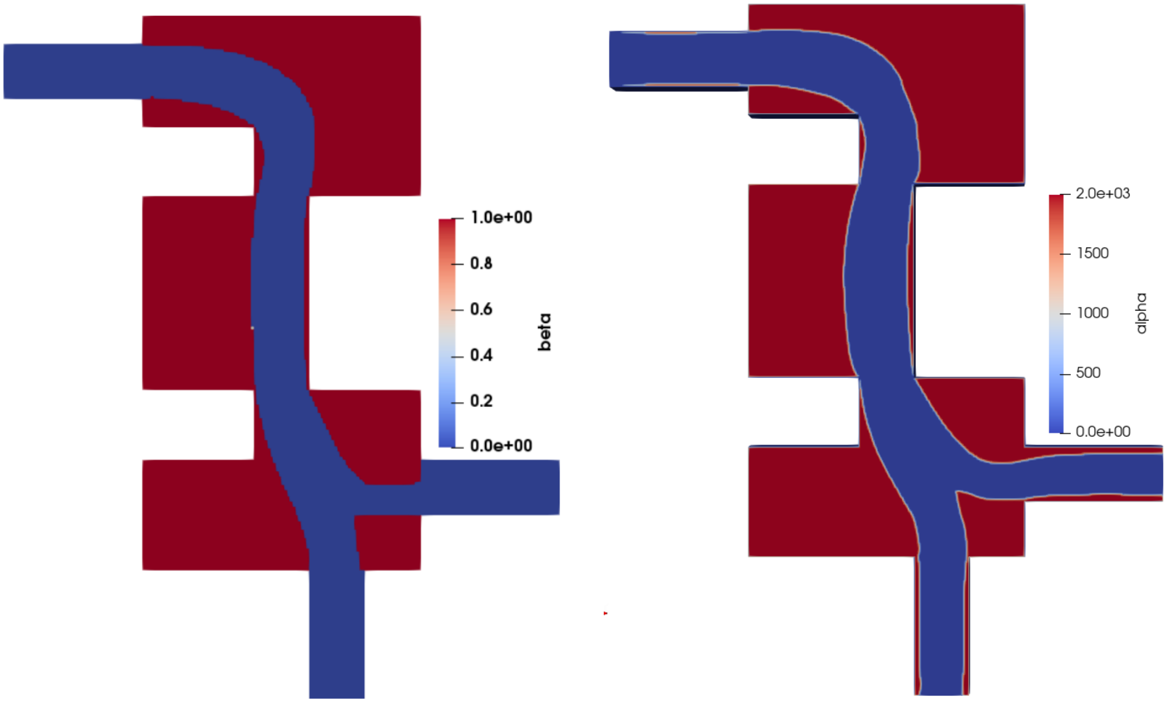


Figure 4.20: Problem description for benchmark case 3, a dual bottleneck duct system. Image credits [15].

In Fig. 4.21 the material distribution of the topology optimization result shown in [15] is compared to the results produced by the implemented solver. In this case we see significant similarities between the two results. The only major difference is the placement of material in the outlets of the implemented solver's solution in 4.21b compared to the published result's abrupt transition to non-material at the outlets 4.21a. This difference was addressed in the previous section.

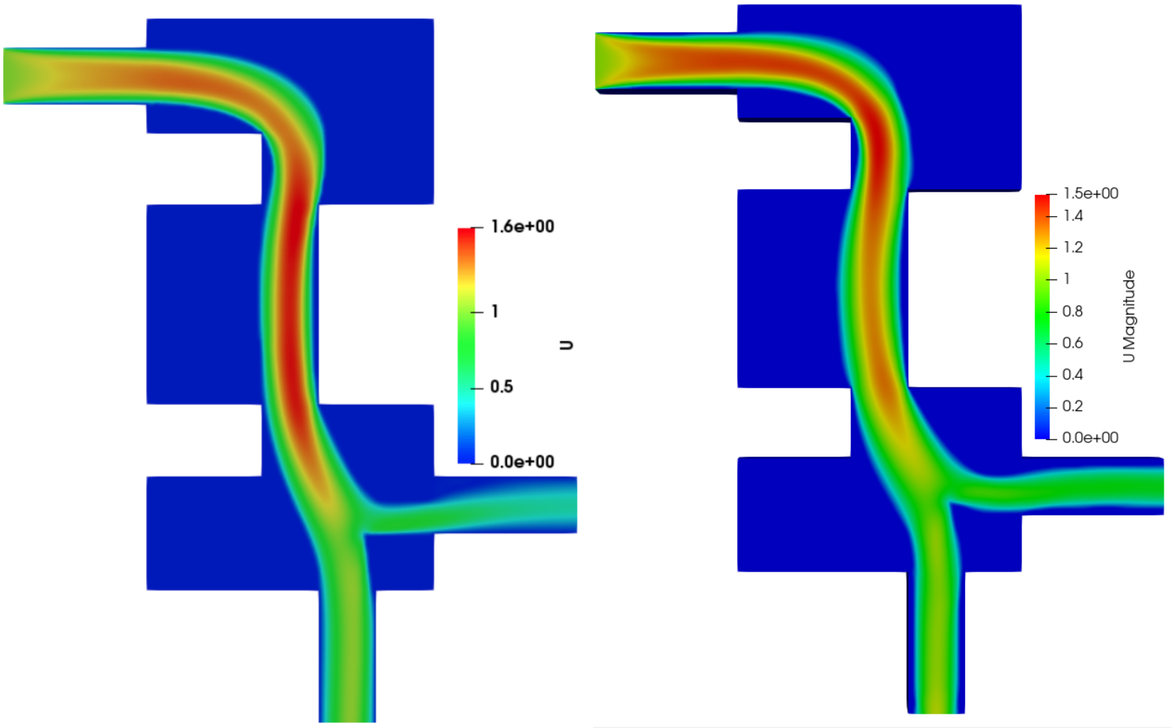


(a) Material distribution shown in [15]

(b) Material distribution of volume constrained solver with power dissipation objective function

Figure 4.21: Comparison of material distribution between results in [15] and the volume constrained solver developed in this paper

Next, the velocity distributions are visualized and we once again observe the significant similarities of the two results.



(a) Velocity profile shown in [15]

(b) Velocity profile of volume-constrained solver with power dissipation objective function

Figure 4.22: Comparison of velocity profiles between results in [15] and the volume-constrained developed solver in the case of the dual bottleneck duct with two outlets

Finally, the velocity streamlines of this case are overlayed on the final optimized geometry and visualized in 4.23.

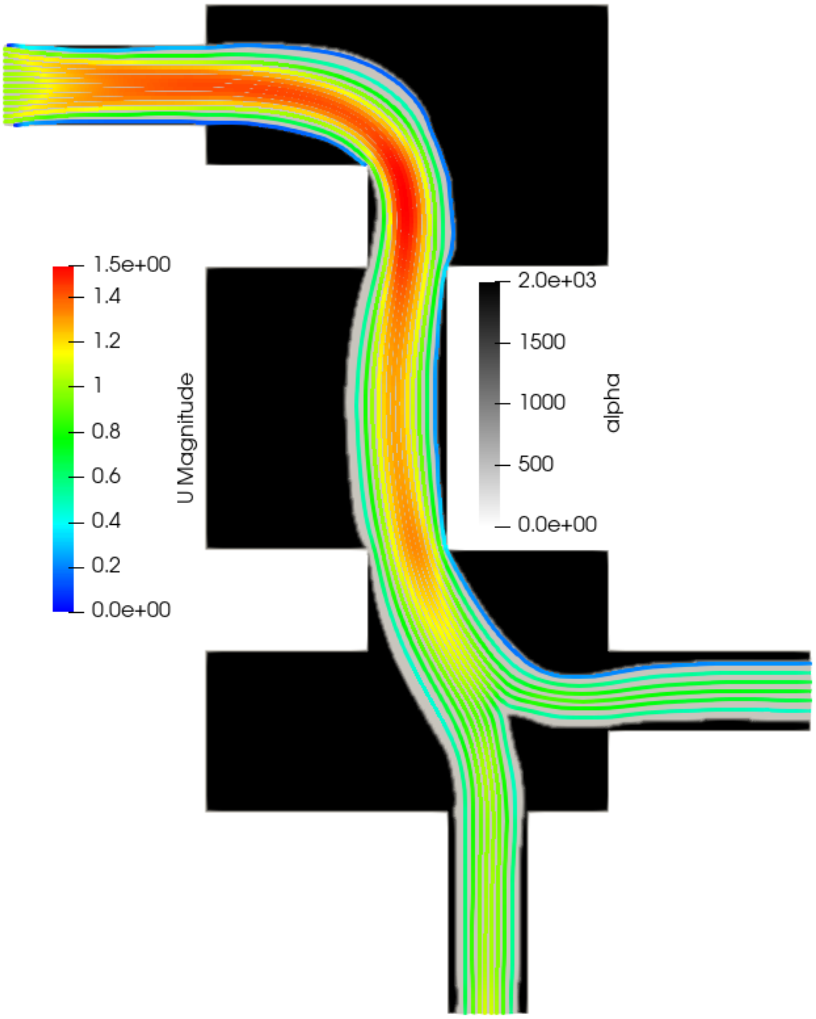


Figure 4.23: Streamlines overlaying the optimized geometry.

Chapter 5

Conclusion

In this work, we extended the capabilities of the OpenFOAM solver *adjointShapeOptimization*. This was done via two additions: first, the power dissipation objective function was programmed into the solver, then an explicit volume constraint was also added. Now the solver may be used to minimize the power dissipation in a geometry under a strict volume constraint, as was done in the numerical results chapter.

In the second and third chapter we laid down the theoretical foundation on which the solver is built. First, in chapter 2, we defined the problem our work sought to solve: the incompressible steady-state Navier-Stokes problem with the Darcy term. The Darcy term represents the porosity the density-based topology optimization implemented here. Then the problem is outlined in an optimization framework in order to minimize residuals representing the conservation of momentum equation and the incompressibility constraint. In chapter 3, we defined the Lagrangian to be minimized as an effective cost function in the optimization problem. Using this Lagrangian, we derived the sensitivity, adjoint equations, and adjoint boundary conditions for the incompressible steady-state Navier-Stokes equations. Then, these equations were specialized in the case of internal flows and the adjoint equations and adjoint boundary conditions were re-derived. Then the topology optimization framework was specified in the case of dissipated power by appropriately using the objective function to define the boundary conditions. Finally, to wrap up the theoretical section, the addition of an explicit volume constraint was discussed. Pseudocode describing this algorithm and a flowchart describing the implemented solver were also provided.

In chapter 4, we proved the fidelity of the implemented solver via several numerical examples. The first example is mainly to visualize and validate the explicit volume constraint used in the solver. A toy problem was constructed and solved while varying the volume constraint to illustrate the degree of control that the constraint has in defining the final geometry. In the next examples we found that the benchmark cases that we ran showed agreement with literature. Using the power dissipation objective function implemented in the solver, we were accurately able to recreate the macroscopic geometric and flow features of the studied case from [7] and [15].

5.1 Future Perspectives

There are a variety of ways in which the work done here can be extended. Certainly one of the most impactful ways would be with the addition of the energy equation to incorporate heat transfer in this solver with the addition of an objective function minimizing the heat retained in the domain. Doing so would mean taking meaningful steps in replicating the work of researchers such as Yaji et al. [49]. Applications of such a solver are particularly important in the current climate where heat minimization solvers are sought after to optimize liquid cooling systems such as those found on electric vehicle batteries. See Appendix E for a discussion of the energy equation.

In addition, on the numerical side, it may be beneficial to program alternative approaches to the volume constraint. Even though these methods are less robust, these methods are better designed and have more advanced update techniques [10] and thus it would be beneficial to see if there are any major differences in the results these alternate volume constraint implementations produce.

Another advancement can be made in addressing the differences between the final material placement in [15] and the cases run here. Figures 4.17 and 4.21 illustrated this difference: the solver implemented here generates non-zero material in the outlets whereas the solver used in [15] does not generate any material on the outlets. Even though the cases benchmarked here showed great agreement with the published results, addressing this challenge may yield some additional improvement.

Bibliography

[1] T. Borrvall and J. Petersson, “Topology optimization of fluids in Stokes flow,” Int. J. Numer. Methods Fluids, vol. 41, no. 1, pp. 77–107, 2003

[2] Othmer C, de Villiers E, Weller HG. Implementation of a continuous adjoint for topology optimization of ducted flows. AIAA-2007-3947, 2007.

[3] H. G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, COMPUTERS IN PHYSICS, VOL. 12, NO. 6, NOV/DEC 1998.
DOCUMENTATION: https://www.openfoam.com/documentation/guides/latest/api/adjointShapeOptimizationFoam_8C.html

[4] Othmer C, A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows, priv. commun., 2007

[5] Bradley, A.M., PDE-constrained optimization and the adjoint method, Creative Commons BY 4.0, 2019, https://cs.stanford.edu/~ambrad/adjoint_tutorial.pdf. Accessed: 17/11/2022

[6] Dwight, Richard and Brezillon, Joël. (2006). Effect of Approximations of the Discrete Adjoint on Gradient-Based Optimization. AIAA Journal. 44. 3022-3071. 10.2514/1.21744.

[7] Papoutsis-Kiachagias, E. M., Giannakoglou, K. C. (2016). Continuous Adjoint Methods for Turbulent Flows, Applied to Shape and Topology Optimization: Industrial Applications. Archives of Computational Methods in Engineering

[8] U. Nilsson, “Description of AdjointShapeOptimizationFoam and how to implement new cost functions,” 2013.

[9] Mosca, R.: Implementation of new boundary conditions for external flow adjoint-base shape optimization. In Proceedings of CFD with OpenSource Software, 2019, Ed

[10] Popovac, M. Continuous Adjoint Topology Optimization of Duct Flow Configurations with Explicit Volume Constraint for Design Variable Update. Energies 2022, 15, 7349. https://doi.org/10.3390/en15197349

[11] Papoutsis-Kiachagias, E.M.; Giannakoglou, K.C. An Adjoint-based Topology Optimization Framework for Fluid Mechanics and Conjugate Heat Transfer in OpenFOAM. In Proceedings of the 8th OpenFOAM Conference, Online, 13–15 October 2020

[12] Lee, Byung Joon and Liou, Meng-Sing and Kim, Chongam, Optimizing a Boundary-Layer-Ingestion Offset Inlet by Discrete Adjoint Approach, AIAA Journal, 48, 9, 2008-2016, 2010, 10.2514/1.J050222, https://doi.org/10.2514/1.J050222

[13] J.R.L. Koch, E.M. Papoutsis-Kiachagias, K.C. Giannakoglou,Transition from adjoint level set topology to shape optimization for 2D fluid mechanics, Computers & Fluids, Volume 150, 2017, Pages 123-138, ISSN 0045-7930

[14] Pironneau O (1984) Optimal shape design for elliptic systems. Springer, New York

[15] Papoutsis-Kiachagias, E.M.; Giannakoglou, K.C. An Adjoint-based Topology Optimization Framework for Fluid Mechanics and Conjugate Heat Transfer in OpenFOAM. In Proceedings of the 8th OpenFOAM Conference, Online, 13–15 October 2020.

[16] Aage N, Poulsen T, Gersborg-Hansen A, Sigmund O (2008) Topology optimization of large scale Stokes flow problems. Struct Multidiscip Optim 35:175–180

[17] Abdelwahed M, Hassine M, Masmoudi M (2009) Optimal shape design for fluid flow using topological perturbation technique. J Math Anal Appl 356:548–563

[18] Andersen C, Sigmund O (2011) Saturated poroelastic actuators generated by topology optimization. Struct Multidiscip Optim 43:693–706

[19] Andreasen C, Sigmund O (2013) Topology optimization of fluid–structure-interaction problems in poroelasticity. J Comput Methods Appl Mech Eng 258:55–62

[20] Bendsoe M, Kikuchi N (1988) Generating optimal topologies in structural design using a homogenization method. J Comput Methods Appl Mech Eng 71:197–224

[21] Kondoh T, Matsumori T, Kawamoto A (2012) Drag minimization and lift maximization in laminar flows via topology optimization employing simple objective function expressions based on body force integration. Struct Multidiscip Optim 45:693–701

[22] Bendsoe M, Sigmund O (2004) Topology optimization: theory, methods and applications. Springer, Berlin

[23] Bruns T (2007) Topology optimization of convection-dominated, steady-state heat transfer problems. Heat Mass Transf 50:2859–2873

[24] Challis V, Guest J (2009) Level set topology optimization of fluids in Stokes flow. Int J Numer Methods Eng 79:1284–1308

[25] Dede E (2010) Multiphysics optimization, synthesis, and application of jet impingement target surfaces. In: Thermal and thermomechanical phenomena in electronic systems (ITherm), 2010 12th IEEE intersociety conference, pp 1–7

[26] Deng Y, Liu Y, Wu J, Wu Y (2013) Topology optimization of steady Navier–Stokes flow with body force. J Comput Methods Appl Mech Eng 255:306–321

[27] Deng Y, Zhang P, Liu Y, Wu Y, Liu Z (2013) Optimization of unsteady incompressible Navier–Stokes flows using variational level set method. Int J Numer Methods Fluids 71(12):1475–1493

[28] Duan X, Ma Y, Zhang R (2008) Shape-topology optimization for Navier–Stokes problem using variational level set method. J Comput Appl Math 222:487–499

[29] Evgrafov A (2005) The limits of porous materials in the topology optimization of Stokes flows. Appl Math Optim 52:263–277

[30] Gersborg-Hansen A, Bendse M, Sigmund O (2006) Topology optimization of heat conduction problems using the finite volume method. Struct Multidiscip Optim 31:251–259

[31] Gersborg-Hansen A, Sigmund O, Haber R (2005) Topology optimization of channel flow problems. Struct Multidiscip Optim 30:181–192

[32] Guest J (2009) Imposing maximum length scale in topology optimization. Struct Multidiscip Optim 37:463–473

[33] Guest J, Prévost J (2006) Topology optimization of creeping fluid flows using a Darcy–Stokes finite element. Int J Numer Methods Eng 66:461–484

[34] Kreissl S, Maute K (2012) Level set based fluid topology optimization using the extended finite element method. Int J Numer Methods Eng 46:311–326

[35] Lions J (1971) Optimal control of systems governed by partial differential equations. Springer, New York

[36] Matsumori T, Kawamoto A, Kondoh T (2010) Topology optimization for fluid–thermal interaction problems. In: 6th China–Japan–Korea joint symposium on optimization of structural and mechanical systems, June 22–25, 2010, Kyoto, Japan, pp 1225–1233

[37] Maute K, Allen M (2004) Conceptual design of aeroelastic structures by topology optimization. Struct Multidiscip Optim 27:27–42

[38] Nocedal J, Wright S (1999) Numerical optimization. Springer, Berlin

[39] Olesen L, Okkels F, Bruus H (2006) A high-level programming-language implementation of topology optimization applied to steady-state Navier–Stokes flow. Int J Numer Methods Eng 65:975–1001

[40] Pironneau O (1984) Optimal shape design for elliptic systems. Springer, New York

[41] Seung-Hyun H, Seonho C (2005) Topological shape optimization of heat conduction problems using level set approach. Numer Heat Transf Part B Fundam 48:67–88

[42] Stanford B, Ifju P (2009) Aeroelastic topology optimization of membrane structures for micro air vehicles. Struct Multidiscip Optim 38:301–316

[43] Wiker N, Klarbring A, Borrvall T (2007) Topology optimization of regions of Darcy and Stokes flow. Int J Numer Methods Eng 69:1371–1404

[44] Yaji K, Yamada T, Yoshino M, Matsumoto T, Izui K, Nishiwaki S (2014) Topology optimization using the lattice Boltzmann method incorporating level set boundary expressions. J Comput Phys 274:158–181

[45] Yoon G (2010) Topological design of heat dissipating structure with forced convective heat transfer. J Mech Sci Technol 24(6):1225–1233

[46] Yoon G, Jensen J, Sigmund O (2007) Topology optimization of acoustic–structure interaction problems using a mixed finite element formulation. Int J Numer Methods Eng 70:1049–1075

[47] Zhou S, Li Q (2008) A variational level set method for the topology optimization of steady-state Navier–Stokes flow. J Comput Phys 227:10178–10195

[48] Zhuang C, Xiong Z, Ding H (2007) A level set method for topology optimization of heat conduction problem under multiple load cases. Comput Methods Appl Mech Eng 196:1074–1084

[49] Kentaro Yaji, Takayuki Yamada, Seiji Kubo, Kazuhiro Izui, Shinji Nishiwaki,A topology optimization method for a coupled thermal–fluid problem using level set boundary expressions, International Journal of Heat and Mass Transfer, Volume 81, 2015, Pages 878-888

[50] V. Subramaniam, T. Dbouk, J.-L. Harion, Topology optimization of conjugate heat transfer systems: A competition between heat transfer enhancement and pressure drop reduction, International Journal of Heat and Fluid Flow, Volume 75, 2019 ,Pages 165-184

Appendices

Appendix A

Derivation of Adjoint Equation and Adjoint Boundary Conditions

The derivation of the adjoint and adjoint boundary conditions are contained in this appendix.

Part I: Let us start with the easier of our two equations, Eq. 3.5 applied to δp:

$$\frac{\partial L}{\partial p}\,\delta p = \frac{\partial J}{\partial p}\,\delta p\ + \int_\Omega \boldsymbol{u}\cdot\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial p}\,\delta p\; d\Omega + \int_\Omega q\frac{\partial \mathcal{R}_p}{\partial p}\,\delta p\; d\Omega = 0. \tag{A.1}$$

We then calculate the derivatives of the state equations $\boldsymbol{\mathcal{R}_v}$ and $\mathcal{R}_p$ with respect to the pressure to find:

$$\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial p}\,\delta_p = \boldsymbol{\nabla}\delta p \qquad \& \qquad \frac{\partial \mathcal{R}_p}{\partial p}\,\delta_p = 0 \tag{A.2}$$

and substitute the above results into Eq. A.1. This yields:

$$\frac{\partial L}{\partial p}\,\delta p = \frac{\partial J}{\partial p}\,\delta p\ + \int_\Omega \boldsymbol{u}\cdot\boldsymbol{\nabla}\delta p\; d\Omega. \tag{A.3}$$

Integrating the second term by parts and separating the contributions of the objective function using the relation of Eq. 3.16 we obtain the following:

$$\frac{\partial L}{\partial p}\,\delta p = \int_\Gamma \left[\boldsymbol{u}\cdot\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial p}\right]\delta p\; d\Gamma + \int_\Omega \left[-\boldsymbol{\nabla}\cdot\boldsymbol{u} + \frac{\partial J_\Omega}{\partial p}\right]\delta p\; d\Omega = 0. \tag{A.4}$$

In general, these integrals must individually be identically zero in the domain for any $\boldsymbol{\delta p}$ that satisfies our primal equations. In the domain, this yields the adjoint equation

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = \frac{\partial J_\Omega}{\partial p} \quad \text{in} \quad \Omega \tag{A.5}$$

and in the boundary, we have the following adjoint boundary condition

$$\boldsymbol{u}\cdot\boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \quad \text{on} \quad \Gamma. \tag{A.6}$$

Part II: Now, let us consider equation 3.4 applied to $\delta\boldsymbol{v}$:

$$\frac{\partial L}{\partial \boldsymbol{v}}\,\delta v = \frac{\partial J}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v}\; + \int_\Omega \boldsymbol{u}\cdot\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v}\,d\Omega + \int_\Omega q\frac{\partial \mathcal{R}_p}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v}\,d\Omega. \tag{A.7}$$

where we use the following relations derived from Eqs. 2.4a and 2.4b

$$\frac{\partial \boldsymbol{\mathcal{R}_v}}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v} = (\delta\boldsymbol{v}\cdot\boldsymbol{\nabla})\,\boldsymbol{v} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\,\delta\boldsymbol{v} - \boldsymbol{\nabla}\boldsymbol{\cdot}\left\{2\nu\boldsymbol{\nabla}^S(\boldsymbol{\delta v})\right\} + \alpha\boldsymbol{\delta v} \tag{A.8}$$

$$\frac{\partial \mathcal{R}_p}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v} = -\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{\delta v}. \tag{A.9}$$

Note that the derivative calculated in Eq. A.8 above uses a constant eddy-viscosity assumption [6] or as it's more commonly know – the "frozen turbulence" approximation. In laminar flows this assumption, in fact, does not introduce errors and in turbulent flows this is a common approximation. However, even in the turbulent regime this is a reasonable approximation; to learn more about where it fails, see [6] or for additional details see [7]).

Then, substituting our results above into the previous equation, we obtain the following relation

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{v}}\,\delta v = {} & \frac{\partial J}{\partial \boldsymbol{v}}\,\delta\boldsymbol{v} \\ & + \int_\Omega \boldsymbol{u}\cdot\left[(\delta\boldsymbol{v}\cdot\boldsymbol{\nabla})\,\boldsymbol{v} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\,\delta\boldsymbol{v} - \boldsymbol{\nabla}\boldsymbol{\cdot}\left\{2\nu\boldsymbol{\nabla}^S(\boldsymbol{\delta v})\right\} + \alpha\,\boldsymbol{\delta v}\right] d\Omega \\ & - \int_\Omega q\boldsymbol{\nabla}\boldsymbol{\cdot}\boldsymbol{\delta v}\,d\Omega.\end{aligned} \tag{A.10}$$

Let's focus our attention on the third component of the second term:

$$\int_\Omega -\boldsymbol{u}\cdot\left[\boldsymbol{\nabla}\boldsymbol{\cdot}\left\{2\nu\boldsymbol{\nabla}^S(\boldsymbol{\delta v})\right\}\right] d\Omega \tag{A.11}$$

as it requires several integration by parts to obtain our final result. Integrating by parts once yields

$$\int_\Gamma -\boldsymbol{n}\cdot\left[2\nu\boldsymbol{\nabla}^s(\delta\boldsymbol{v})\right]\cdot\boldsymbol{u}\,d\Gamma + \int_\Omega \boldsymbol{\nabla u} : \left\{2\nu\boldsymbol{\nabla}^S(\boldsymbol{\delta v})\right\} d\Omega. \tag{A.12}$$

Integrating by parts again, we get

$$\int_\Gamma \left\{-\boldsymbol{n}\cdot\left[2\nu\boldsymbol{\nabla}^s(\delta\boldsymbol{v})\right]\cdot\boldsymbol{u} + 2\nu\boldsymbol{n}\cdot\boldsymbol{\nabla}^s\boldsymbol{u}\cdot\delta\boldsymbol{v}\right\}\,d\Gamma + \int_\Omega -\boldsymbol{\nabla}\boldsymbol{\cdot}(2\nu\boldsymbol{\nabla}^s\boldsymbol{u})\cdot\delta\boldsymbol{v}\,d\Omega. \tag{A.13}$$

Finally, integrating Eq. A.10 by parts and using the above result gives us:

$$\begin{aligned}\frac{\partial L}{\partial \boldsymbol{v}}\,\delta v = &\int_\Gamma \left[\boldsymbol{n}(\boldsymbol{u}\cdot\boldsymbol{v}) + \boldsymbol{u}(\boldsymbol{v}\cdot\boldsymbol{n}) + 2\nu\boldsymbol{n}\cdot\boldsymbol{\nabla}^s\boldsymbol{u} - q\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial \boldsymbol{v}}\right]\cdot\delta\boldsymbol{v}\,d\Gamma \\ &- \int_\Gamma \boldsymbol{n}\cdot[2\nu\boldsymbol{\nabla}^s(\delta\boldsymbol{v})]\cdot\boldsymbol{u}\,d\Gamma \\ &+ \int_\Omega \left[-\boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{v} - (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\cdot(2\nu\boldsymbol{\nabla}^s\boldsymbol{u}) + \alpha\boldsymbol{u} + \boldsymbol{\nabla}q + \frac{\partial J_\Omega}{\partial \boldsymbol{v}}\right]\cdot\delta\boldsymbol{v}\,d\Omega = 0\end{aligned} \tag{A.14}$$

This equation is valid for any δv that satisfies our state equations, only if we require the integrals are individually zero. Thus we obtain the following adjoint NS equations from the above:

$$-(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{u} - \boldsymbol{\nabla}\boldsymbol{u}\cdot\boldsymbol{v} + \boldsymbol{\nabla}q - \boldsymbol{\nabla}\cdot(2\nu\boldsymbol{\nabla}^s\boldsymbol{u}) + \alpha\boldsymbol{u} = -\frac{\partial J_\Omega}{\partial \boldsymbol{v}} \quad \text{in} \quad \Omega \tag{A.15}$$

This in addition to Eq. A.5:

$$\boldsymbol{\nabla}\cdot\boldsymbol{u} = \frac{\partial J_\Omega}{\partial p} \quad \text{in} \quad \Omega$$

define our full set of adjoint equations.

We also obtain the following adjoint boundary condition

$$\begin{aligned}&\int_\Gamma \left[\boldsymbol{n}(\boldsymbol{u}\cdot\boldsymbol{v}) + \boldsymbol{u}(\boldsymbol{v}\cdot\boldsymbol{n}) + 2\nu\boldsymbol{n}\cdot\boldsymbol{\nabla}^s\boldsymbol{u} - q\boldsymbol{n} + \frac{\partial J_\Gamma}{\partial \boldsymbol{v}}\right]\cdot\delta\boldsymbol{v}\,d\Gamma \\ &\qquad\qquad - \int_\Gamma \boldsymbol{n}\cdot[2\nu\boldsymbol{\nabla}^s(\delta\boldsymbol{v})]\cdot\boldsymbol{u}\,d\Gamma = 0\end{aligned} \tag{A.16}$$

This taken in addition to Eq A.6:

$$\boldsymbol{u}\cdot\boldsymbol{n} = -\frac{\partial J_\Gamma}{\partial p} \quad \text{on} \quad \Gamma.$$

define our set of adjoint boundary conditions.

Appendix B

Alternative Approach to The volume Constraint

In order to minimize our objective with a constraint on the total volume occupied by the porosity function α, we need to account for the additional constraint by modifying our original Lagrangian L defined in Eq. 3.1. The augmented Lagrangian is labeled L_{aug} and can be written as follows:

$$L_{\text{aug}} = L - \lambda_{vol} c_{\text{vol}} + w c_{\text{vol}}^2 \tag{B.1}$$

Where c_{vol} is the volumetric constraint we wish to fulfill; λ_{vol} is the corresponding Lagrange multiplier and w is a scalar weight. Now our goal is to reduce L_{aug} instead of L. Then according to the literature, we can define the volumetric constraint as follows ([7]):

$$c_{\text{vol}} = \left\{ \frac{\int_\Omega \left(1 - \frac{\alpha}{\alpha_{\max}}\right) \, d\Omega}{\int_\Omega \, d\Omega} - V_{\text{target}} \right\}^2 \tag{B.2}$$

Here V_{target} is our target volume, α is the Darcy porosity term, and $\alpha_{\max}$ is the maximum selected value of the porosity.

Note that c_{vol} has absolutely no dependence on the primary variables p and $\boldsymbol{v}$. As such, our adjoint and adjoint boundary equations do not change at all! The only piece necessary to re-derive is the sensitivity. That is:

$$\frac{\partial L_{\text{aug}}}{\partial \alpha} = \frac{\partial L}{\partial \alpha} - \lambda_{vol} \frac{\partial c_{\text{vol}}}{\partial \alpha} + 2 w c_{\text{vol}} \frac{\partial c_{\text{vol}}}{\partial \alpha} \tag{B.3}$$

where we use the fact

$$\frac{\partial c_{\text{vol}}}{\partial \alpha} = 2 \left\{ \frac{\int_\Omega \left(1 - \frac{\alpha}{\alpha_{\max}}\right) \, d\Omega}{\int_\Omega \, d\Omega} - V_{\text{target}} \right\} \left\{ \frac{\int_\Omega \left(-\frac{1}{\alpha_{\max}}\right) \, d\Omega}{\int_\Omega \, d\Omega} \right\} \tag{B.4}$$

or when simplified

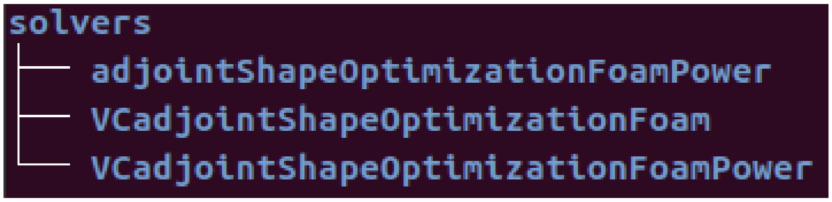
$$\frac{\partial c_{\text{vol}}}{\partial \alpha} = -\frac{2}{\alpha_{\max}} \left\{ \frac{\int_\Omega \left(1 - \frac{\alpha}{\alpha_{\max}}\right) \, d\Omega}{\int_\Omega \, d\Omega} - V_{\text{target}} \right\}. \tag{B.5}$$

Thus combining the result above with Eq. B.3 we arrive at the modification of the sensitivity equation necessary in the case of a volume constraint.

Appendix C

Tutorial

In the course of this work, there were several solvers written. Each of these is visualized below. The development of each solver was essentially a step on the way to developing the final solver: a volume constrained solver which uses power dissipation. However, each solver has some degree of utility and each was used to visualize at least one figure in this paper, as such their inclusion is important here to be able to recreate all the results.



```
solvers
├── adjointShapeOptimizationFoamPower
├── VCadjointShapeOptimizationFoam
└── VCadjointShapeOptimizationFoamPower
```

Figure C.1: Solvers written and used for the completion of this thesis. VC → VolumeConstrained.

adjointShapeOptimizationFoamPower In this solver, the power dissipation objective function is implemented. For a thorough review of the details of the implementation, see [8]. That implementation, however, has some lines of code that use deprecated OpenFOAM commands and so will not run on OpenFOAM versions available now. The current implementation is updated to run on modern versions of OpenFOAM.

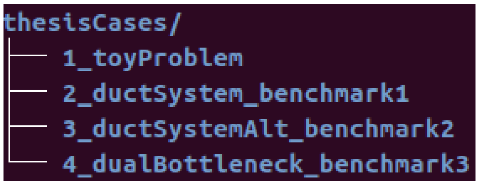
VCadjointShapeOptimizationFoam This solver uses the standard OpenFOAM solver as a base and expands it with the addition of a volume constraint. The objective function for this solver is pressure losses, see [3].

VCadjointShapeOptimizationFoam This combines the two previous solvers; it minimizes power dissipation as a cost function while under a volume constraint.

To compile all the solvers, simply enter the solver directory and run the necessary bash script in the following way:

```
$ ./compileSolvers
```

In my final implementation of both volume constrained solvers, I added two important user defined parameters. The first parameter is ***volumeFraction*** – this parameter defines what percent of the volume you wish to be occupied by material. For example a value of 0.7 will set the volume constraint such that 70% of the total volume will be occupied by material. The other important parameter is: ***iterationToEnforceVolumeConstraint***. Simply, this specifies the iteration of the solver where the volume constraint starts being applied.



```
thesisCases/
├── 1_toyProblem
├── 2_ductSystem_benchmark1
├── 3_ductSystemAlt_benchmark2
└── 4_dualBottleneck_benchmark3
```

Figure C.2: Directory of all problems run for this thesis.

The directory above includes all cases run for this thesis. All the cases can be run in one-shot by entering the directory and running the following command

```
$ ./runAllCases
```

To run new cases simply use the appropriate case folder and modify it for the purposes you need.

Appendix D

Objective Function Evolution Script

In addition to the mathematical changes that were made to the the OpenFoam solver, the output of the code was modified to display the value of the objective function at each iteration of the calculation. Thus, the value of the objective function can be used to track the state and stability of the optimizer. For simplicity, a bash script was written tied to python code to automatically visualize the objective function as it evolves through the calculation. To use the script, simply enter the case file of a simulation that has been run to completion and execute the visualizer script using the following command:

```
$ ./visualize
```

Appendix E

Addition of the Energy Equation

The purpose of this addition is to ease the development of a solver optimizing conjugate heat transfer problems. I have implemented the energy equation so that the solver can now solve for temperature as an inherent part of the solution. The job of the next researcher will simply be to expand on this development by implementing the adjoint equations and adjoint boundary conditions for a thermal-fluid governed by the incompressible, steady state Navier-Stokes equations with the addition of the energy equation. This will allow optimizing topologies for heat based objective functions. In this development, the work of Subramaniam, Dbouk et al. [50] will be a valuable resource.

E.1 Problem Statement

The governing equations for the coupled thermal-fluid problem are stated below for a domain of interest Ω:

$$-\boldsymbol{\nabla}\cdot\boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{E.1a}$$

$$(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v} + \boldsymbol{\nabla}p - \boldsymbol{\nabla}\cdot\left\{2\nu\boldsymbol{\nabla}^S\boldsymbol{v}\right\} + \alpha\boldsymbol{v} = 0 \qquad \text{in } \Omega, \tag{E.1b}$$

$$(\boldsymbol{v}\cdot\boldsymbol{\nabla})T - K\nabla^2 T = 0 \qquad \text{in } \Omega, \tag{E.1c}$$

where the new variables K and T represent the thermal diffusivity coefficient and temperature respectively.

In addition to the mass (Eq. E.1a), momentum (Eq. E.1b), and energy (Eq. E.1c) equations, we require appropriate boundary conditions to complete the problem statement. On the Dirichlet boundary we have

$$\begin{cases} \boldsymbol{u} = \boldsymbol{u}_{\mathrm{D}}, \\ T = T_D, \end{cases} \quad \text{on } \Gamma_{\mathrm{D}}. \tag{E.2}$$

where T_D is constant value representing the prescribed temperature on the Dirichlet boundary. On the Neumann boundary the conditions are:

$$\begin{cases} (-p\boldsymbol{I} + 2\nu\boldsymbol{\nabla}^S\boldsymbol{v})\boldsymbol{n} = \boldsymbol{q}_N, \\ -n \cdot \boldsymbol{\nabla}T = 0 \end{cases} \quad \text{on } \Gamma_\mathrm{N}. \tag{E.3}$$

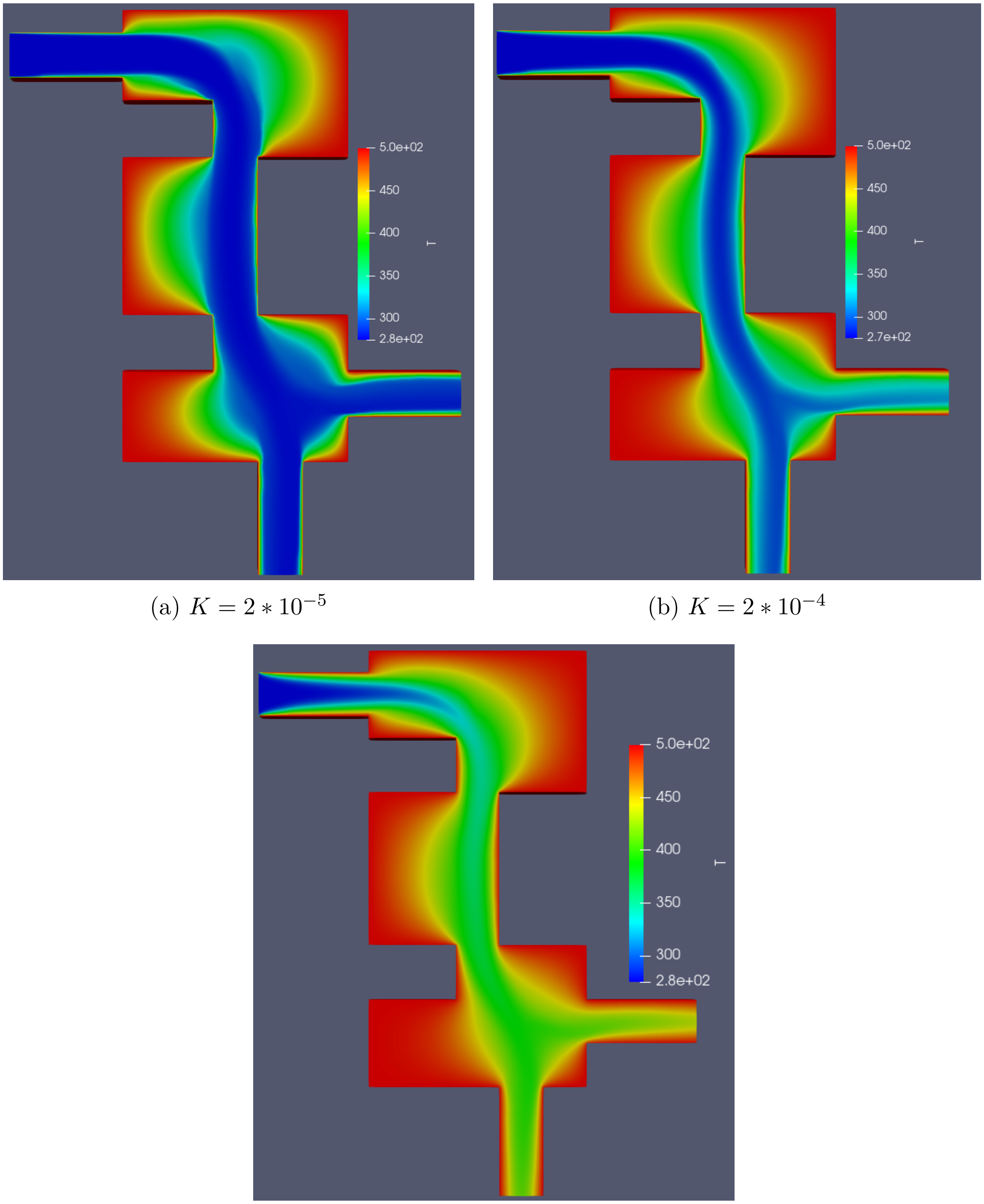
Thus with Eqs. E.1, E.2, and E.3 we define the set equations to model a coupled thermal-fluid problem.

E.2 Results

The model described above was implemented in OpenFOAM as an addition to the solver described in the thesis which minimizes power dissipation. There is no optimization being done in regard to temperature, however, this solver will form the basis for that development. Some preliminary results are shown below.

The geometry studied is the dual bottleneck duct, the same as the third benchmarking case shown in 4.20. The flow problem is also kept the same with the Reynolds number being 200. Thus it is important to keep in mind that the optimized geometry and flow are the same as shown in Fig. 4.23. Now, however, we specify some thermal boundary conditions: the inlet is prescribed to flow in with a cold fluid and the walls are set to a high temperature. Thus, we expect the heat to diffuse into the geometry from the walls and heat the colder fluid. The fluid temperature of the inflow is set to 280 Kelvin and the walls are set to 500 Kelvin.

In Fig. E.1 we plot the solution as the thermal diffusivity constant is varied. Recalling that for low values of the diffusivity constant we expect less temperature exchange between objects; we have a sharp temperature gradient i.e. in such a case we expect the fluid to remain cold. At high values of the thermal diffusivity constant, we expect temperature exchange to be higher; we have a a much more even temperature distribution i.e. we expect our fluid to be more warmed. We see in Fig E.1a that for the low value of the diffusivity constant the fluid remains at relatively the same temperature and there is not much heat exchange. Increasing this value by an order of magnitude we see in Fig. E.1b that heat is being diffused into the fluid at a higher level, but the temperature at core of the fluid stays relatively constant. Finally, increasing the thermal diffusivity constant by yet another order of magnitude we see in Fig E.1c that heat has much more efficiently diffused into the fluid thus increasing the fluids temperature significantly by the time it reaches the two outlets.



(a) $K = 2 * 10^{-5}$

(b) $K = 2 * 10^{-4}$

(c) $K = 2 * 10^{-3}$

Figure E.1: Solution of the coupled thermal-fluid problem on the optimized geometry shown in 4.23 as the thermal diffusivity constant is varied.