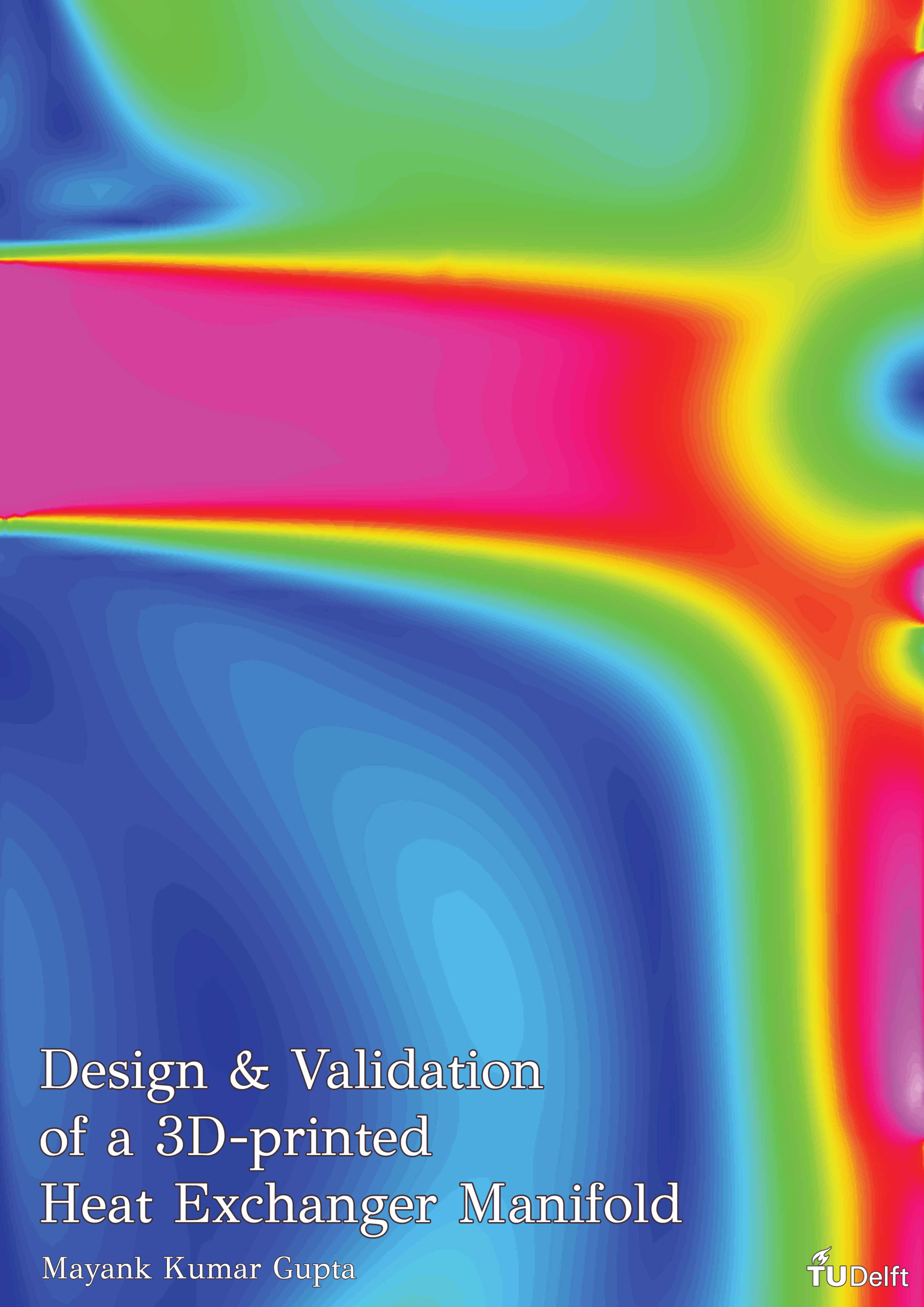


Design & Validation of a 3D-printed Heat Exchanger Manifold

Mayank Kumar Gupta

Design & Validation of a 3D-printed Heat Exchanger Manifold

by

Mayank Kumar Gupta

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	5470013
Project duration:	December 1, 2022 – August 25, 2023
Supervisor(s):	Prof. dr. ir. K. Hooman, TU Delft
	Dr. L. Arnaud, Université de Toulouse (External)

TUDelft

Acknowledgements

First of all, I'd like to thank Dr. ir. Kamel Hooman for his supervision throughout this project. His willingness to make time for discussions at short notice, and the amount of thought he puts into his advice has been crucial in my success. I would also like to thank Dr. Lionel Arnaud. While we have not had as regular a communication, his help with getting up to speed with previous work and rectifying problems in the resin prints has also been crucial. I would also like to thank Dr. Renato Picelli and Lucas Siqueira, for letting me use not only the optimisation methodology developed by them, but also their code. Their painstaking help in building on the optimization code made possible the exploration of so many optimization cases.

The experimentation methodology was unique due to the constraints of time and resources. It is Bart who helped me navigate these limitations, without whom I would not have been able to set things up and actually perform the validation. I would also like to thank Simone for teaching me to use the resin printer to manufacture my models, and answering my doubts through several initial prototypes. I would also like to thank Pawel for providing the microscope images of the surface of the print. I cannot also go without acknowledging the contributions of Heng, Jelle, Ankit, and Dennis. COMSOL can be a finicky software sometimes by itself, to say nothing of using it with the DelftBlue supercomputer. It is with their help that I was able to figure it out.

I would also like to acknowledge the support of my friends in the flex room - Fathaah, Thomas, Vignesh, Daniel, Jose, Akhilesh, and Panji. Together we worked on our theses and had endless discussions over copious amounts of free coffee. It surely made what is supposed to be a solo project seem like a more social endeavour. Lastly, I would like to thank my friends and family for not only their support during the thesis, but also through the process of moving to a new country and successfully graduating from this challenging study program.

Mayank Kumar Gupta
Delft, August 2023

Abstract

Metal additive manufacturing has enabled great diversity and design freedom in heat exchanger design. However, these benefits cannot be adequately realised without optimizing its ancillary components as well, specifically the inlet and outlet manifolds. Optimizing the manifolds can significantly improve the flow distribution inside the heat exchanger core. This in turn will improve the heat exchange performance and reduce flow obstruction. More importantly, it can reduce the size of the setup, which makes for much cheaper and faster manufacturing. Literature on optimization of flows is quite vast and extensive, but experimental validation is lacking. It is not well understood whether 2D optimized parts hold up to 3D validation, and whether it differs for laminar and turbulent regimes. In this work, a manifold is numerically optimized in 2D and 3D for various flow regimes including both laminar and turbulent flow. The ideal set of assumptions for each specific case is then prescribed based on the results. Validation is performed for some of the geometries after manufacturing them via 3D-printing.

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AM	Additive Manufacturing
CFD	Computational Fluid Dynamics
GT	Geometry Trimming
ILP	Integer Linear Programming
SIMP	Solid Isotropic Material with Penalization
TO	Topology Optimization
TOBS	Topology Optimization of Binary Structures
TPMS	Triply Periodic Minimal Surface
...	

Symbols

Symbol	Definition	Unit
d	Width	[mm]
ED	Energy Dissipation	[m^2/s^2]
h	Height	[m]
$\mathbf{K}$	Viscous/Deviatoric Stress Tensor	
k	Turbulent Kinetic Energy	
m	Mass-flow Rate	[kg/s]
MF	Maldistribution Factor	
Nu	Nusselt Number	
Re	Reynolds Number	
t	Time	[s]
u	Velocity	[m/s]
V	Volume	
w	Weight	
$\mathbf{x}$	Design Variables	
α	Density	
β	Element Flip Limit	
ϵ	Turbulent Dissipation Rate	
ε	Constraint Relaxation Parameter	

Symbol	Definition	Unit
ι	Convergence tolerance	
$\mathbf{\Lambda}$	Lagrange Multiplier Vector	
μ	Dynamic Viscosity	[Pa s]
ρ	Density	[kg/m^3]
Ω	Flow Domain	
ω	Specific Dissipation Rate	

Chapter 1

Introduction

A variety of heat exchangers are used across multiple industries as varied as medicine, transportation, shipping, manufacturing, agriculture, food and drinks, aerospace and war. These range in size from multiple metres to sub-millimetre scales. All have the same objective: to transfer maximum possible heat from the source to the working fluid, with minimum wastage. In many contexts however, additional concerns like size, weight, pumping losses, and cost also become extremely relevant. These must often be balanced with effective heat transfer, creating a major challenge for engineers.

1.1 Metal Additive Manufacturing (AM) and its Effect on Heat Exchanger Design

The development of metal AM techniques has pushed the envelope in heat exchanger design. One relevant family of techniques has been powder bed fusion, illustrated in fig. 1.1. It involves laying powdered material in layers and fusing or sintering parts of each layer via selective deposition of energy. This way, the part is built up in layers. Typically, support structures are not required as the unfused powder provides the required support.

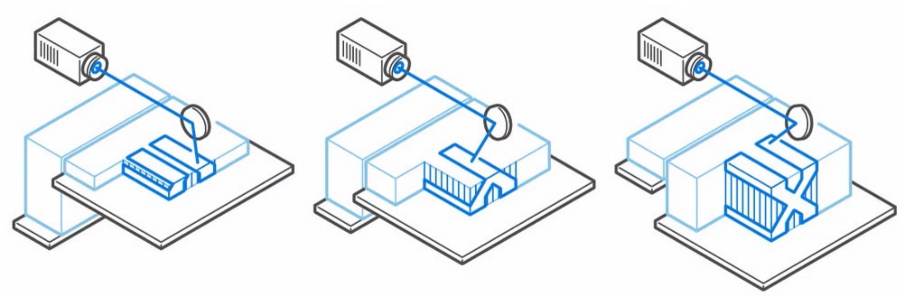


Figure 1.1: Schematic representation of a powder-bed fusion process. (credit: All3DP[1])

This form of manufacturing provides significantly greater freedom in part design than conventional subtractive techniques, as it is unencumbered by considerations on how tools will access the surface of the workpiece. This allows for a near-direct translation of designs from the optimization package to manufacturing. The entire volume available can therefore now be optimized to meet an objective function with little to no consideration towards manufacturability. Many bio-inspired design features, too complex for conventional techniques, can also be included. Additional elements of the heat exchanger, such as manifolds or mounting hardware, can be integrated with the core into a single piece. This allows for several important advantages in the final heat-exchanger design:

- Performance achieved is similar to conventional designs while requiring significantly lower pressure drop, therefore reducing power requirements for the setup.

- Much higher surface area can be packed into much smaller volumes, thus reducing size of the setup significantly. This also reduces the weight of the setup, which is especially important in aerospace applications. Part consolidation also plays an important role in this aspect.

- The reduction or elimination of joints and seals, necessitated by conventional manufacturing, also reduces the possibility of leaks, an important maintenance concern.[2]

- Custom designs for the requirements of each consumer become possible, as manufacturing cost is unrelated to scale.

However, for all its benefits, powder-bed fusion remains expensive. Fixed costs are still over half a million dollars for direct metal laser sintering, and over a million dollars for electron-beam melting.[3] Here, one of the biggest strengths of AM parts also provides an opportunity for cost reduction. Laureijs et al. determined that reducing the size of parts can save costs not only through the reduction of powder used per print, but also through the printing of several parts in one build.[3] This provides a strong motivation to make not only the heat exchanger core, but its ancillary components as well extremely compact.

Another relevant quirk of AM is the surface quality, or lack thereof. Most forms of additive manufacturing produce parts with rough surfaces. Given the powdered nature of the feedstock and its quick melting and cooling, this is also true of metal AM. Cabanettes et al.[4] determined that this roughness can be in the order of 20 μm for selective laser melting and electron beam melting. They also find that both the nature and magnitude of roughness can change significantly with orientation and post treatment using heat. As Nieuwstadt et al.[5] point out, wall roughness becomes relevant for turbulent flows when it becomes comparable in scale to the viscous sub-layer. This means that for high Reynolds numbers (Re), where the viscous sub-layer is very thin, wall roughness can affect heat exchanger's performance. While roughness is bound to reduce with improvements in AM technology itself and post-treatment, it is likely to remain an important design consideration for devices with high Re flows.

1.2 Schwartz-Cells in Heat Exchanger Cores

The design freedom afforded by AM has so far been extensively exploited by designing random structures for heat exchange. These, such as foams and stochastic networks, are exceptionally light and well-performing. However, the lack of repeatability poses a challenge for engineers. An alternative that preserves much of the advantages of random structures while adding predictability of properties are lattice structures. These are formed by using surfaces to connect a system of vertices, creating complex internal volumes that can exchange heat or mass through the generated surface.

A type of lattice structure of particular interest is the triply-periodic minimal surface (TPMS). The best known example of a minimal surface is the soap film formed between two circular rings. It is observed that this does not hold the expected cylindrical shape but a catenoid shape, as seen in fig. 1.2. This shape, characterised by a circle of decreasing circumference from both directions, achieves the minimum possible surface. This makes it the most stable surface for the prescribed geometry. When this concept is applied in three directions, and the resulting surface repeated over a volume, we get a TPMS. The smooth surface of a TPMS neatly divides the volume into two equal, intertwined yet separate parts. This produces a light, strong structure excellent for application in the core of a heat exchanger. A TPMS core can produce a Nusselt number (Nu) several-fold higher than conventional designs, while being far more compact.[6] While this comes at the cost of a higher pressure drop, the improvements in performance are considered worth the penalty.[6]

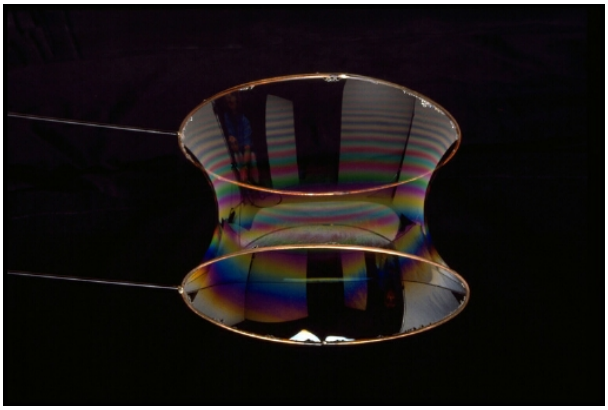


Figure 1.2: Soap film between two rings forming a catenoid. (credit: Exploratorium[7])

Despite research into TPMS being new, several types have been studied so far: Schwartz primitive, Schwartz diamond, Gyroid, Neovius etc. The review by Dutkowski et al. suggests that of these the Schwartz diamond geometry shows the most promise.[6] While the pressure drop through it is higher than the rest, the increase in Nu outstrips this penalty.[6] The Schwartz diamond, illustrated in fig. 1.3, is described by eqn. 1.1. Here parameter C describes the wall thickness.

$$f(x,y,z) = \cos(x)\ \cos(y)\ \cos(z) - \sin(x)\ \sin(y)\ \sin(z) = C \tag{1.1}$$

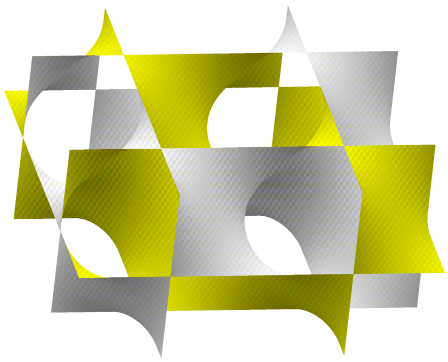


Figure 1.3: A Schwartz diamond surface. (credit: Wikimedia commons[8])

The geometry considered in this work is a Schwartz diamond core already optimized by a team at Université de Toulouse, in collaboration with Total energies. Its dimensions are 100 mm × 200 mm × 100 mm, and the distribution of inlets and outlets is described in fig. 1.4.

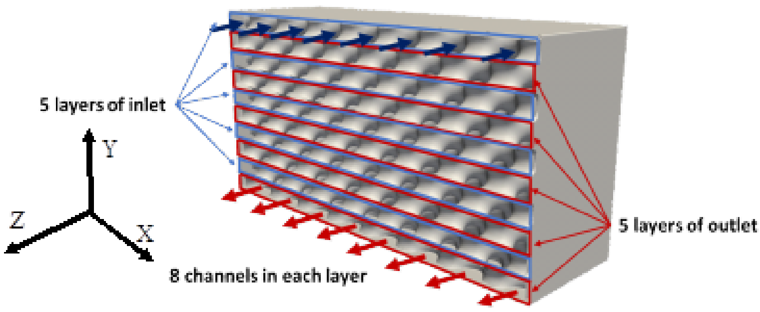


Figure 1.4: Schwartz diamond core considered in the work.

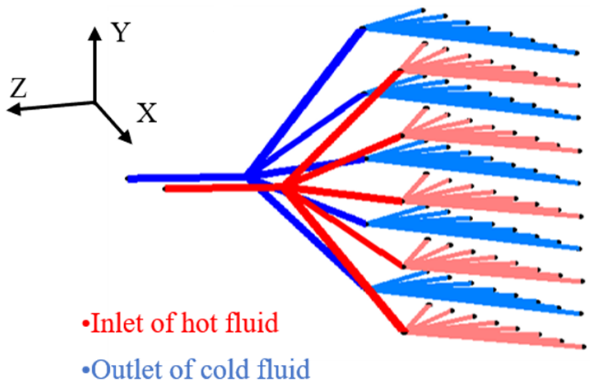


Figure 1.5: Schema of inlet and outlet manifolds for the core in fig. 1.4.

In fig. 1.5, we see the two-stage inlet manifold that is envisaged for the core. The first stage distributes the flow from a single inlet to the five horizontal levels. The second stage distributes the flow for its respective level into eight outlets which enter the core. In continuation of previous work, only the second stage is discussed and optimized here.

1.3 Heat Exchanger Manifolds

While previous sections have largely focused on the design of the heat exchanger core and its implications, it has long been known that proper flow distribution to the core is also critical. While maldistribution can take several forms, in this work it is defined by differing flow rates through each outlet of the manifold. This is quantified using a maldistribution factor ($MF_{\%}$), which is defined below:

$$\mathrm{MF}_{\%} = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\frac{m_i - \bar{m}}{\bar{m}}\right)^2} \times 100 \qquad (1.2)$$

This definition is also used in other work in literature, for example by Minqiang et al.[9] Dąbrowski[10] recommends this definition as it can consistently characterise the evolution of multiple properties across reynold's numbers, indicating that it is robust.

Work on plate-fin heat exchangers[11] and various other geometries[12] shows that maldistribution significantly reduces the thermal performance of a heat exchanger by creating hot spots. Additionally, work on plate heat exchangers[13] suggests that it can also increase the pressure drop required. This is doubly relevant for this work given the higher pressure drop of the Schwartz diamond geometry, as described in the previous section. It is apparent therefore that a manifold which causes significant maldistribution will thus undo or at least significantly diminish the advantages gained through optimization and AM.

Alongside performance concerns, an important metric for manifold design is the degree of compactness. Space-efficiency is not only desirable in its own right, it also reduces overall heat exchanger volume and thus manufacturing cost significantly. Even when paired with the optimised core considered in this work, the unoptimized manifold consists of nearly 50% of the heat exchanger's total volume. This can thus make or break the economic case for the introduction of an optimized heat exchanger to the market. It is therefore imperative that the manifold is optimised just as the core has been for AM.

1.4 Numerical Optimization

The manifold can thus be designed via numerical optimization, which can refer to either parametric, shape or topology optimization. Parametric or size optimization (see fig. 1.6a) involves parametrizing the geometry by specifying widths, lengths, radii etc at various points, and then finding the ideal combination of these parameters. In shape optimization (see fig. 1.6b), the outer boundary of the geometry is optimized instead, which allows one to achieve far more complex geometries. In topology optimization (TO), (see fig. 1.6c) the entire domain is defined as porous. The optimizer then adjusts this porosity at each point to very low values (thus solid) or very high values (thus empty). For flow-optimization, the empty region becomes the optimum path for the fluid.

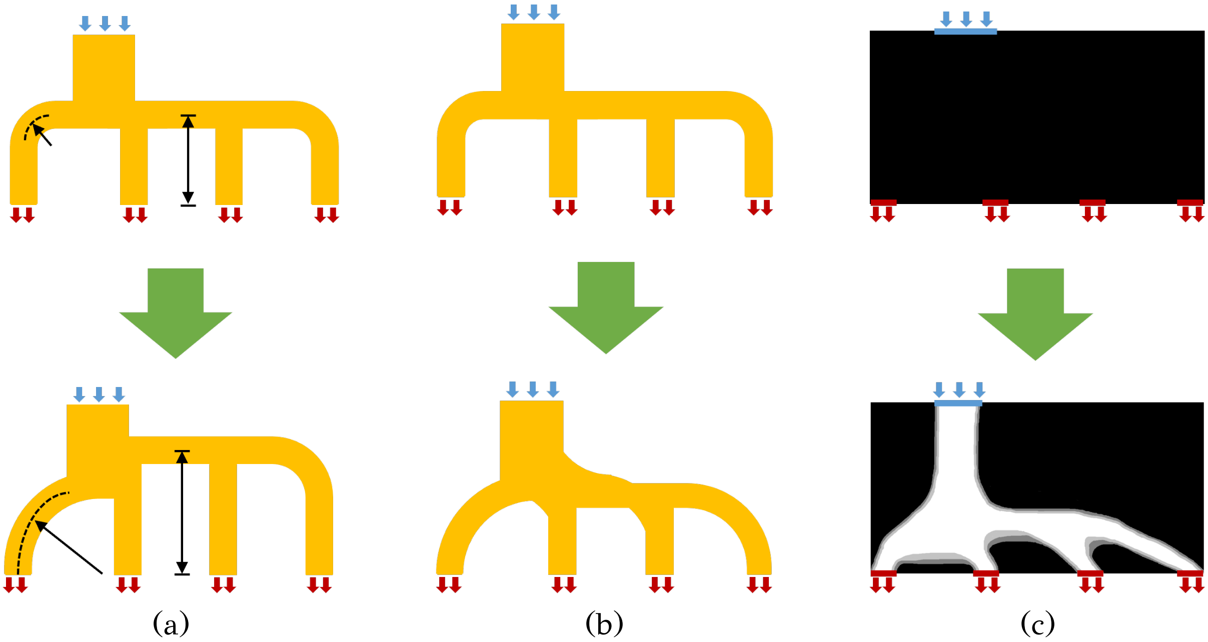


Figure 1.6: (a) Parametric (b) Shape and (c) Topology optimization illustrated.

1.4.1 Topology Optimization

It can thus be inferred that TO has the best potential to optimize - not only does it explore the full domain, it is less dependent on the engineer's initial guess. This is why it is the method of choice for this work. Density-based TO, the specific flavour used, can be summarised as follows:

1. An initial guess for the domain is fed into the optimizer. This guess can be a fully empty domain for flow optimization, or an engineer's guess.
2. The domain is divided into several discrete elements on which the porosity will be adjusted and the objective function will be evaluated.
3. This geometry is then evaluated using an appropriate simulation code or package to check the value of the objective function.
4. The sensitivities are calculated at each point in the domain. This tells us how changing the porosity of each element affects the objective function.
5. The optimizer updates the geometry according to these sensitivities, by adding/removing material from certain elements so as to make the most progress toward the objective.
6. We then return to step 3. The loop continues until convergence is reached.

1.4.2 The History of Flow Topology Optimization

Given its origins in structural mechanics, TO was initially formulated on finite-element solvers for flow problems as well. Early work, such as the pioneering study of Borrvall and Petersson,[14] was concerned with stokes flows. The use of incompressible stokes flow equations however meant their methodology was quite limited in applicability. Gersborg-Hansen et al.[15] extended this work by incorporating the effects of inertia through use of the navier-stokes equations, and later by formulating it over a more conventional finite-volume solver.[16] This meant a larger range of Re could be considered, and optimization research would be more applicable to industry. However, until this point work was still relevant only to laminar flows, where viscous forces dominate over inertial ones.

Othmer[17] was the first to formulate a TO problem for turbulent flows. This however, came with a caveat - their formulation neglected the variation in turbulent viscosity due to changes in the design variables, thus making the 'frozen turbulence' assumption. This reduces the accuracy of the calculated sensitivities of the objective function to the various design variables. Kontoleontos et al.[18] corrected this by using the Spallart-Allmaras turbulence model and solving adjoint equations for it. This allowed them to correct the discrepancies in the frozen turbulence assumption. Dilgen et al.[19] further improved upon this by devising a method to compute the exact sensitivities via automatic differentiation. Their proposed technique works with a variety of closure models like k-ω and Spallart-Allmaras while being computationally cheap.

The density-based methods described above cannot use a discrete 0,1 formulation for density. Attempting to do so typically results in 'checker-boarding,' where the optimizer produces extremely fine features with similarly small spaces between them. This is undesirable from both a manufacturing and end-use standpoint.

A continuous formulation is thus used, allowing for densities between 0 (fully solid) and 1 (fully liquid). This results in 'grayscale' areas (neither fully solid nor empty). These must be manually removed or filled in later, since it is presently not clear how to interpret these areas during manufacturing. Penalization methods do exist to minimize the extent of these regions, like the widely-used Solid Isotropic Material with Penalization (SIMP) scheme proposed by Bendsoe and Kikuchi.[20] Such schemes penalize the performance of intermediate densities. In the case of flow, this would mean that grayscale regions resemble solid regions more closely, and are thus not preferred.

The presence of such regions, however small, creates a fundamental problem in turbulent flow simulation. Turbulent flows are by nature high Re flows, which means that there is a large difference of magnitude between relevant length scales. In other words, the simulation code must resolve flow at length scales much smaller than the size of the geometry. Therefore, if the shape of the geometry itself is unclear at these scales, it will not be possible to simulate it accurately. The optimizer would thus be working with a limited understanding of the actual performance of the geometry. An exact geometry is needed at each iteration of the optimization process.

Fortunately, a very stable method has recently been developed with a discrete formulation for density, known as 'topology optimization of binary structures' (TOBS). In what is a common thread in the optimization field, this method was first proposed in the structural domain by Sivapuram and Picelli.[21] Souza et al.[22] then applied this to flow topology optimization. Picelli et al.[23] then added a 'geometry-trimming' (GT) procedure to the method, which removes solid portions from the design and meshes the trimmed geometry separately. This stops flow from entering solid regions, reduces domain size for faster computation, and allows for advanced meshing techniques such as mesh inflation to produce accurate results efficiently. This updated method, known as TOBS-GT, is used for this work, and will be explained in detail in the following section.

1.5 Optimized Manifold for Laminar Flow

The team at Université de Toulouse have also done some work on producing an optimized manifold for the core discussed in section 1.2. They used a border optimization technique developed by Dapogny et al.[24] in FreeFEM++. The objective function was chosen to minimise the pressure drop and $MF_{\%}$.

Unfortunately, FreeFEM++ can only work with laminar flows, and thus an Re of 168 was chosen. The initial design, defined in a 2D domain, is illustrated in fig. 1.7. The result obtained (see fig. 1.8), achieves an $MF_{\%}$ of 26.05%, improving on their parametrically-optimized reference model, which achieves 46%.

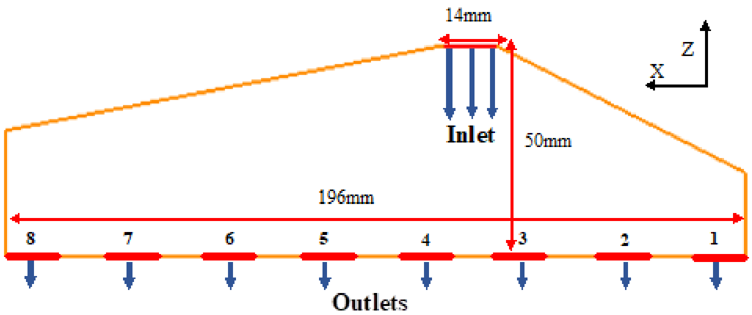


Figure 1.7: Initial geometry for the optimizer, with inlets and outlets defined.

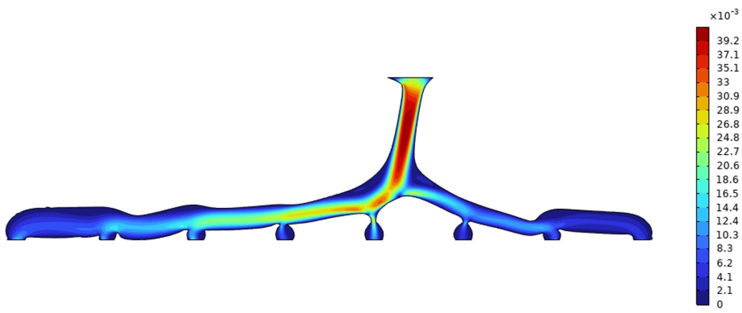


Figure 1.8: 2D velocity contours (in m/s) in the shape-optimized geometry for laminar flow.

However, this geometry loses most of its benefits when run with turbulent flow. With an Re of 8103, $MF_{\%}$ rises up to 39.34%. They do demonstrate the possibility of manually adding walls inside the manifold to redirect the flow and thus reduce significantly the turbulent $MF_{\%}$. This process is tedious, so a method to obtain a true optimised geometry for turbulent flow remains desirable.

1.6 Motivation

Great advances have been made in the field of optimization in the past 20 years. However, as Alexandersen et al.[25] note in their extensive review of literature, experimental validation of these optimized geometries is quite lacking. This is concerning as most optimization workflows make two extremely important assumptions:

- Optimization is typically carried out in 2D, as 3D optimization is too computationally expensive. This means the effect of the top and bottom walls is neglected.
- Most optimizers assume theoretical, smooth walls. Yet most of this complicated geometry will be manufactured via AM, which produces extremely rough surfaces compared to traditional manufacturing. As discussed in section 1.1, this can change the flow characteristics significantly.

1.7 Research Questions

The above discussion of literature raises the following research questions:

- What is the correct balance between the compactness, pressure drop and $MF_{\%}$ of a manifold for an optimizer?
- What are the differences between optimization with a 2D and 3D domain?
- Can observed differences be incorporated into the optimizer/flow model while preserving 2D optimization?
- How does a manifold optimized for turbulent flow differ from one optimized for laminar flow?
- What is the effect of rough surfaces manufactured by AM on manifold operation in laminar and turbulent regimes?
- How can any observed roughness effects be incorporated into the optimizer?

Chapter 2

Methodology

2.1 Optimization

The TOBS-GT method largely resembles the one in section 1.4.1, with the addition of a geometry-trimming step. The adjusted methodology therefore becomes:

1. An initial domain is fed into the optimizer, which is written in MATLAB.
2. Solid parts of the domain, if any, are removed via a GT script.
3. This trimmed geometry is then transferred to COMSOL where it meshed.
4. The objective function and sensitivities are calculated at each point in the COMSOL mesh.
5. This information is relayed back to MATLAB and interpolated over a structured mesh used in the optimizer.
6. The optimizer updates the geometry using Integer linear programming (ILP).
7. We then return to step 2. The loop continues until convergence is reached.

Let us explore the optimization methodology in detail.

2.1.1 Problem definition

A TOBS-GT problem, like most others, is defined as the minimization of an objective $f(\mathbf{x})$, subject to constraints $g_i(\mathbf{x}) \leq \overline{g_i}$, $i \in [1, N_g]$. The objective and all constraints are functions of the design variables $\mathbf{x}$, a vector that decides whether each element is solid (0) or fluid (1). Mathematically, we write this as:

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) \\ \text{Subject to} \quad & g_i(\mathbf{x}) \leq \overline{g_i} \quad i \in [1, N_g] \\ & x_j \in \{0, 1\} \quad j \in [1, N_d] \end{aligned} \tag{2.1}$$

Where N_d is the number of elements in the mesh used for optimization.

2.1.2 Problem Linearisation

The problem must then linearised to be solvable using ILP. This is achieved using Taylor's approximation. The objective function and constraints can be approximated as:

$$\begin{aligned} f(\mathbf{x}) &= f\left(\mathbf{x}^k\right) + \frac{\partial f\left(\mathbf{x}^k\right)}{\partial \mathbf{x}} \cdot \Delta\mathbf{x}^k + O\left(\left\|\Delta\mathbf{x}^k\right\|_2^2\right) \\ g_i(\mathbf{x}) &= g_i\left(\mathbf{x}^k\right) + \frac{\partial g_i\left(\mathbf{x}^k\right)}{\partial \mathbf{x}} \cdot \Delta\mathbf{x}^k + O\left(\left\|\Delta\mathbf{x}^k\right\|_2^2\right) \end{aligned} \tag{2.2}$$

With $(\cdot)^k$ denoting a quantity at the k^{th} iteration, and $O\left(\left\|\Delta\mathbf{x}^k\right\|_2^2\right)$ being the second-order error which is discarded. The linearised problem can then be written as:

$$\begin{array}{lll} \text{Minimize} & \left.\dfrac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}^k} \Delta\mathbf{x}^k & \\ \text{Subject to} & \left.\dfrac{\partial g_i}{\partial \mathbf{x}}\right|_{\mathbf{x}^k} \Delta\mathbf{x}^k \leq \overline{g}_i - g_i^k & i \in [1, N_g] \\ & \Delta x_j \in \{-x_j, 1 - x_j\} & j \in [1, N_d] \end{array} \tag{2.3}$$

Where the second constraint ensures that the design variable at every element remains either 1 or 0. The design variables can then be updated as:

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \Delta\mathbf{x}^k \tag{2.4}$$

The linearisation approximation in eq. 2.3 is valid only for very small changes in the objective and constraints. Therefore, the number of elements changing from solid to empty and vice-versa in a single step must be limited as:

$$\left\|\Delta x^k\right\|_1 \leq \beta N_d \tag{2.5}$$

Where the number of changes is limited to a small fraction β of the total number of elements. Similarly, whenever the difference between the current value of a constraint and its target is large, it must be artificially reduced. This is achieved by multiplying it by a small number ε_i. This has the effect of 'relaxing' the constraint, and ensuring the ILP has a feasible solution. Mathematically, we write this as:

$$\Delta g_i = \begin{cases} -\varepsilon_i g_i^k & : \overline{g_i} < (1 - \varepsilon_i)\, g_i^k \\ \overline{g_i} - g_i^k & : \overline{g_i} \in \left[(1 - \varepsilon_i)\, g_i^k, (1 + \varepsilon_i)\, g_i^k\right] \\ \varepsilon_i g_i^k & : \overline{g_i} > (1 + \varepsilon_i)\, g^k \end{cases} \tag{2.6}$$

2.1.3 Integer Linear Programming

ILP is then used to solve the problem. An ILP problem is simply a linear programming problem with the additional constraint that the solution must be an integer. Picelli et al.[23] use a branch and bound solver, as devised by Land and Doig,[26] to solve it in TOBS-GT. An initial solution is first obtained by solving a linear programming problem, without demanding an integer. This is then used to create multiple branches of linear programs, which are solved with additional bounds to yield an integer solution. ILP is implemented via the CPLEX optimization suite from IBM, which provides better performance than the `intlinprog` solver available in MATLAB.

2.1.4 Sensitivity Analysis

As discussed in previous sections, TOBS is a gradient-based method. Therefore, the optimizer must know the sensitivity of the objective function and any constraints to changes in the density of each element. This is calculated using the adjoint method by Haftka and Gürdal,[27] which is explained in this section.

The governing equations are satisfied by the state solution $\mathbf{U}(\mathbf{x})$, and the residual $\mathbf{R}(\mathbf{x}, \mathbf{U}(\mathbf{x}))$ becomes zero. Based on this information, the objective function $f(\mathbf{x}, \mathbf{U}(\mathbf{x}))$ can be augmented with a Lagrange multiplier vector $\mathbf{\Lambda}^T$ to yield a Lagrangian function $\mathscr{L}$.

$$\mathscr{L} = f + \mathbf{\Lambda}^T \mathbf{R} \tag{2.7}$$

If the residual is zero, the Lagrangian becomes equal to the objective. Eq. 2.7 can be differentiated with respect to the design variable $\mathbf{x}$ as:

$$\begin{aligned}\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathbf{x}} &= \frac{\partial f}{\partial \mathbf{x}} + \frac{\partial f}{\partial \mathbf{U}}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}\mathbf{x}} + \mathbf{\Lambda}^T\left(\frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \frac{\partial \mathbf{R}}{\partial \mathbf{U}}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}\mathbf{x}}\right) \\ &= \frac{\partial f}{\partial \mathbf{x}} + \mathbf{\Lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{x}} + \underbrace{\left(\frac{\partial f}{\partial \mathbf{U}} + \mathbf{\Lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)}_{=0}\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}\mathbf{x}}\end{aligned} \tag{2.8}$$

Since $(\mathrm{d}\mathbf{U}/\mathrm{d}\mathbf{x})$ is computationally prohibitive to calculate, the above indicated part is set to zero. This gives us the adjoint equation, which can be written as:

$$\left(\frac{\partial \mathbf{R}}{\partial \mathbf{U}}\right)^T \mathbf{\Lambda} = \left(\frac{\partial f}{\partial \mathbf{U}}\right)^T \tag{2.9}$$

Sensitivities are then calculated as:

$$\left(\frac{\mathrm{d}\mathscr{L}}{\mathrm{d}\mathbf{x}}\right) = \left(\frac{\partial f}{\partial \mathbf{x}}\right)^T - \mathbf{\Lambda}^T\frac{\partial \mathbf{R}}{\partial \mathbf{x}} \tag{2.10}$$

This computation is performed in COMSOL Multiphysics, and efficiently finds the exact sensitivities.

2.1.5 Filtering

In order to avoid the checker-boarding problem, Picelli et al.[28] use a filtering scheme for the densities and sensitivities. The following steps are used for the sensitivity filtering:

1. Sensitivity data is first transferred from the mesh generated in COMSOL using the smoothed geometry to the elements of the TOBS mesh.
2. Sensitivity is filtered within a specified minimum radius r_{min}.

The filtered sensitivity is calculated as:

$$\widetilde{\frac{\partial f}{\partial x_j}} = \frac{1}{\sum_{m \in N_m} H_{j,m}} \sum_{m \in N_m} H_{j,m} \frac{\partial f}{\partial x_m} \tag{2.11}$$

Where j the element for which sensitivity is being filtered, N_m is the set of elements m whose centres are within the filter radius, and $H_{j,m}$ is a weight factor written as:

$$H_{j,m} = \mathtt{max}(0, r_{\text{min}} - \mathtt{dist}(x_j, x_m)) \tag{2.12}$$

Closer elements therefore contribute more to the filtered sensitivity. This filtered sensitivity is also averaged across iterations to ease the convergence of optimization:

$$\widetilde{\frac{\partial f}{\partial x_j}}^{k} = \frac{\widetilde{\frac{\partial f}{\partial x_j}}^{k} + \widetilde{\frac{\partial f}{\partial x_j}}^{k-1}}{2} \tag{2.13}$$

The filtered and averaged sensitivity for the objective is then used in place of the true sensitivity in the linearised problem in eq. 2.3.

Density filtering is also used in this work to smooth the contours of the geometry output of the optimizer. Similarly to the sensitivities, the density is filtered using a filter radius. A level set function then extracts the contour of 0.5 density as the wall of the manifold. This is then transferred to COMSOL.

2.1.6 Covergence Criteria

The convergence model from Huang and Xie[29] is used, as given below:

$$\frac{\left|\sum_{i=1}^{N} (C_{k-i+1} - C_{k-N-i+1})\right|}{\sum_{i=1}^{N} C_{k-i+1}} \leq \iota \tag{2.14}$$

Where ι is the allowable convergence tolerance. Stable convergence must be achieved for atleast $2N$ successive iterations for optimization to stop.

2.2 Governing Equations and Numerical Analysis

2.2.1 Continuity Equation

In this work, only incompressible flows are considered, so the fluid density does not vary across the domain. Therefore, the continuity equation can be written as:

$$\rho \nabla \cdot \mathbf{u} = 0 \tag{2.15}$$

2.2.2 Momentum Equation

The COMSOL laminar flow interface uses the navier-stokes momentum conservation equations, which are written as:

$$\rho(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla \cdot [-p\mathbf{I} + \mathbf{K}] + \mathbf{F} \tag{2.16}$$

With p representing pressure, $\mathbf{K}$ the deviatoric stress tensor, and $\mathbf{F}$ the body force. The deviatoric stress tensor can further be expressed as a function of the velocities and dynamic viscosity μ as:

$$\mathbf{K} = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\top}\right) \tag{2.17}$$

2.2.3 Turbulence Equations

Two models are tested for optimization, the k-ϵ model and the k-ω model, with the latter being recommended for internal flows. Both models introduce additional transport equations to calculate additional variables that allow for closure of the navier-stokes equation.

The k-ϵ Model

COMSOL implements the standard k-ϵ model.[30] As the name suggests, two additional variables are calculated: the turbulent kinetic energy k, and the turbulent dissipation rate ϵ. These are calculated via the following equations:

$$\begin{aligned} \rho(\mathbf{u} \cdot \nabla)k &= \nabla \cdot \left[\left(\mu + \frac{\mu_{\mathrm{T}}}{\sigma_k}\right)\nabla k\right] + P_k - \rho\epsilon \\ \rho(\mathbf{u} \cdot \nabla)\epsilon &= \nabla \cdot \left[\left(\mu + \frac{\mu_{\mathrm{T}}}{\sigma_\epsilon}\right)\nabla \epsilon\right] + C_{\epsilon 1}\frac{\epsilon}{k}P_k - C_{\epsilon 2}\rho\frac{\epsilon^2}{k} \end{aligned} \tag{2.18}$$

Where μ_{T} is turbulent viscosity, P_k is the production term, and $C_{\epsilon 1}$, $C_{\epsilon 2}$, σ_k and σ_ϵ are experimentally derived model constants. The model is not adept at describing flow near walls. Therefore, analytical expressions known as wall functions are used for this purpose.

The k-ω Model

COMSOL uses the Wilcox revised k-ω model,[30] which calculates the turbulent kinetic energy k, and the specific dissipation rate ω. These are calculated via the following equations:

$$\begin{aligned}\rho(\mathbf{u}\cdot\nabla)k &= \nabla\cdot\left[\left(\mu+\frac{\mu_T}{2}\right)\nabla k\right]+P_k-\beta^*\rho\omega k\\ \rho(\mathbf{u}\cdot\nabla)\omega &= \nabla\cdot\left[\left(\mu+\frac{\mu_T}{2}\right)\nabla\omega\right]+\frac{13\omega}{25k}P_k-\rho\beta\omega^2\end{aligned} \tag{2.19}$$

Where β and β^* are model constants. COMSOL automatic wall treatment is used, which employs wall functions only when the mesh near the boundary is not ideal to compute flow directly.

To aid convergence in cases where it is elusive, the strategy of inconsistent stabilization is used. To stabilize the navier-stokes computation, artificial diffusion is defined as follows:

$$c_{\text{art}} = \delta h||\mathbf{u}|| \tag{2.20}$$

Where h is the element size, $||\mathbf{u}||$ is the advection in the mesh, and δ is a tuning parameter, typically set as 0.25. This is added to the physical diffusion, i.e. the viscosity as $(\mu/\rho)+c_{\text{art}}$. This added diffusion is isotropic, and 'inconsistent' as the added diffusion remains regardless of solution accuracy. Artificial diffusion is similarly added to the turbulence equations.

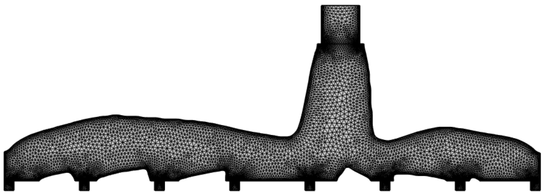
2.2.4 Solution Methods

For 2D CFD computation, the PARADISO method was used. This is a fast and quite robust method based on lower-upper decomposition, and is also the default recommendation in COMSOL documentation. For 3D CFD computation, the MUMPS method was used. While it is largely similar in nature to PARADISO, it also has support for cluster computing, which helps in solving larger problems on more powerful machines. This more than makes up for the fact that it is slightly slower than PARADISO.

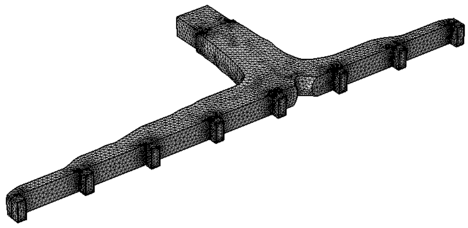
2.2.5 Mesh Independence

COMSOL automatic settings are largely used across the work to handle meshing, both during the optimization and numerical re-evaluation of generated geometry. As recommended in documentation, P1+P1 discretisation is used, which implies linear elements for both velocity and pressure calculation. This is both computationally cheap and numerically stable as it prevents spurious oscillations.

COMSOL offers 9 automatic meshing settings ranging from 'extremely fine' (1) to 'extremely coarse' (9). After a mesh independence check on one of the optimization outputs (see fig. 2.1), the 'finer' (2) setting is used for 2D and 'coarse' (6) for 3D. The results of the independence check are plotted in fig. 2.2 and 2.3.



(a)



(b)

Figure 2.1: Selected mesh refinement level for 2D (a) and 3D (b) flows.

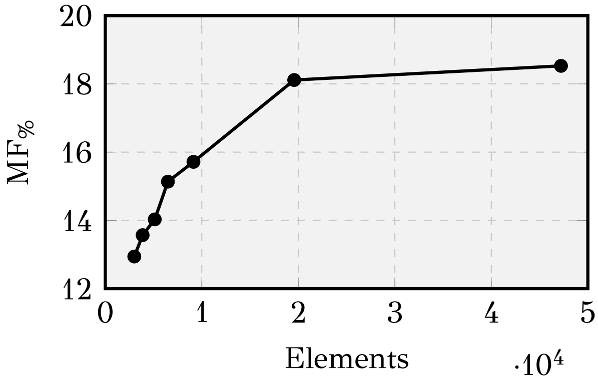


Figure 2.2: 2D mesh independence study (settings 2-8).

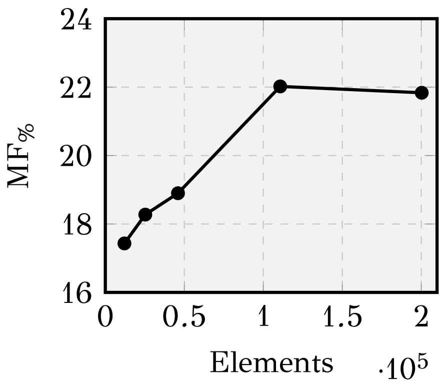


Figure 2.3: 3D mesh independence study (settings 5-9).

2.3 Experimental Validation

Experimental validation was then performed to verify the $MF_{\%}$ of the optimized geometry. For 2D optimization, the flow domain was converted to 3D by simply extruding the geometry by 8 mm. A manifold was then designed around this domain and manufactured via a resin printer with a layer height of 50 µm.

A traditional means of measuring $MF_{\%}$ would be to directly measure flow from each outlet via a flow meter. While this would be extremely accurate, it would necessitate the connection of eight separate flow meters, one for each outlet. This is because introducing back-pressure to any one of the outlets significantly changes the pattern of flow. Therefore, flow-meters would have to be installed on every outlet simultaneously. Venturi-meters and orifice plates similarly introduce uneven back-pressures that largely neutralize the differences in outflow between each outlet.

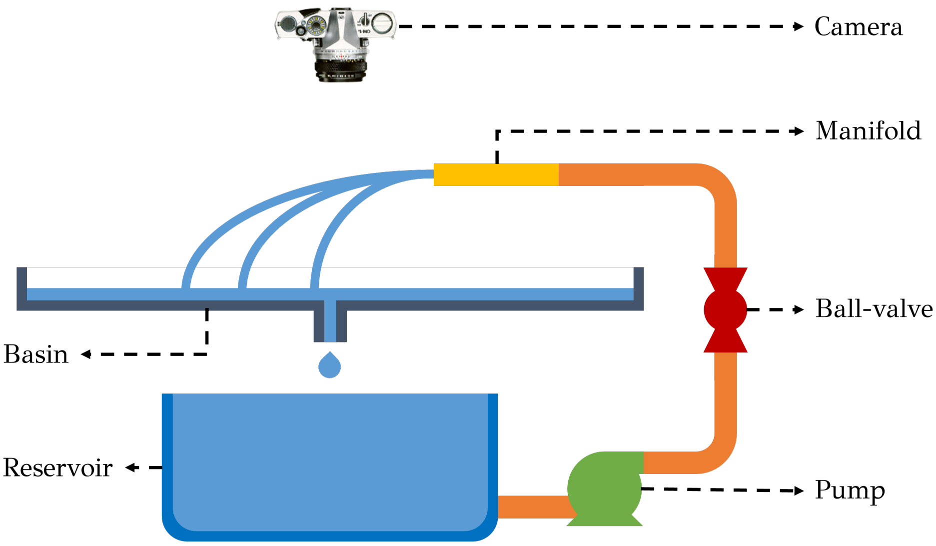


Figure 2.4: Experimental setup used for validation.

An unorthodox method was therefore used to measure $MF_{\%}$, which will be explained in this section. The manifold is placed level at a certain height from a flat basin. Water is pumped into the manifold and allowed to freely stream out of each outlet and land on the basin. The flow from each outlet is then recorded by means of a camera and some image processing. Water collected in the basin is also measured to determine total flow rate and flows back into a reservoir from which it is pumped again. For more clarity, the setup is illustrated in fig. 2.4.

2.3.1 Scaling Analysis

We know from basic kinematics that the water exiting the outlet follows roughly the shape of a parabola. The time it remains airborne is proportional to the height of the outlet as:

$$t_{\mathrm{f}} \propto \sqrt{h_{\mathrm{o}}} \tag{2.21}$$

The horizontal distance travelled by the water while airborne is therefore proportional as:

$$s = u \cdot t_{\mathrm{f}} \propto u \sqrt{h_{\mathrm{o}}} \tag{2.22}$$

Provided the manifold is level, the height is the same for each outlet. This distance thus becomes an accurate proxy for the velocity and therefore flow rate, and can be used to directly calculate $MF_{\%}$ without requiring velocity measurements.

2.3.2 Measurement Considerations

To minimize error in practice, the camera was held atleast 1.5m above the setup with a narrower focal length of 50 mm to avoid distortion caused by wide angle lenses. Any unavoidable distortion was corrected using the respective lens profile in Adobe Lightroom. A lower shutter speed of 1/30 s was also used to smooth any irregularities in the flow.

imageJ was the software of choice for processing these images. Lengths of each outlet jet were calculated in terms of the number of pixels taken up in the image, since only the relative size of each jet is relevant for $\mathrm{MF}_{\%}$ calculation.

2.3.3 Error Analysis

Error analysis was performed on this experimental technique using the guidelines from Moffat.[31] The measured outflow from each individual outlet was observed to have an uncertainty of ±15%. The effect of this value on the uncertainty in MF was calculated as:

$$\begin{aligned} \delta R_{X_i} &= \frac{\partial R}{\partial X_i}\delta X_i \\ \delta \mathrm{MF}_{M_i} &= \frac{-\delta M_i}{14\cdot \mathrm{MF}\cdot M_i} = 3.5\% \end{aligned} \tag{2.23}$$

We can then combine the contributions of all the outlets as:

$$\begin{aligned} \delta R &= \sqrt{\sum_{i=1}^{N}\left(\frac{\partial R}{\partial X_i}\delta X_i\right)^2} \\ \delta \mathrm{MF} &= \sqrt{\sum_{i=1}^{8}\delta M_i^2} = 10\% \end{aligned} \tag{2.24}$$

Therefore, an error margin of ±10% was prescribed.

2.3.4 Limitations

Laminar flow has proven challenging to validate. For the experiment to represent operation with a heat exchanger, the manifold must be full of fluid. For extremely low flow rates, this requires a certain amount of evenly-applied back-pressure. This would require the closing of the loop, which would make the chosen method difficult to apply. Therefore, only turbulent flows are validated here.

A recommendation to future users of this method would be to use more viscous alternatives to water such as oils or glycerol if validating laminar flow. This preserves the high flow rate while allowing for low Re. However, most such fluids are tedious to clean up, may require alternate pumps, and prevent the testing of turbulent flow with the same loop with water. This was therefore not attempted in this work.

Chapter 3

Results and Discussion

3.1 Problem Definition

To begin optimization, the first step is to clearly define the optimization domain, and the optimization problem in a form synonymous with eq. 2.3.

3.1.1 Flow Domain and Boundary Conditions

A 196 mm × 50 mm flow domain Ω is defined for 2D optimization (see fig. 3.1). Unlike the previous team, the domain is left rectangular to explore the optimization area as fully as possible. One inlet $\partial\Omega_{\text{in}}$ of 14 mm width is prescribed, with eight outlets $\partial\Omega_{\text{out},i}$ of width 3.5 mm each on the opposing wall. For 3D optimization, the domain is identical, save for a z-height of 8 mm.

At the inlet, a fully developed velocity profile is prescribed in accordance with the Re set for optimization. This is done via the 'fully developed flow' setting available in COMSOL. Instead of defining this flow profile via an equation, a virtual inlet duct is created. Upon this duct the correct amount of pressure is defined so as to produce the requested average inlet velocity. This average velocity is calculated from Re as:

$$u_{\text{av,in}} = \frac{\mu}{\rho d_{\text{in}} \text{Re}} \tag{3.1}$$

Where d_{in} is the width of the inlet. All outlets are set to atmospheric pressure, with walls assumed to be smooth with the no-slip condition.

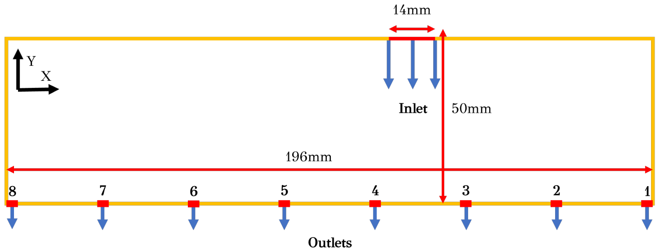
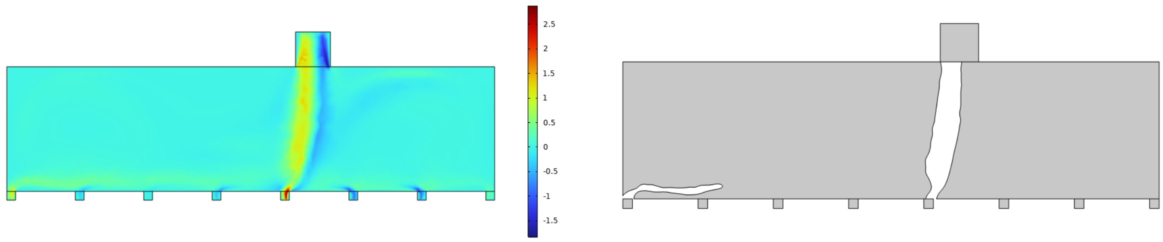


Figure 3.1: Domain used for optimization in this work.

3.1.2 Objective Function

Initially, the objective function was set as minimizing the MF. The optimizer behaves as expected, with the sensitivities indicating that it intends to constrict the outlets receiving the greatest flow, as seen in fig. 3.2a. However, this behaviour is stronger than expected, and always results in one or several outlets being blocked within a single iteration, as seen in fig. 3.2b. Once blocked, it is difficult to unblock an outlet. This is because once regions are trimmed out by the GT algorithm, they are not evaluated for sensitivity. They are assumed to have zero sensitivity and preserve their solid state.



(a) Sensitivity (b) Output Geometry

Figure 3.2: Initial sensitivity and subsequent output with MF as objective. Positive sensitivity indicates the element is preferred as solid, and vice versa.

Therefore, a weighted objective function is devised, with $MF_{\%}$ being weighted against ED, the energy dissipation of the flow. For a 3D domain, ED can be written as:

$$\mathrm{ED} = \mathbf{K}\nabla\mathbf{u} + \alpha(u^2 + v^2 + w^2) \tag{3.2}$$

Since the orders of magnitude of these two objectives can be well apart, they cannot be added directly with a weight. They must therefore be non-dimensionalised. For MF this is done by dividing it by the maximum possible MF achievable, ie. in the case when only one outlet is receiving all the flow. This can be calculated as:

$$\begin{aligned} \mathrm{MF}_{\max} &= \sqrt{\frac{1}{7}\left(7\left(\frac{0-(m_{\text{in}}/8)}{(m_{\text{in}}/8)}\right)^2 + \left(\frac{m_{\text{in}}-(m_{\text{in}}/8)}{(m_{\text{in}}/8)}\right)^2\right)} \\ &= \sqrt{\frac{7\cdot 1 + 7\cdot 7}{7}} \\ &= \sqrt{8} \end{aligned} \tag{3.3}$$

For ED this is done by dividing it by the value achieved in the initial domain/geometry, ie. ED_0. The weighed and non-dimensionalised objective function therefore becomes:

$$\begin{aligned} f(\mathbf{x}) &= w \cdot \overline{\mathrm{ED}} + (1-w)\overline{\mathrm{MF}} \\ &= w\frac{\mathrm{ED}}{\mathrm{ED}_0} + (1-w)\frac{\mathrm{MF}}{\sqrt{8}} \end{aligned} \tag{3.4}$$

With w being the weight. The value of w is chosen as the minimum with which it is possible to obtain an optimized geometry with all outlets receiving non-zero flow. Unfortunately, at present the only way to do this is tedious parameter tuning. A manifold was optimized for flows at several different Re, and the objective function was tuned by adjusting w. w was found to have a log-linear relation with Re, as illustrated in fig. 3.3. This means that the difficulty of optimizing for MF decreases more strongly with Re than that of optimizing for ED.

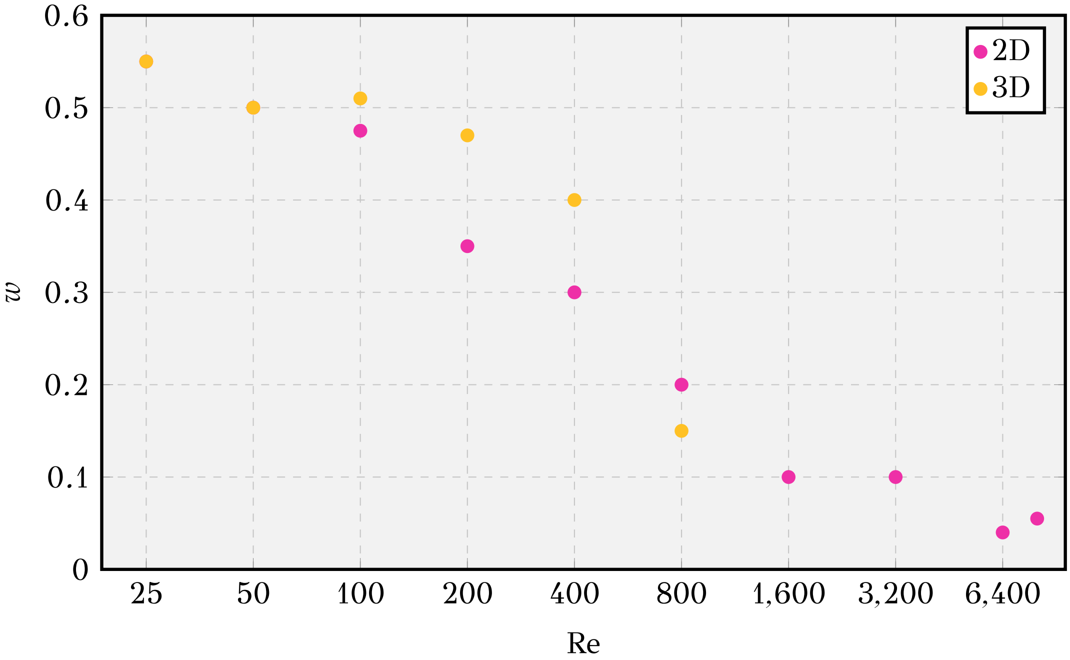


Figure 3.3: Variation of minimum viable weight with Re.

3.1.3 Constraints

With the discussion from the previous subsection, it seems logical to first implement a minimum mass-flow constraint on each outlet. However, this was unable to be implemented in this work for two reasons:

- For every outlet that flow is constrained for, an independent COMSOL evaluation must be performed for sensitivity to the constraint. Thus it proved prohibitive computationally, as the effort is multiplied nine times (one objective and eight constraints).
- The flow is constrained across the entire length from inlet to outlet. This often changes the optimized geometry significantly, thus having more than a merely constraining effect.

Therefore, the approach discussed in the previous section with a weighted objective function was retained. Only a minimum volume constraint $\overline{g}$ was applied to the problem, with this volume being simply defined as the average density of elements in the domain. It was observed that the optimizer prefers reaching $\overline{g}$ over minimizing the objective, as seen in fig. 3.4. Therefore, $\overline{g}$ also becomes a parameter for tuning that affects the optimized output.

As observed in fig. 3.4, there is an inflection point where the objective is compromised with further volume reduction. The strategy chosen for picking $\overline{g}$ is then to first set it extremely low, find this inflection point, then use it to obtain a converged optimization result. While optimizing for flows at several different Re and tuning $\overline{g}$ for each, it was also found to have a log-linear relation with Re, as illustrated in fig. 3.5.

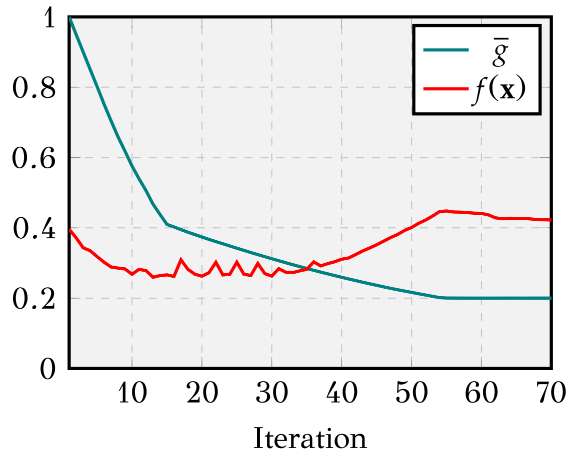


Figure 3.4: Optimization with $\overline{g}$ set too low.

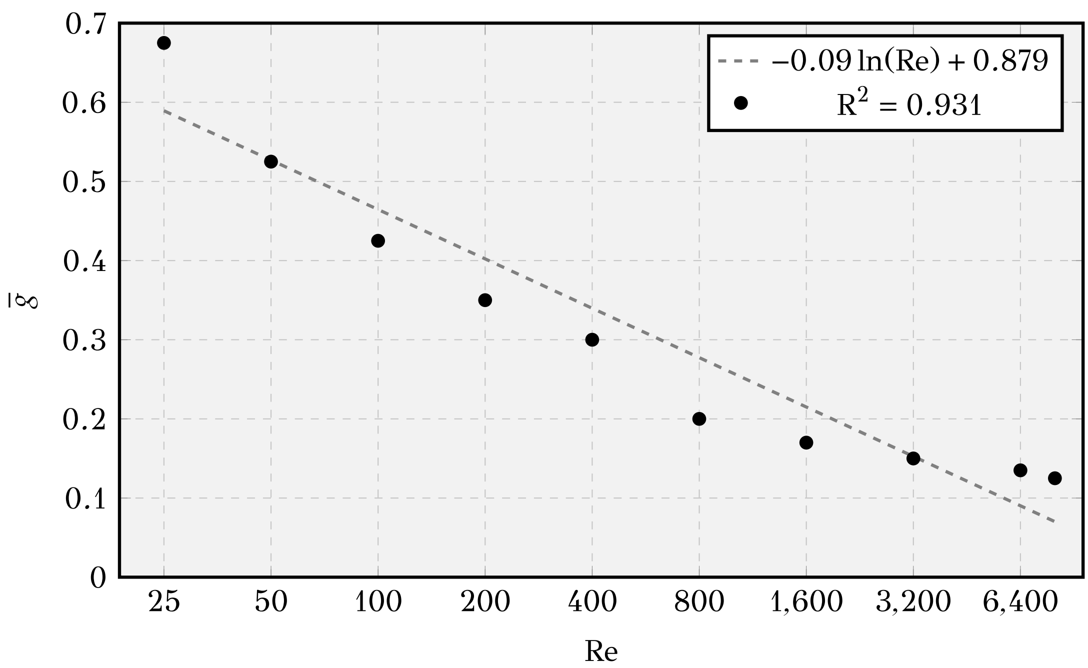


Figure 3.5: Variation of preferred volume constraint with Re

This implies that manifolds designed for higher Re values can be made more compact without compromising $f(\mathbf{x})$. A possible reason for this is that as inertial forces dominate over viscous ones with increasing Re, the flow 'self-organizes' into narrower channels. This means a larger proportion of the domain is a dead zone, allowing the manifold design to exclude most of it. This effect can be observed more visually in fig. 3.9.

3.1.4 Parameters

Parameter	**Value**
Sensitivity Filter Radius	0.020
Density Filter Radius	0.040
ε	0.020
β	0.050
ι	0.001
N	3

Table 3.1: Parameters used for the TOBS-GT optimizer in the 2D domain.

Parameter	**Value**
Sensitivity Filter Radius	0.050
Density Filter Radius	0.050
ε	0.020
β	0.050
ι	0.001
N	3

Table 3.2: Parameters used for the TOBS-GT optimizer in the 3D domain.

3.1.5 Defined Problem

The final problem definition can then be written as follows:

$$
\begin{aligned}
&\text{Minimize} && \frac{\partial}{\partial \mathbf{x}}\left(w\frac{\mathrm{ED}}{\mathrm{ED}_0} + (1-w)\frac{\mathrm{MF}}{\sqrt{8}}\right)\Bigg|_{\mathbf{x}^k} \Delta\mathbf{x}^k \\
&\text{Subject to} && \frac{1}{N_d}\,\frac{\partial\left(\sum_{i=1}^{N_d} x_i\right)}{\partial \mathbf{x}}\Bigg|_{\mathbf{x}^k} \Delta\mathbf{x}^k \le \overline{g} - \frac{\left(\sum_{i=1}^{N_d} x_i\right)^k}{N_d} \\
& && \Delta x_j \in \{-x_j, 1-x_j\} \qquad\qquad j \in [1, N_d]
\end{aligned}
\tag{3.5}
$$

3.2 Optimization Output

3.2.1 Comparing 2D and 3D Simulation

Initial CFD tests on the optimised geometry produced by the team at Université de Toulouse indicated that simulating in 2D or 3D significantly affected the flow pattern. While a 2D approximation of the geometry produced an $MF_{\%}$ of 37.41%, for a 3D simulation the result was only 26.05%. Comparing fig. 1.8 and 3.6, one can also see that the contours for both simulations are quite different. This is likely because the 2D simulation assumes that the domain is infinite in the precise axis where in reality the domain is thinnest. One might question then how this 2D assumption affects the outcome of optimization.

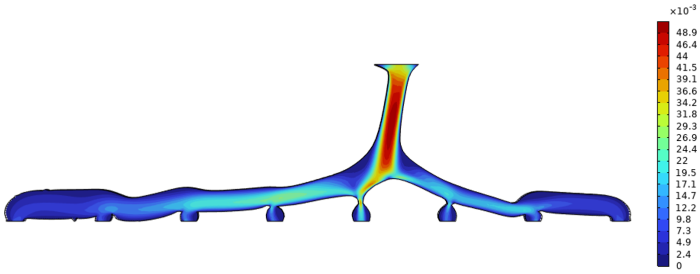


Figure 3.6: 3D velocity contours (in m/s) in the shape-optimized geometry for laminar flow.

In fig. 3.7 and 3.8 we can see that while the value of performance metrics differs between 2D and 3D simulation of the initial domain, their evolution with respect to Re is quite similar. The exception to this is at low Re values (25-100), where $MF_{\%}$ stays constant for the 3D case but not the 2D one.

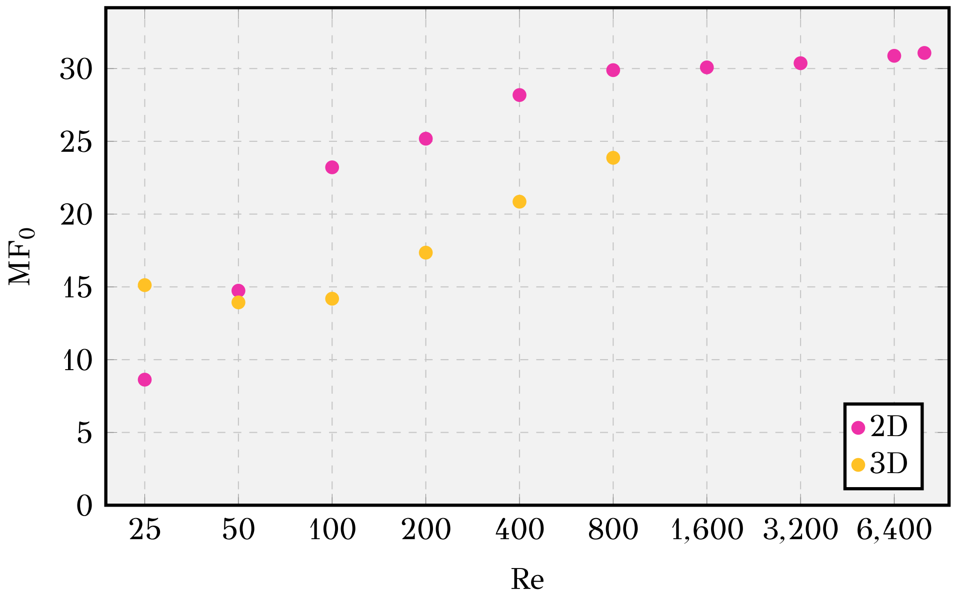


Figure 3.7: $MF_{\%}$ inside 2D and 3D initial domains for various Re values.

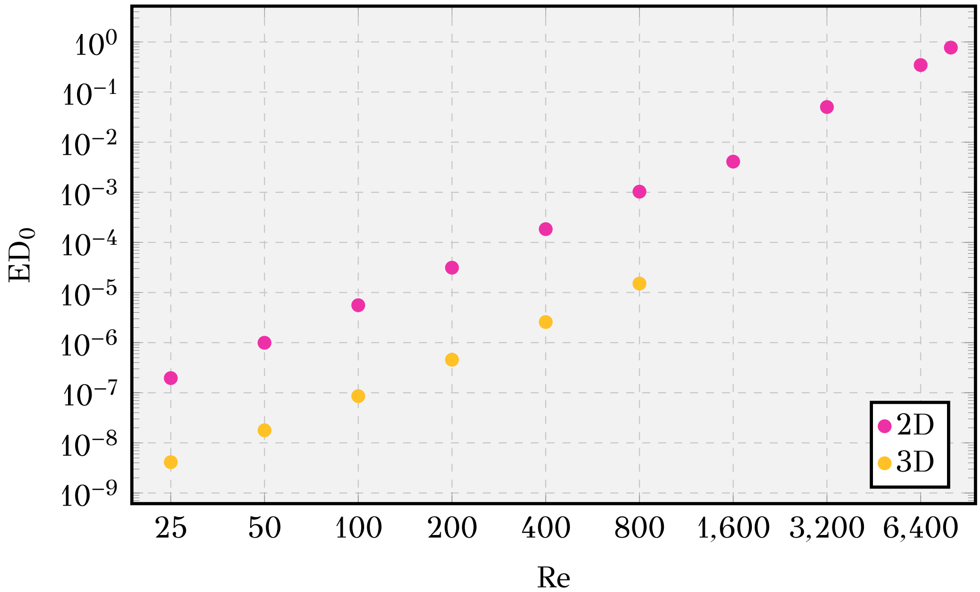
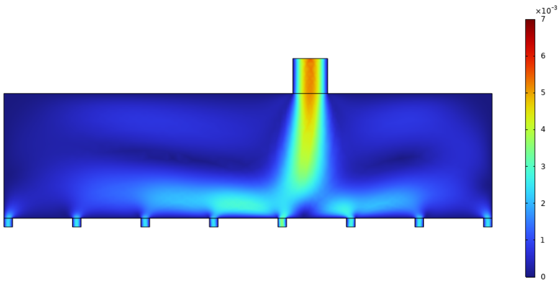
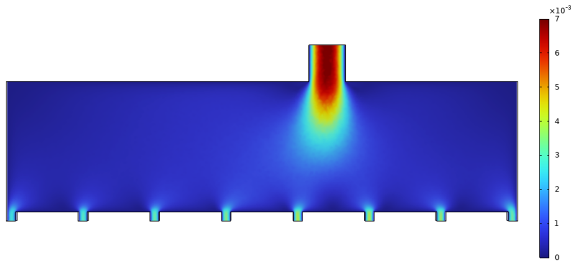


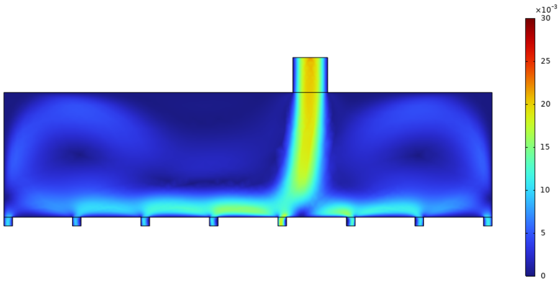
Figure 3.8: ED inside 2D and 3D initial domains for various Re values.



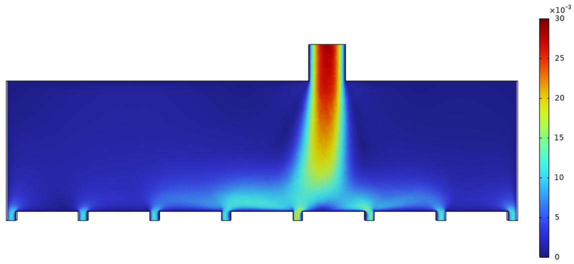
(a) Re = 50, 2D



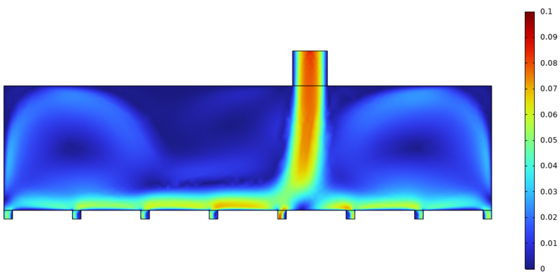
(b) Re = 50, 3D



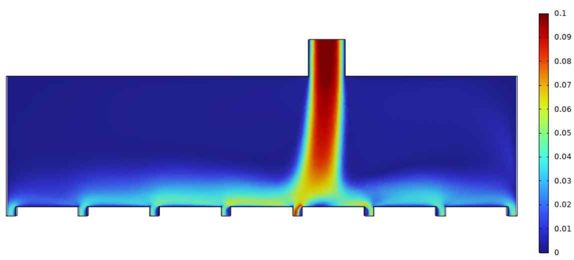
(c) Re = 200, 2D



(d) Re = 200, 3D



(e) Re = 800, 2D



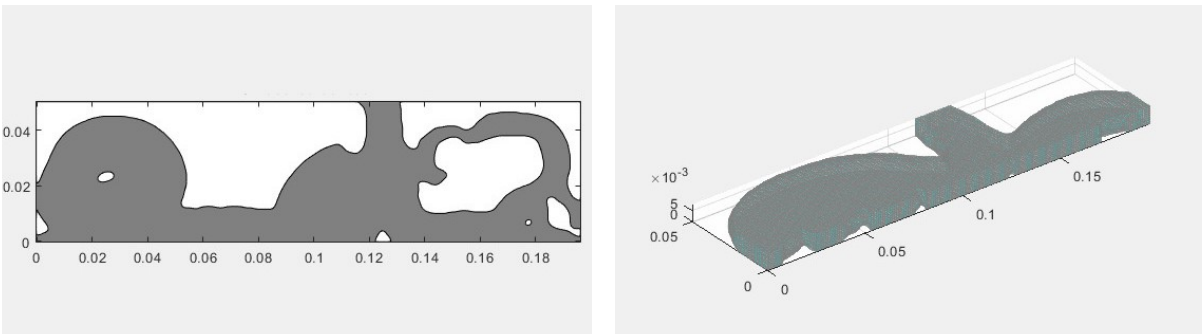
(f) Re = 800, 3D

Figure 3.9: Comparison of velocity contours (in m/s) in the initial domain.

However, in fig. 3.9, we can see a considerable difference in the nature of flow, especially at lower Re values between the 2D and 3D initial domain geometries. All 3D domains show negligible flow recirculation compared to the 2D ones. Comparing fig. 3.9a and 3.9b, it can be noted that while the former looks similar to high Re flows, in the latter the flow splits up just as it enters the domain. As we see later, this affects the design philosophy quite considerably. In contrast, the flow profile in the higher Re cases in fig. 3.9e and 3.9f is quite similar, save for the lack of recirculation.

3.2.2 Comparing 2D and 3D Optimization

Trends observed in the initial domain also predictably affect the optimization output. As seen in fig. 3.10, 3D optimization does not consider recirculation effects during the optimization process either. Moreover, as expected, there are striking differences in the 2D and 3D-optimized geometries for low Re values, and greater similarity between the high Re designs, save for the addition of a recirculation loop. Comparing the manifolds for an Re of 50 in fig. 3.11a and 3.11b, we see that the former splits off far closer to the outlets than the latter. This implied that 3D optimization has value for all Re over 2D optimization. Finally, we see that the manifolds for high Re are not only lower in volume, but closer to the outlets (and therefore the heat exchanger), making the entire assembly far more compact.

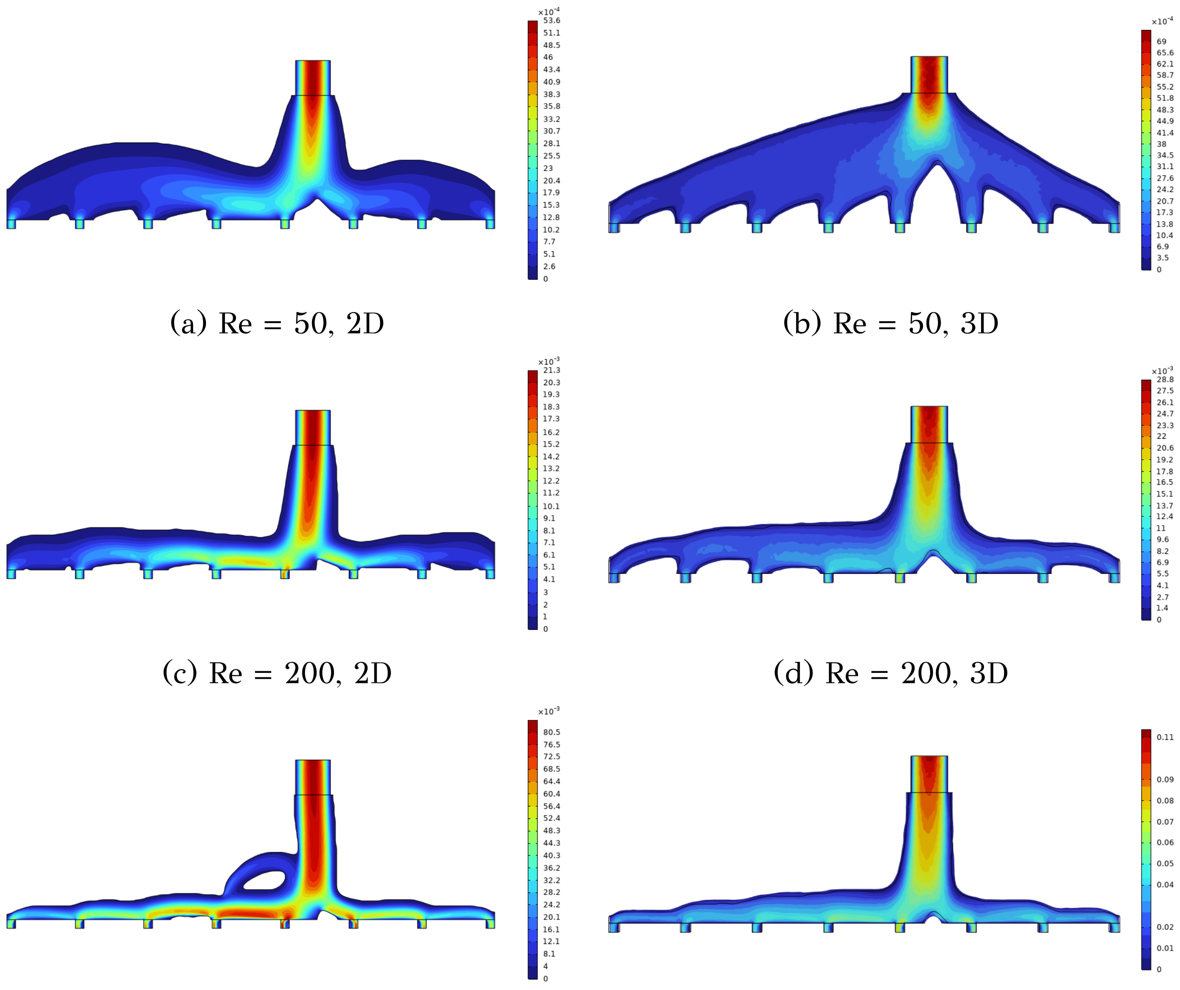


(a) 2D

(b) 3D

Figure 3.10: Comparison of 2D and 3D optimization with an inlet Re of 800 at the 15th iteration.

Looking at the performance of these manifolds in 3.13, we see that $f(\mathbf{x})$ follows similar trends. Higher Re manifolds distribute flow more equally (but not always), because we are able to set w lower. Overall performance however improves quite linearly with Re. Curiously, as opposed to fig. 3.7, we see that 3D-optimized manifolds appear to perform worse than their 2D-optimised counterparts. However, the 2D-optimization carries the assumption discussed in the beginning of the section - that the geometry is infinite in the z-direction. It stands to reason then that were the 2D geometry extruded and evaluated in a 3D solver, it would perform worse than the respective 3D optimized design.



(a) Re = 50, 2D (b) Re = 50, 3D

(c) Re = 200, 2D (d) Re = 200, 3D

(e) Re = 800, 2D (f) Re = 800, 3D

Figure 3.11: Comparison of velocity contours (in m/s) in the TO geometry.

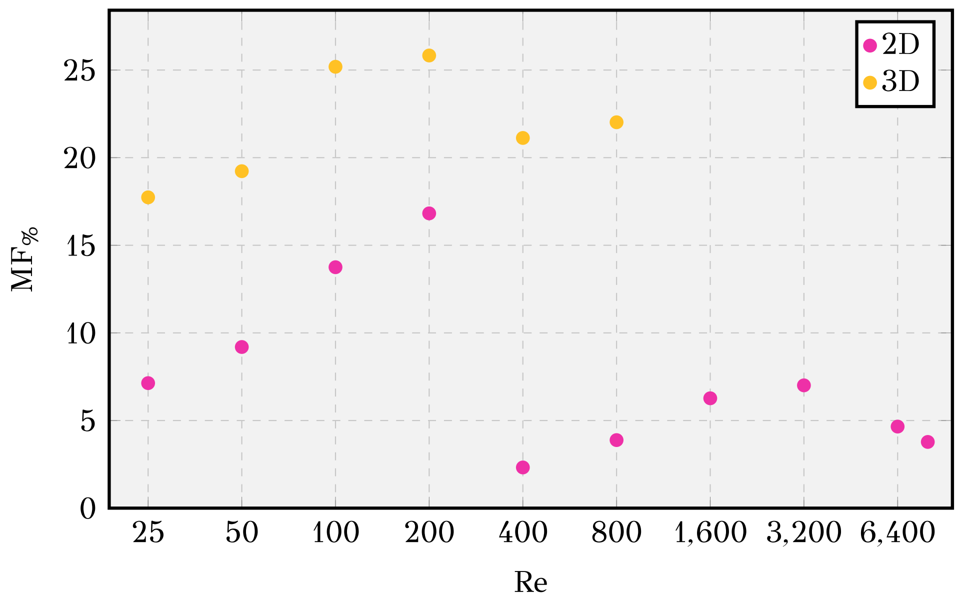


Figure 3.12: $MF_{\%}$ inside 2D and 3D optimized geometry for various Re values.

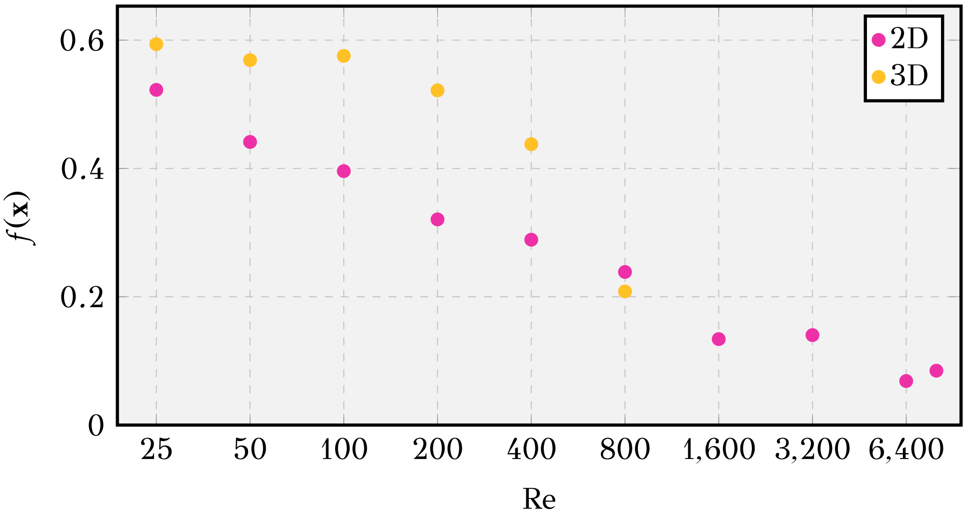


Figure 3.13: $f(\mathbf{x})$ for 2D and 3D optimized geometry for various Re values.

Lastly, the optimized manifolds are compared to the initial domain by plotting relative values for $MF_{\%}$ and $f(\mathbf{x})$, which can be defined as:

$$\begin{aligned} MF_r &= \frac{MF}{MF_0} \\ f(\mathbf{x})_r &= \frac{f(\mathbf{x})}{f(\mathbf{x})_0} \end{aligned} \tag{3.6}$$

This information, in fig. 3.14 and 3.15, allows us to track the optimization process at each Re. The lower MF_r and $f(\mathbf{x})_r$ the more effective the optimization has been. This reveals something interesting - in 2D optimization, performance improves over the initial domain for all Re, but 3D optimization reveals that performance improves only for the high Re manifolds. Therefore, for lower Re, there is in actuality a tradeoff between performance and compactness, while 2D optimization results would have us believe otherwise.

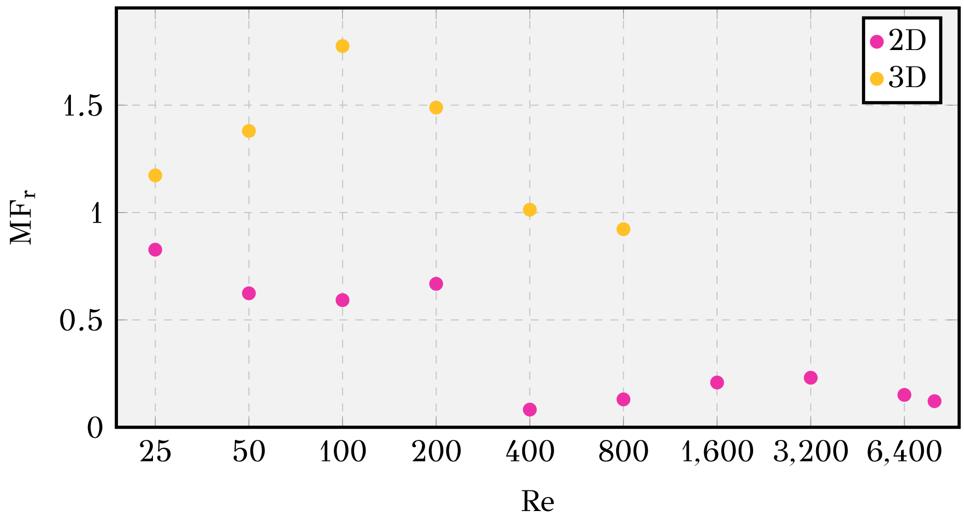


Figure 3.14: MF_r for various Re values.

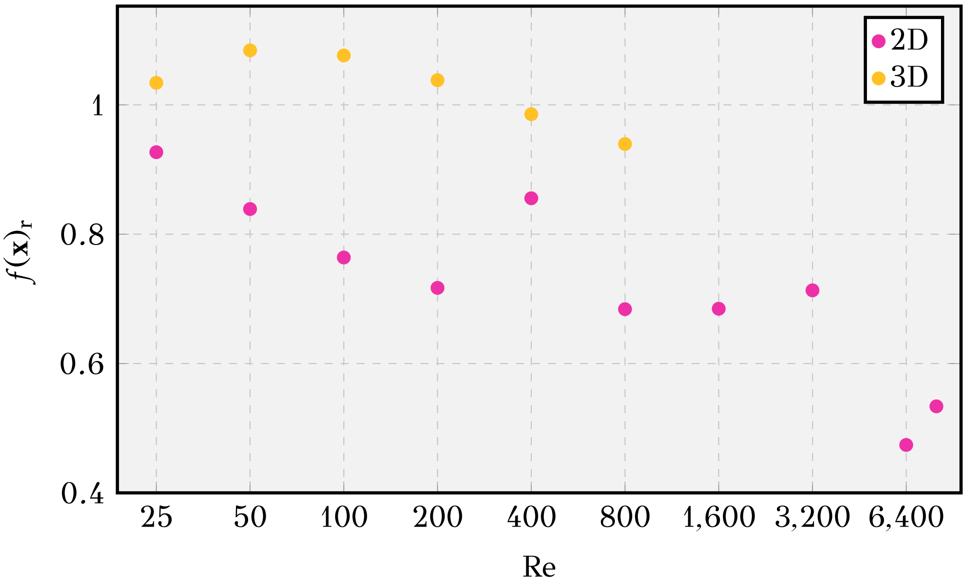
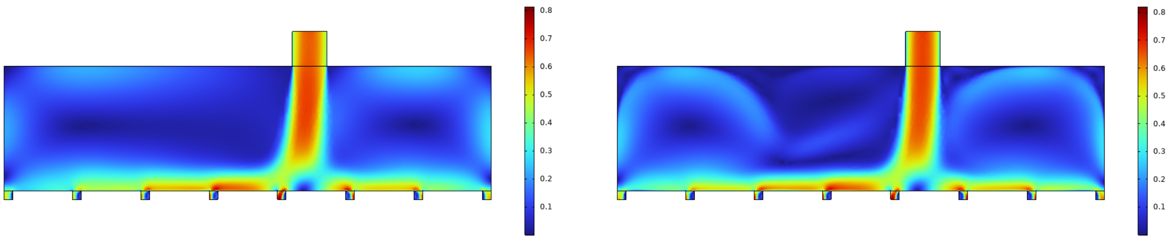


Figure 3.15: $f(\mathbf{x})_r$ for various Re values.

3.2.3 Comparing Laminar and Turbulent Optimization

As discussed before, both k-ϵ and k-ω models were evaluated for turbulent optimization. While initial work was done with the former, it was observed that the model is not very accurate at describing internal flows, as illustrated in fig. 3.16. The flow contours appear to be poorly defined which naturally affects the decisions taken by the optimizer. Therefore, all optimization results presented in this work for turbulent flow are with the k-ω model.

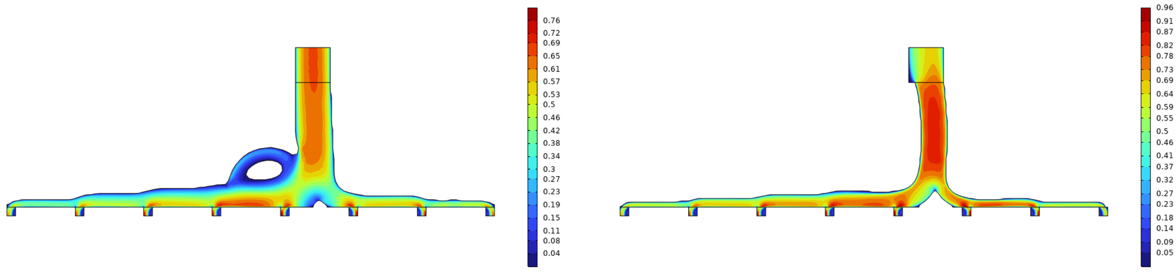


(a) (b)

Figure 3.16: Comparison of the (a) k-ϵ and (b) k-ω models.

The trend of lowering $f(\mathbf{x})$ continues with the onset of turbulence. This is likely because of turbulent flow mixing, which helps distribute the flow evenly across multiple outlets. Optimization was then performed for an inlet Re of 8000, which is the boundary condition the previous team envisaged in the final heat exchanger during operation. Two geometries, illustrated in fig. 3.17, were produced - one by driving the weight and volume as low as possible, and another with more conservative specifications. Their performance is tabulated in table 3.3.

Comparing fig. 3.11e and 3.17a, the turbulent manifold is a gradual evolution of the laminar designs for high Re. It is also far more compact and closer to the outlets than the lower Re manifolds and the one designed by the previous team.



(a) $w = 0.1$, $\overline{g} = 0.2$ (b) $w = 0.055$, $\overline{g} = 0.125$

Figure 3.17: Velocity contours (in m/s) in manifolds designed with (a) more conservative and (b) extreme values using TOBS-GT for Re=8000.

Parameter	Conservative Design	Extreme Design
$MF_{\%}$	7.429	3.778
ED	2.957×10^{-4}	5.939×10^{-4}
$f(\mathbf{x})$	0.024	0.013

Table 3.3: Performance of manifolds designed using TOBS-GT for Re=8000.

Comparing to fig. 3.17a, the extreme geometry is more compact, and also lacks the separate loop seen in the conservative design, which is hard to manufacture. However, the effect of reducing w is that ED rises considerably, to about double the value. This means that while the extreme manifold distributes flow better, pressure drop is far higher.

Despite all the optimization that has been performed in this work, it is possible that the performance improvement of the high Re geometries over the low Re designs is marginal. To rule out this possibility, the low Re optimised designs were tested for flow at the highest inlet Re of 8000, and the comparison of $MF_{\%}$ can be seen in fig. 3.18.

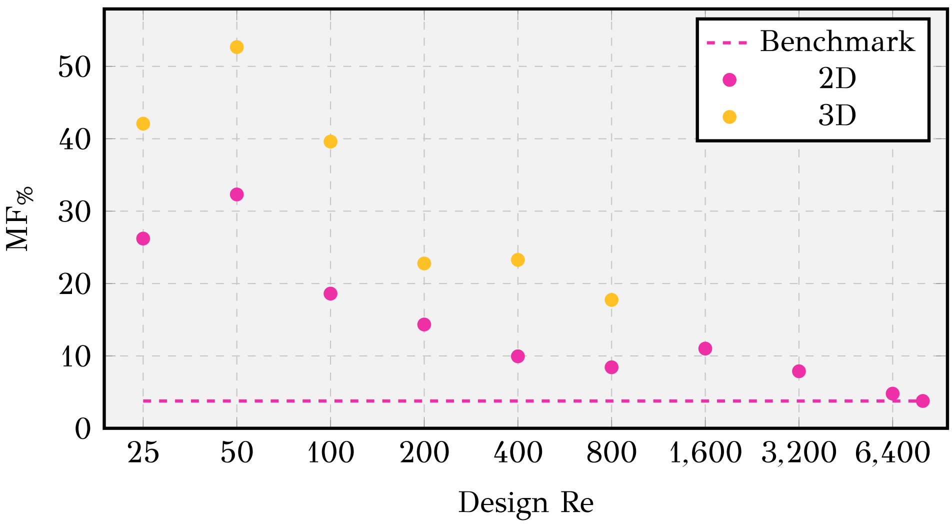
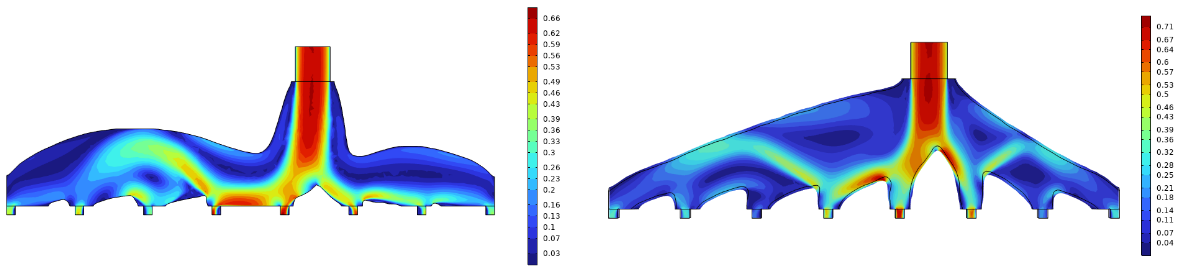


Figure 3.18: $MF_{\%}$ of manifolds optimized for other Re values benchmarked against the one optimised for an Re of 8000.



(a) Re = 50, 2D

(b) Re = 50, 3D

Figure 3.19: Velocity contours (in m/s) in manifolds designed for low Re being used for an Re of 8000.

It is observed that the manifolds under-perform not only compared to the manifold specifically optimized for an Re of 8000, but also compared to their own performance at their respective design Re value. This can also be observed in the flow contours within the design (see fig. 3.19), which shows the poor distribution quite visually. The manifold performance improves almost uniformly as the Re of optimization approaches the Re of operation.

3.3 Experimental Validation

Physical models of the manifolds were produced with a 2-part design: a resin 3D-printed bottom section and a laser-cut top plate. Using the method described in section 2.3, validation was performed first on the manifold produced by the previous team to dial in the method. Then the two geometries specifically designed for an Re of 8000 were also validated. Throughout the process, validation has been tricky to achieve.

3.3.1 Shape-optimized Laminar Flow Manifold

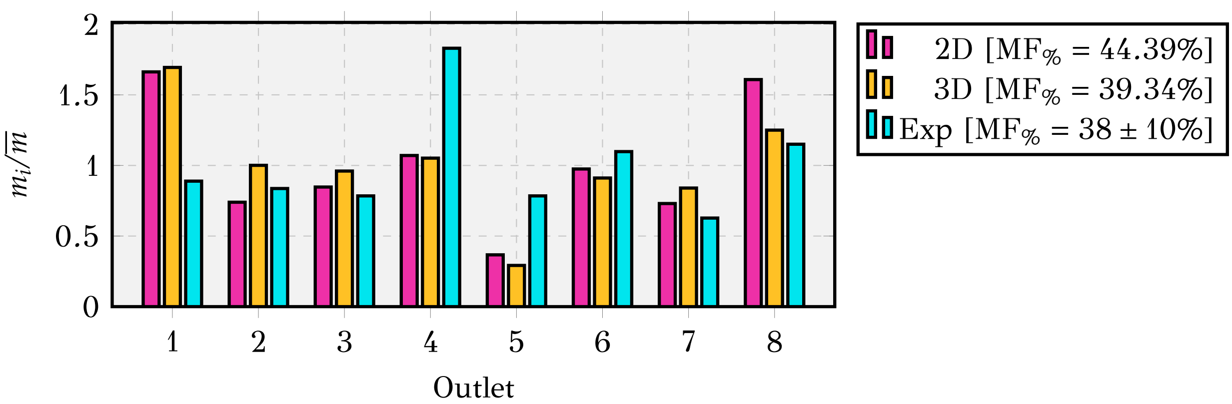


Figure 3.20: A comparison of the 2D and 3D models of laminar flow manifold optimised by the previous team at an Re of 8000 with experimental validation.

This manifold shows a rather conflicted validation. The CFD model was quite sensitive to small changes in mesh sizing parameters, and the geometry had a large number of thin zones with sudden curvature changes. In fig. 3.20 we can see that while the $MF_{\%}$ is very close for 3D CFD and the experiment, the flow patterns are not at all similar. While CFD related problems were not seen in the TOBS-GT geometries, the 3D-printed model was adjusted for those tests by removing the outlet tubes seen in fig. 3.21.

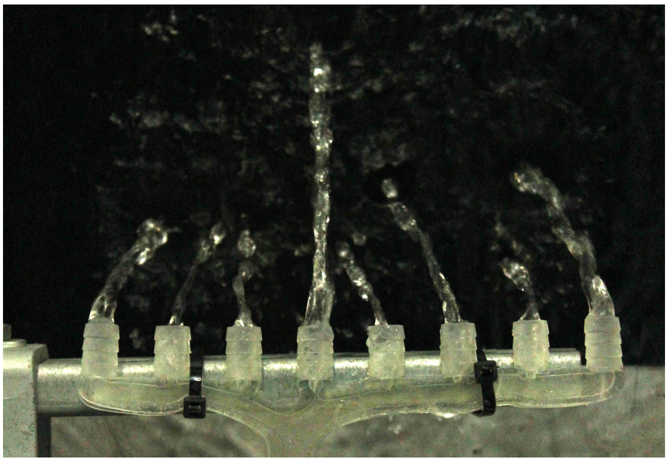


Figure 3.21: Experimental validation of geometry from fig. 3.6.

3.3.2 Topology-optimized Turbulent Flow Manifold with Conservative Parameters

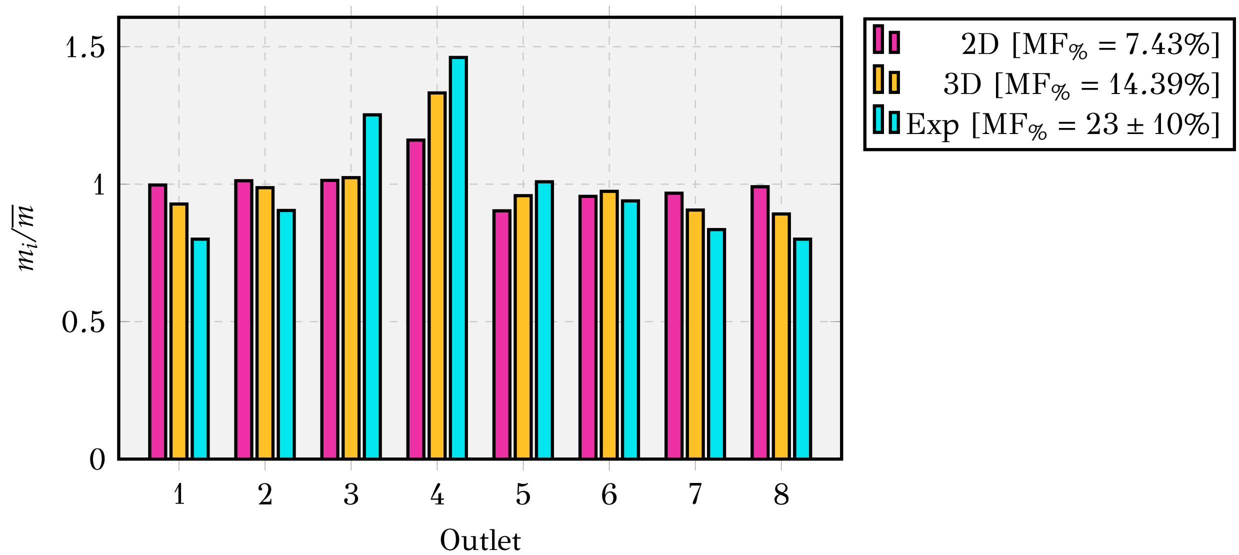


Figure 3.22: A comparison of the 2D and 3D models of manifold optimised for an Re of 8000 with conservative parameters with experimental validation.

This design is validated - not only is the final value of the $MF_{\%}$ for the 3D CFD model within the experimental error range, the flow patterns also match. The same outlets are preferred in the CFD and experimental case. An expected result is that 2D model not only fails validation quantitatively due to the disparity between $MF_{\%}$ values, but also qualitatively as the flow patterns do not match as closely. This suggests that the behaviour used by the 2D optimizer of using flow recirculation zones is perhaps non-physical.

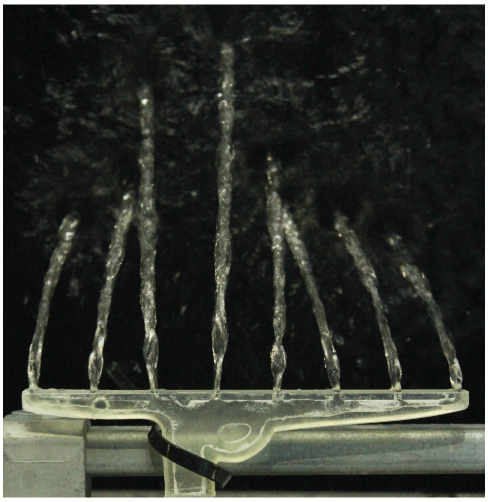


Figure 3.23: Experimental validation of geometry from fig. 3.17a.

3.3.3 Topology-optimized Turbulent Flow Manifold with Extreme Parameters

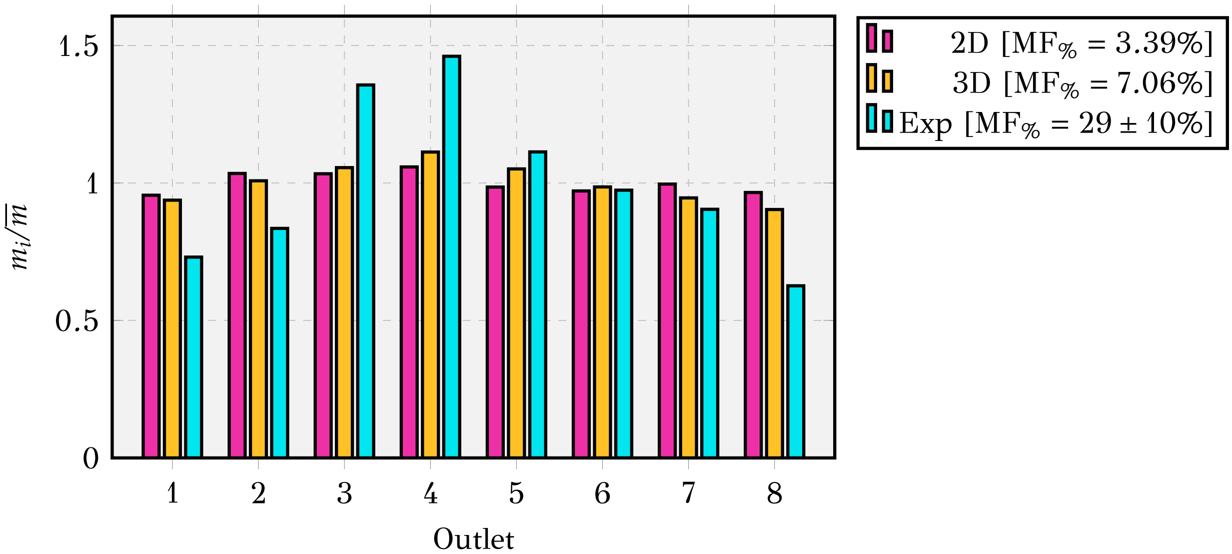


Figure 3.24: A comparison of the 2D and 3D models of manifold optimised for an Re of 8000 with extreme parameters with experimental validation.

The extreme design was not successfully validated. The disparity between the 3D CFD and experimental $MF_{\%}$ value is too great, even though there is a similarity between the outlet flow patterns. We do see again the trend here for 2D results to be more different both numerically and qualitatively.

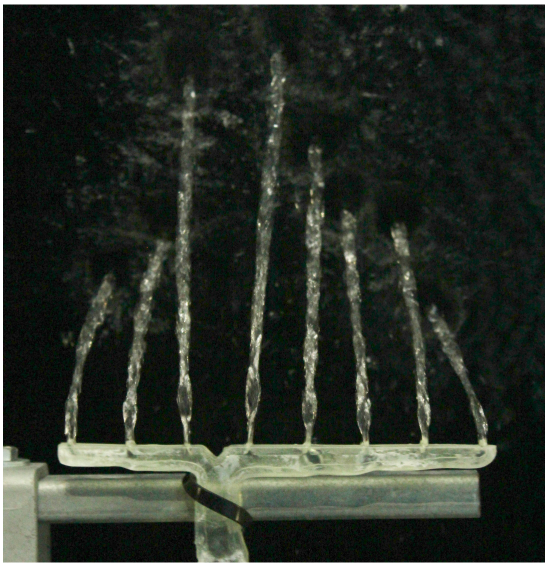


Figure 3.25: Experimental validation of geometry from fig. 3.17b.

Validation work shows us that it is difficult to produce optimized geometry that works just like its CFD model. There are some potential reasons these models are troublesome to validate:

- Optimization and CFD modelling was done with the assumption of fully developed flow at the inlet, as discussed in section 3.1.1. For this condition to be realized in practice, there needs to be a development region of atleast 10 times the inlet width.[32] Not only is this condition not satisfied in experimental validation due to the dimensions of the 3D printer, it might not be satisfied in operation. As seen in fig. 1.5, the first stage is likely to affect the inlet boundary condition for the manifold in the second stage, which might not be fully developed. Comparing fig. 3.26 and 1.8, we see that this can significantly affect the flow, and hence the optimization.

- The roughness of a 3D-printed design has not been considered during optimization, which assumes smooth walls. Especially for turbulent flow, this may also affect the result. The roughness characteristics are also likely to be different for the resin-printed prototype, and the metal-printed final design.

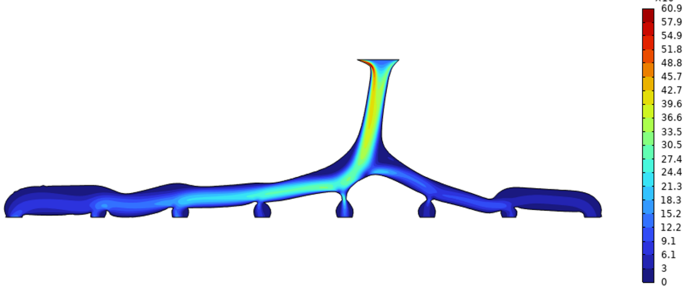


Figure 3.26: 2D velocity contours (in m/s) in the shape-optimized geometry for laminar flow, with uniform velocity prescribed at inlet.

3.4 Roughness and its Effects

As stated in section 1.1, for roughness to meaningfully affect the flow, it must have sufficient height to pierce the viscous sub-layer in a turbulent flow. This condition is simple to check using correlations for channel flows. First, we calculate the skin-friction coefficient using Prandtl's formula as:

$$C_\mathrm{f} = \frac{0.027}{\mathrm{Re}^{1/7}} = 0.007 \tag{3.7}$$

We then calculate the friction velocity as:

$$u_* = \sqrt{C_\mathrm{f} u_\mathrm{free}^2 / 2} = 0.041\,\mathrm{m/s} \tag{3.8}$$

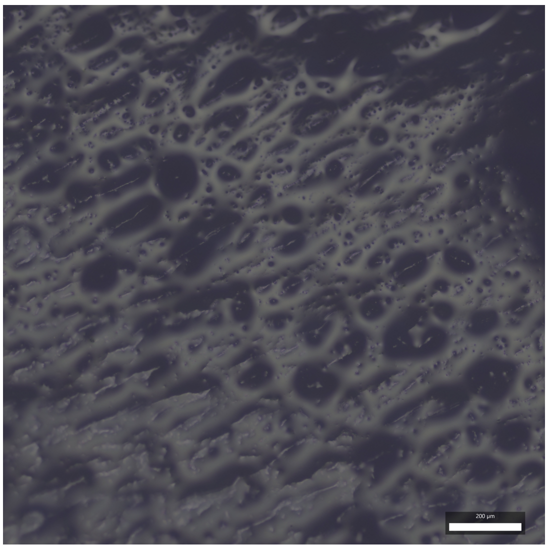
Where u_free is the free-stream velocity ($\approx 0.675\,\mathrm{m/s}$). With these values, we can calculate the height of the viscous sub-layer for channel flows ($y^+ = 5$) as:

$$y = \frac{y^+ \nu}{u_*} \approx 120\,\mu\mathrm{m} \tag{3.9}$$

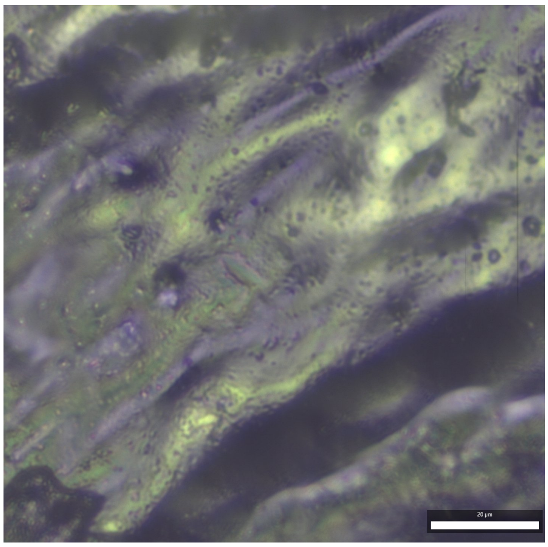
When wall roughness is comparable to this height, we can assume that there will be an effect on the flow, and hence the performance of the manifold.

3.4.1 Surface of Resin-Printed Part

To understand the nature of roughness, the manifold was photographed with a microscope. As seen in 3.27, the surface has 50 µm wide striations, corresponding neatly with the layer height specified in the resin-printer.



(a) 10x magnified (scale: 200 µm)



(b) 40x magnified (scale: 20 µm)

Figure 3.27: Microscope images of the surface of a resin-printed prototype.

Since it is difficult to photograph the surface from a place perpendicular to it with the microscope, the height of the surface structures could not be estimated. However, if the height of the structures is comparable in magnitude to their width, we can see that it would be well within the viscous sub-layer.

3.4.2 Numerical Modelling of Roughness

This roughness was modelled in COMSOL as sand-grain roughness. The k-ω model was used again, but this time with wall functions. COMSOL allows for specification of roughness by modifying the wall functions. The geometry from fig. 3.17a was used.

Various roughness heights were evaluated to observe the hypothetical effect. From the results in fig. 3.28, we can see that there is a slight decrease in $MF_{\%}$ with increasing roughness height, but the effect is nevertheless quite minor below a roughness height of 800 µm. We can conclude then that it is already well within the capabilities of resin-printers to produce a functionally smooth surface for such flow rates.

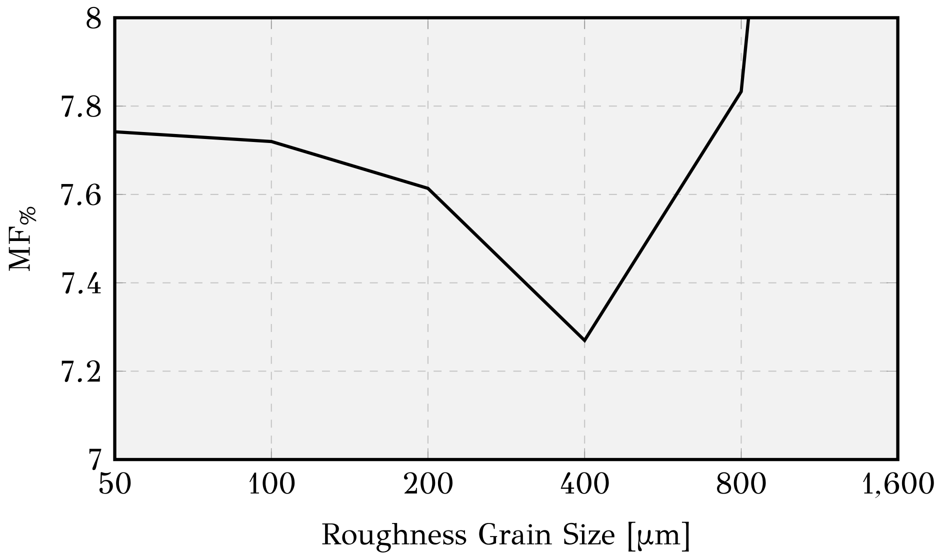


Figure 3.28: Effect of roughness on performance.

There are two main caveats to this conclusion:

1. The roughness is not necessarily in the form of uniform sand-grain roughness, but as non-continuous ridges of a certain width. It is not certain that the two have a similar effect on the flow, and more research would be needed on the latter, especially accounting for print orientation. This might necessitate a different solver.
2. The final part is to be printed with metal, which will again present differing roughness characteristics to the resin-printed prototypes.

Chapter 4

Conclusions

The following conclusions were reached over the course of this work:

- Tests on the shape-optimised manifold created by the previous team showed that switching to turbulence in a manifold designed for laminar flow significantly worsens performance. Therefore, in this work the effect of increasing reynolds number was analysed on optimisation results.
- Optimization must be done for the exact boundary conditions under which the part is expected to operate. A manifold optimized for laminar flow cannot be expected to perform well with turbulent flow. This also means that if a manifold is expected to operate at multiple flow regimes, this should be factored into optimisation.
- With the chosen optimization methodology and objective function definition, manifolds optimized for turbulent flow are able to achieve better flow distribution than ones designed for laminar flow at even their respective design conditions.
- Manifolds designed for higher reynolds numbers are also able to achieve a more compact design without compromising the flow distribution. This geometry not only has lower volume overall but is closer to the outlets, thus producing a compact heat exchanger overall.
- Where laminar flow (especially at lower reynolds numbers) is expected, it is worthwhile to optimize with a 3D domain, despite the drastic increase in computational cost. In these cases, the out-of-plane components of the geometry (such as the top and bottom wall) have a strong effect on the final optimized design.
- 3D optimization is also recommended for higher reynolds numbers, despite the overall geometry concept being similar. The 2D optimizer tends to overvalue the influence of recirculation effects, thus adding tough-to-manufacture yet useless loops to the manifold design.
- For reynolds numbers of 8000, the role of roughness can be largely ignored in most resin-printed parts.

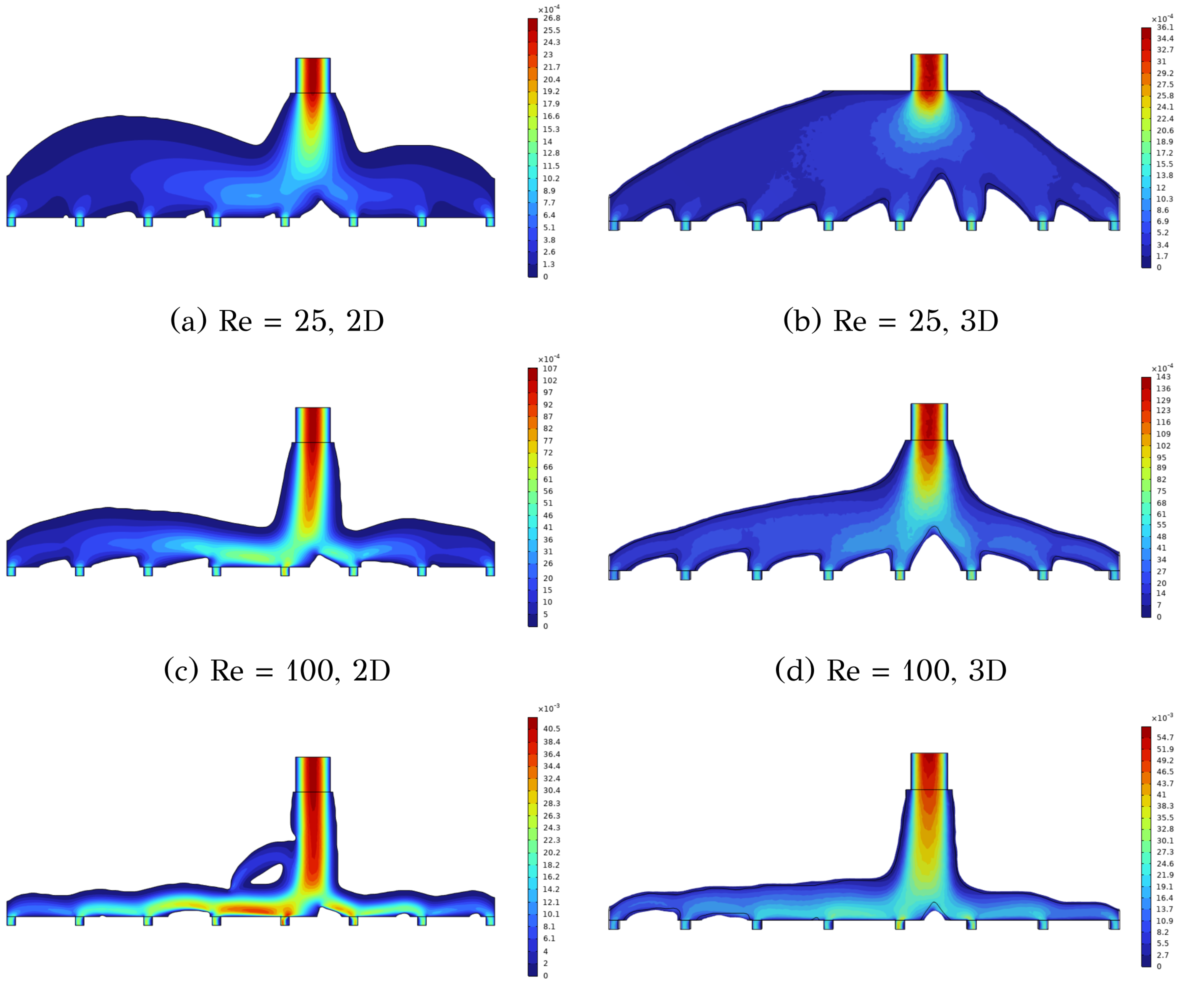
Chapter 5

Recommendations

The following recommendations can be made for future investigation based on the limitations of this work:

- The inlet boundary condition for the second stage can be studied by conducting CFD and validation of the first stage. This will make the optimization process more reflective of real operating conditions.
- As mentioned, roughness could not be faithfully modelled in this work, and might well disrupt the flow for metal-printed parts. A potential project could physically characterise, then model, and finally optimize with the roughness of a physical metal 3D-printed part.
- An exploration of 3D turbulent optimization can show whether the trends identified with increasing reynolds numbers for differences between 2D and 3D optimization can be extended for turbulent flow. With the non-linear nature of the transition to turbulence, this is not guaranteed.
- A better validation technique is needed for future work. The error margin with the technique used in this work is too high. It also lacks the capability to validate lower flow rates, including the entire laminar flow regime unless viscous fluids other than water are used. A possible alternative could be particle image/tracking velocimetry or simply using a separate flow meter for each individual outlet.
- There is also potential to validate the geometry at various stages of optimization. Validating intermediate geometries produced before the final iteration can verify that optimization is based on sound physical models throughout.
- The rules found in this work have far-reaching consequences for setting up a manifold optimization problem. However, optimization needs to be performed with a variety of boundary conditions in a variety of domains to establish if they are indeed general rules.
- Building a TOBS-GT optimization code based on the finite volume method more commonly used in CFD applications can enable far better performance than the finite element method-based solution used here.

Appendix



(a) Re = 25, 2D

(b) Re = 25, 3D

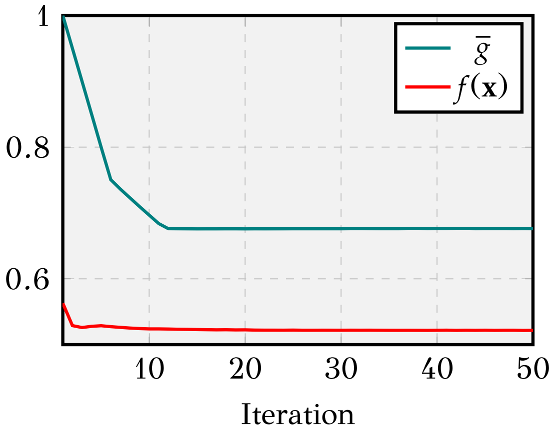
(c) Re = 100, 2D

(d) Re = 100, 3D

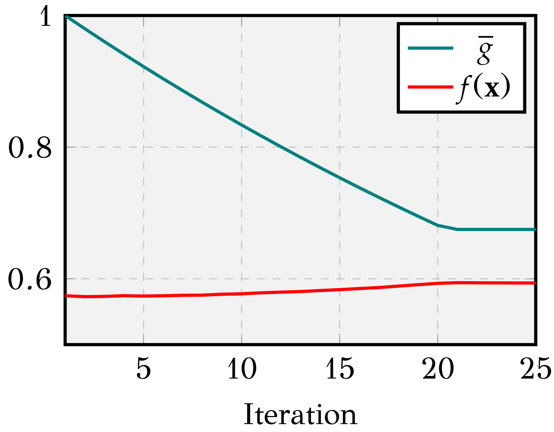
(e) Re = 400, 2D

(f) Re = 400, 3D

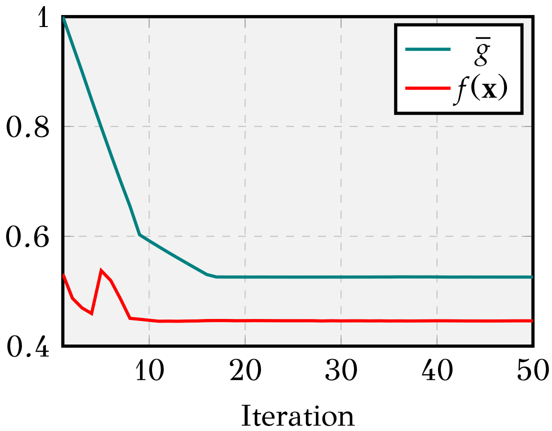
Figure 5.1: Comparison of velocity contours (in m/s) in the TO geometry.



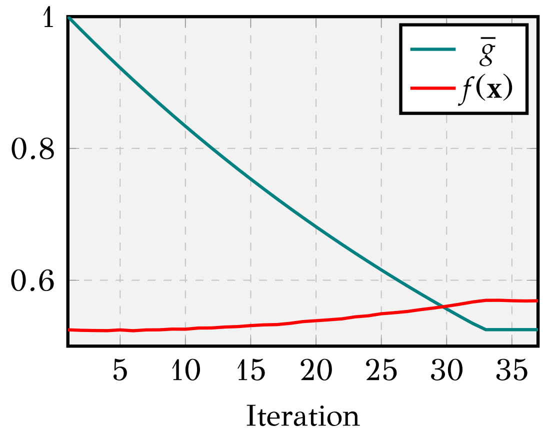
(a) Re = 25, 2D



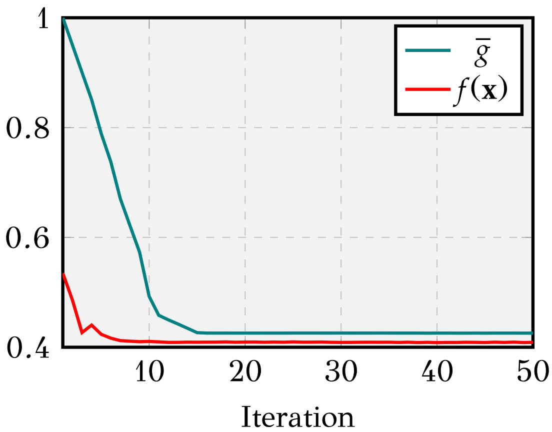
(b) Re = 25, 3D



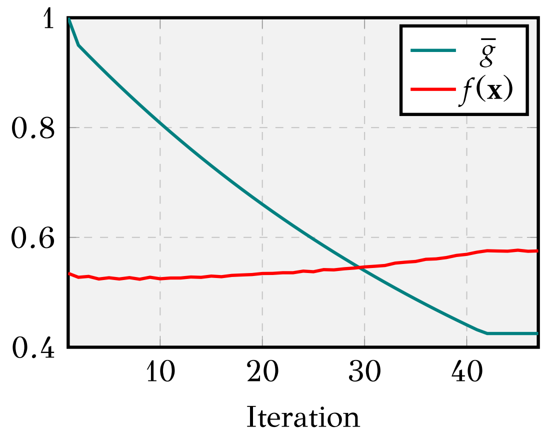
(c) Re = 50, 2D



(d) Re = 50, 3D

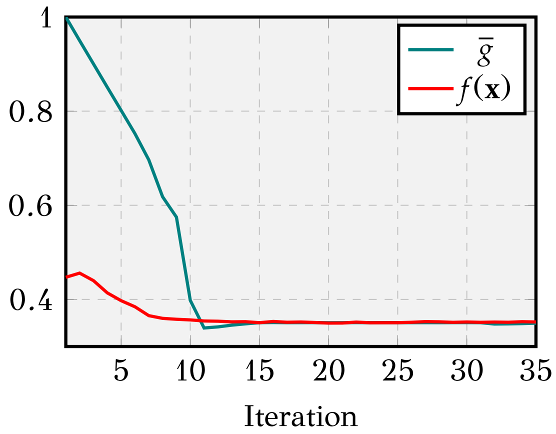


(e) Re = 100, 2D

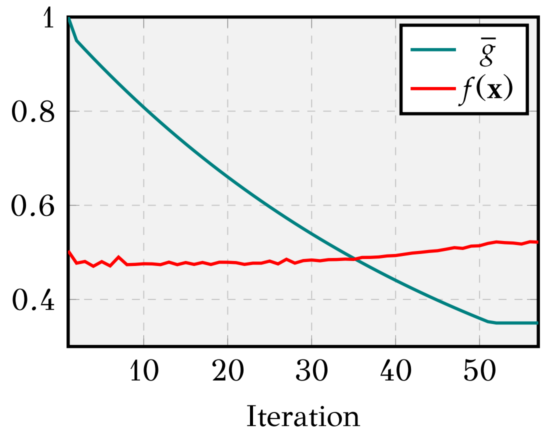


(f) Re = 100, 3D

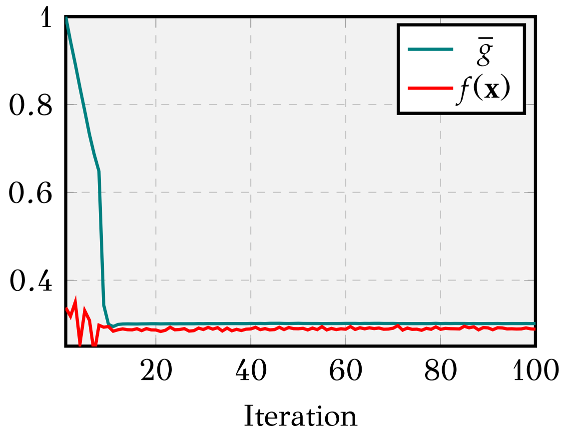
Figure 5.2: Optimization progress information for laminar manifolds.



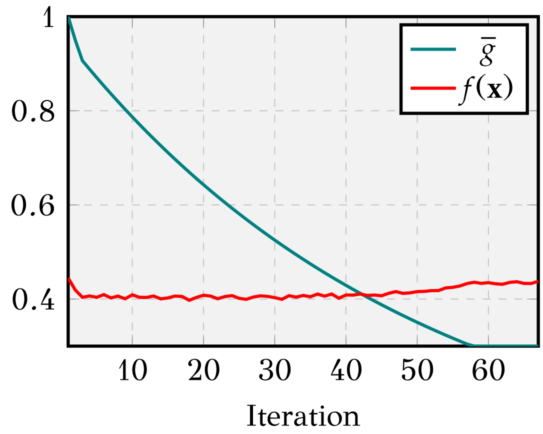
(a) Re = 200, 2D



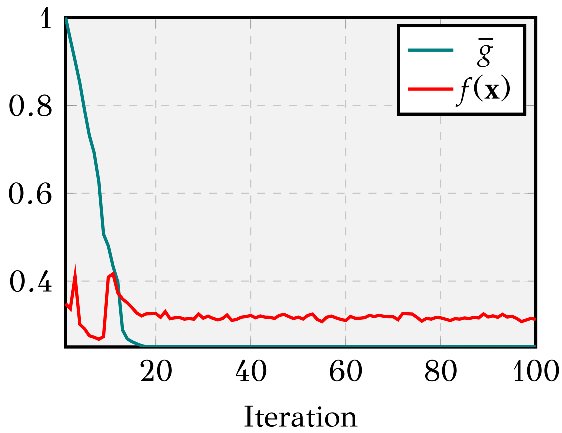
(b) Re = 200, 3D



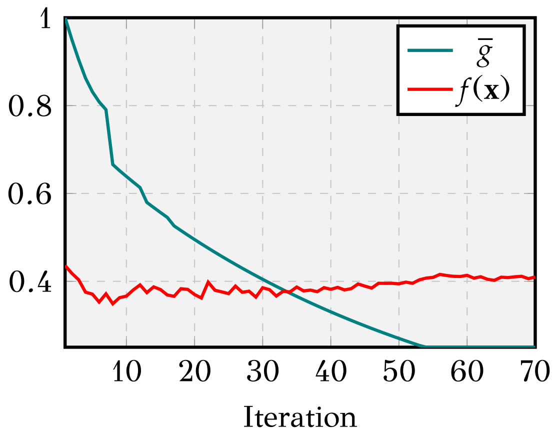
(c) Re = 400, 2D



(d) Re = 400, 3D

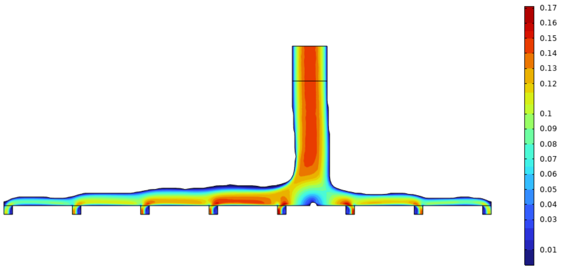


(e) Re = 800, 2D

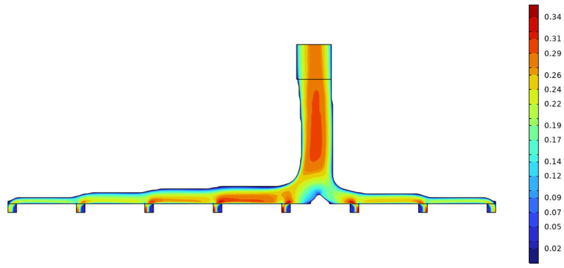


(f) Re = 800, 3D

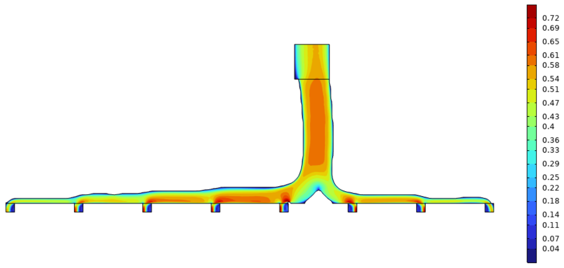
Figure 5.3: Optimization progress information for laminar manifolds (continued).



(a) Re = 1600, 2D

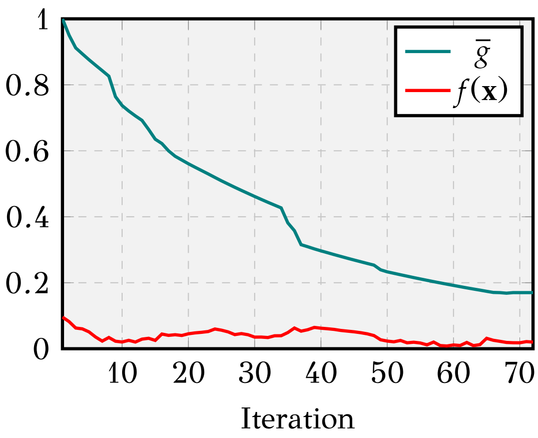


(b) Re = 3200, 2D

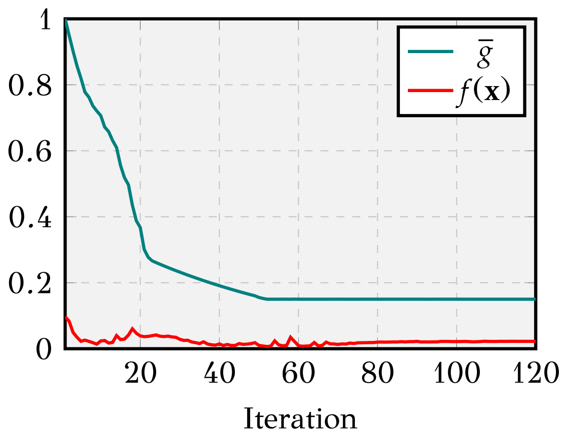


(c) Re = 6400, 2D

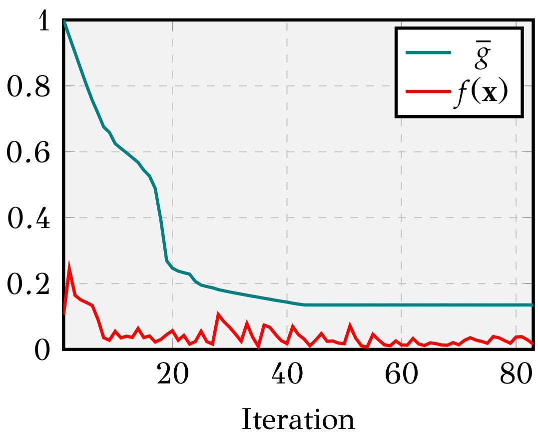
Figure 5.4: Velocity contours (in m/s) in the TO geometry for turbulent flow.



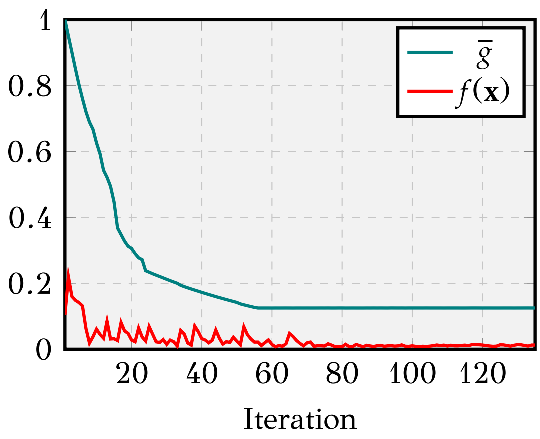
(a) Re = 1600



(b) Re = 3200



(c) Re = 6400



(d) Re = 8000 (extreme parameters)

Figure 5.5: Optimization progress information for turbulent manifolds.

References

[1] All3DP. "The 7 main types of 3d printing technology in 2022." (2021), [Online]. Available: https://all3dp.com/1/types-of-3d-printers-3d-printing-technology/.

[2] J. McDonough, "A perspective on the current and future roles of additive manufacturing in process engineering, with an emphasis on heat transfer," *Thermal Science and Engineering Progress*, vol. 19, p. 100 594, 2020, ISSN: 2451-9049. DOI: 10.1016/j.tsep.2020.100594.

[3] R. E. Laureijs, J. B. Roca, S. P. Narra, C. Montgomery, J. L. Beuth, and E. R. H. Fuchs, "Metal Additive Manufacturing: Cost Competitive Beyond Low Volumes," *Journal of Manufacturing Science and Engineering*, vol. 139, no. 8, May 2017, 081010, ISSN: 1087-1357. DOI: 10.1115/1.4035420.

[4] F. Cabanettes, A. Joubert, G. Chardon, *et al.*, "Topography of as built surfaces generated in metal additive manufacturing: A multi scale analysis from form to roughness," *Precision Engineering*, vol. 52, pp. 249–265, 2018, ISSN: 0141-6359. DOI: 10.1016/j.precisioneng.2018.01.002.

[5] F. T. M. Nieuwstadt, B. J. Boersma, and J. Westerweel, "Turbulent flows," in *Turbulence: Introduction to Theory and Applications of Turbulent Flows*. Cham: Springer International Publishing, 2016, pp. 87–123, ISBN: 978-3-319-31599-7. DOI: 10.1007/978-3-319-31599-7_6.

[6] K. Dutkowski, M. Kruzel, and K. Rokosz, "Review of the state-of-the-art uses of minimal surfaces in heat transfer," *Energies*, vol. 15, no. 21, 2022, ISSN: 1996-1073. DOI: 10.3390/en15217994.

[7] Exploratorium. "Catenoid." (2008), [Online]. Available: https://www.flickr.com/photos/exploratorium/2421180181.

[8] Anders Sandberg (Wikimedia commons). "Approximation of the schwarz d minimal surface generated using ken brakke's surface evolver program." (2012), [Online]. Available: https://en.wikipedia.org/wiki/Schwarz_minimal_surface#/media/File:Schwarz_D_Surface.png.

[9] P. Minqiang, Z. Dehuai, T. Yong, and C. Dongqing, "Cfd-based study of velocity distribution among multiple parallel microchannels," *J. Comput*, vol. 4, no. 11, pp. 1133–1138, 2009.

[10] P. Dąbrowski, "Comparison of various flow maldistribution quantification methods in mini heat exchangers," *Scientific Reports*, vol. 13, no. 1, p. 11 482, Jul. 2023, ISSN: 2045-2322. DOI: 10.1038/s41598-023-38784-5.

[11] J. Wen and Y. Li, "Study of flow distribution and its improvement on the header of plate-fin heat exchanger," *Cryogenics*, vol. 44, no. 11, pp. 823–831, 2004, ISSN: 0011-2275. DOI: 10.1016/j.cryogenics.2004.04.009.

[12] K. Zhang, M.-J. Li, H. Liu, J.-G. Xiong, and Y.-L. He, "A general and rapid method to evaluate the effect of flow maldistribution on the performance of heat exchangers," *International Journal of Thermal Sciences*, vol. 170, p. 107 152, 2021, ISSN: 1290-0729. DOI: 10.1016/j.ijthermalsci.2021.107152.

[13] P. R. Bobbili, B. Sunden, and S. K. Das, "An experimental investigation of the port flow maldistribution in small and large plate package heat exchangers," *Applied Thermal Engineering*, vol. 26, no. 16, pp. 1919–1926, 2006, ISSN: 1359-4311. DOI: 10.1016/j.applthermaleng.2006.01.015.

[14] T. Borrvall and J. Petersson, "Topology optimization of fluids in stokes flow," *International Journal for Numerical Methods in Fluids*, vol. 41, no. 1, pp. 77–107, 2003. DOI: 10.1002/fld.426.

[15] A. Gersborg-Hansen, O. Sigmund, and R. B. Haber, "Topology optimization of channel flow problems," *Structural and Multidisciplinary Optimization*, vol. 30, no. 3, pp. 181–192, Sep. 2005, ISSN: 1615-1488. DOI: 10.1007/s00158-004-0508-7.

[16] A. Gersborg-Hansen, M. P. Bendsøe, and O. Sigmund, "Topology optimization of heat conduction problems using the finite volume method," *Structural and Multidisciplinary Optimization*, vol. 31, no. 4, pp. 251–259, Apr. 2006, ISSN: 1615-1488. DOI: 10.1007/s00158-005-0584-3.

[17] C. Othmer, "A continuous adjoint formulation for the computation of topological and surface sensitivities of ducted flows," *International Journal for Numerical Methods in Fluids*, vol. 58, no. 8, pp. 861–877, 2008. DOI: 10.1002/fld.1770.

[18] E. A. Kontoleontos, E. M. Papoutsis-Kiachagias, A. S. Zymaris, D. I. Papadimitriou, and K. C. Giannakoglou, "Adjoint-based constrained topology optimization for viscous flows, including heat transfer," *Engineering Optimization*, vol. 45, no. 8, pp. 941–961, 2013. DOI: 10.1080/0305215X.2012.717074.

[19] C. B. Dilgen, S. B. Dilgen, D. R. Fuhrman, O. Sigmund, and B. S. Lazarov, "Topology optimization of turbulent flows," *Computer Methods in Applied Mechanics and Engineering*, vol. 331, pp. 363–393, 2018, ISSN: 0045-7825. DOI: 10.1016/j.cma.2017.11.029.

[20] M. P. Bendsøe and N. Kikuchi, "Generating optimal topologies in structural design using a homogenization method," *Computer Methods in Applied Mechanics and Engineering*, vol. 71, no. 2, pp. 197–224, 1988, ISSN: 0045-7825. DOI: 10.1016/0045-7825(88)90086-2.

[21] R. Sivapuram and R. Picelli, "Topology optimization of binary structures using integer linear programming," *Finite Elements in Analysis and Design*, vol. 139, pp. 49–61, 2018, ISSN: 0168-874X. DOI: 10.1016/j.finel.2017.10.006.

[22] B. C. Souza, P. V. M. Yamabe, L. F. N. Sá, S. Ranjbarzadeh, R. Picelli, and E. C. N. Silva, "Topology optimization of fluid flow by using integer linear programming," *Structural and Multidisciplinary Optimization*, vol. 64, no. 3, pp. 1221–1240, Sep. 2021, ISSN: 1615-1488. DOI: 10.1007/s00158-021-02910-6.

[23] R. Picelli, E. Moscatelli, P. V. M. Yamabe, *et al.*, "Topology optimization of turbulent fluid flow via the tobs method and a geometry trimming procedure," *Structural and Multidisciplinary Optimization*, vol. 65, no. 1, p. 34, Jan. 2022, ISSN: 1615-1488. DOI: 10.1007/s00158-021-03118-4.

[24] C. Dapogny, P. Frey, F. Omnès, and Y. Privat, "Geometrical shape optimization in fluid mechanics using freefem++," *Structural and Multidisciplinary Optimization*, vol. 58, no. 6, pp. 2761–2788, Dec. 2018, ISSN: 1615-1488. DOI: 10.1007/s00158-018-2023-2.

[25] J. Alexandersen and C. S. Andreasen, "A review of topology optimisation for fluid-based problems," *Fluids*, vol. 5, no. 1, 2020, ISSN: 2311-5521. DOI: 10.3390/fluids5010029.

[26] A. H. Land and A. G. Doig, "An automatic method of solving discrete programming problems," *Econometrica*, vol. 28, no. 3, pp. 497–520, 1960, ISSN: 00129682, 14680262. DOI: 10.2307/1910129.

[27] R. T. Haftka and Z. Gürdal, *Elements of Structural Optimization*. Dordrecht: Springer Netherlands, 1992, pp. 255–304, ISBN: 978-94-011-2550-5. DOI: 10.1007/978-94-011-2550-5.

[28] R. Picelli, R. Sivapuram, and Y. M. Xie, "A 101-line matlab code for topology optimization using binary variables and integer programming," *Structural and Multidisciplinary Optimization*, vol. 63, no. 2, pp. 935–954, Feb. 2021, ISSN: 1615-1488. DOI: 10.1007/s00158-020-02719-9.

[29] X. Huang and Y. M. Xie, "Bi-directional evolutionary topology optimization of continuum structures with one or multiple materials," *Computational Mechanics*, vol. 43, no. 3, pp. 393–401, Feb. 2009, ISSN: 1432-0924. DOI: 10.1007/s00466-008-0312-0.

[30] D. Wilcox, *Turbulence Modeling for CFD*. DCW Industries, Incorporated, 1994, ISBN: 9780963605108. [Online]. Available: https://books.google.nl/books?id=VwlRAAAAMAAJ.

[31] R. J. Moffat, "Describing the uncertainties in experimental results," *Experimental Thermal and Fluid Science*, vol. 1, no. 1, pp. 3–17, 1988, ISSN: 0894-1777. DOI: https://doi.org/10.1016/0894-1777(88)90043-X.

[32] Y. Çengel and J. Cimbala, *Fluid Mechanics: Fundamentals and Applications*. McGraw-Hill Education, 2018, ISBN: 9781259921902. [Online]. Available: https://books.google.nl/books?id=DqFJvgAACAAJ.